ICCC 2014

June 10-13, 2014

Ljubljana, Slovenia

Proceedings

of the

Fifth International Conference on

Computational Creativity

Editors:

Simon Colton

Dan Ventura

Nada Lavrač

Michael Cook

Proceedings of the
Fifth International Conference
on Computational Creativity
ICCC-2014
June 10-13, 2014
Ljubljana, Slovenia

Editors: Simon Colton, Dan Ventura, Nada Lavrač, Michael Cook

Jožef Stefan Institute

Ljubljana

Slovenia

http://computationalcreativity.net/iccc2014

First published 2014

TITLE: PROCEEDINGS OF THE FIFTH INTERNATIONAL CONFERENCE ON COMPUTATIONAL CREATIVITY

EDITOR: SIMON COLTON, DAN VENTURA, NADA LAVRAČ, MICHAEL COOK

ISBN: 978-961-264-055-2

Cover page design: Damjan Demšar

Logo design: Damjan Demšar

Technical editors: Dragana Miljković, Nada Lavrač and Senja Pollak

Foreword

Computational Creativity is the art, science, philosophy and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviors that unbiased observers would deem to be creative. As a field of research, this area is thriving, with progress in formalizing what it means for software to be creative, along with many exciting and valuable applications of creative software in the sciences, the arts, literature, gaming and elsewhere.

The Fifth International Conference on Computational Creativity (ICCC-2014) was held from June 10 to 13, 2014 at the Jožef Stefan Institute, Ljubljana, Slovenia. It was the fifth in the series of ICCC conferences, following the conferences held in Sydney (2013), Dublin (2012), Mexico City (2011) and Lisbon (2010). This conference series was preceeded by several International Joint Workshops on Computational Creativity (IJWCC), held in Madrid (2008), London (2007), Riva del Garda (2006), Edinburgh (2005) and Madrid (2004).

The ICCC-2014 proceedings includes 38 technical papers (which were presented in 25 minute slots) and 13 late breaking papers (delivered in 10 minute slots). In addition, it also includes the abstract of the invited talk by Oliver Deussen from the University of Konstanz, with the title: Non-photorealistic Rendering Getting Physical.

In addition to the scientific programme, there was a rich accompanying programme (see http://computationalcreativity.net/iccc2014/) including an art exhibition entitled: You/Me/It, curated by Ian Gouldstone, a tutorial session with an introduction to Computational Creativity and a focus on engineering creative Twitterbots, presented by Geraint Wiggins and Tony Veale, and the launch of the book entitled "Hand-Made By Machines: An Illustrated Guide to Creativity in Humans and Computers" (robotcomix.com) by Tony Veale.

The conference was truly international, with authors of accepted papers coming from different countries including: Australia, Canada, Denmark, Finland, France, Germany, Greece, Indonesia, Ireland, Italy, Japan, Malta, Mexico, Norway, Poland, Portugal, Slovenia, Spain, the United Kingdom and the United States. The conference was attended by 90 registered participants, in addition to numerous remote participants of the conference via streaming on Twitch.

We are very grateful to the local organization committee at the Jožef Stefan Institute, including Senja Pollak, Mili Bauer, Dragana Miljković, Damjan Demšar and Tina Anžič, who worked tirelessly to bring together all of the strands of the conference and co-located events, and went beyond the call of duty to arrange for this to be a great conference. In addition, Tuula Juvonen in the Computational Creativity group at Goldsmiths College has worked hard to organize many aspects of the art exhibition, and we would like to thank her very much for this.

We would like to thank Oliver Deussen for presenting a keynote talk at the conference. We would similarly like to thank the artists at the exhibition for agreeing to join the event, with special thanks to the curator and organizer of the exhibition, Ian Gouldstone, who, along with colleagues Phoenix Fry and Laura Bushell, has done a great job under difficult time constraints to organize and document the first co-located art exhibition at the conference. Similar thanks go to Tony Veale and Geraint Wiggins for organizing the first tutorial session at the conference, and we hope this will be a feature of future events.

The conference would be nothing without the sterling efforts of the many authors who contributed papers containing great research, and we would like to thank both the authors whose papers were accepted, and those whose papers we were unfortunately not able to take. The reviewing process for this year's conference was very rigorous, with each paper getting three thorough reviews, equating to more than 60,000 words of feedback for the authors - an amazing statistic, which shows how engaged and encouraging the Computational Creativity community is.

We are extremely grateful to the programme committee members who undertook and organised these reviews: John Barnden, Oliver Bown, David C Brown, Nick Bryan-Kinns, Win Burleson, F. Amílcar Cardoso, John Gero, Pablo Gervás, Ashok Goel, Andrés Gómez de Silva Garza, Paulo Gomes, Jeremy Gow, Kazjon Grace, Amy Hoover, Anna Jordanous, Robert Keller, Ramon Lopez De Mantaras, Penousal Machado, Brian Magerko, Mary Lou Maher, Neil Maiden, Ruli Manurung, Jon McCormack, David C. Moffat, Nick Montfort, Diarmuid O'Donoghue, Francois Pachet, Philippe Pasquier, Alison Pease, Francisco Pereira, Rafael Pérez y Pérez, Mark Riedl, Graeme Ritchie, Rob Saunders, Gillian Smith, Ricardo Sosa, Oliviero Stock, Julian Togelius, Hannu Toivonen, Paulo Urbano, Lav Varshney, Tony Veale, Geraint Wiggins and Georgios Yannakakis.

In addition to the programme committee, we would like to pass on our thanks to the additional reviewers who devoted time and energy to the conference: Ricardo de Aldama, Ben Bogart, Charles Callaway, Fiammetta Ghedini, Carlos León, James Maxwell, Marco Marchini, Dragana Miljković, Hugo Gonçalo Oliveira, Senja Pollak and Jasmina Smailović. We would also like to thank the select band of people who helped out with reviewing the late breaking papers (not mentioned by name, to maintain anonymity).

As usual, the conference has been steered beautifully by the Association for Computational Creativity committee, to whom we are very grateful, with particular help from the organisers of the PROSECCO network. We are also very grateful for exposure of the conference from the Association for the Advancement of Artificial Intelligence. Finally, we would like to acknowledge with many thanks the financial support we received from the Jožef Stefan Institute, the EU FP7 programme via the PROSECCO network, the Office of Naval Research Global and the Engineering and Physical Sciences Research Council in the UK.

Simon Colton, Dan Ventura, Nada Lavrač and Michael Cook
ICCC-2014 Conference Chairs

Ljubljana, June 2014

Table of Contents

From Isolation to Involvement: Adapting Machine Creativity Software to Support Human-Computer Co-Creation

Anna Kantosalo, Jukka M. Toivanen, Ping Xiao, Hannu Toivonen
Department of Computer Science and Helsinki Institute for Information Technology HIIT
University of Helsinki, Finland
anna.kantosalo@helsinki.fi, jukka.toivanen@cs.helsinki.fi, ping.xiao@helsinki.fi, hannu.toivonen@cs.helsinki.fi

Abstract

This paper investigates how to transform machine creativity systems into interactive tools that support human-computer co-creation. We use three case studies to identify common issues in this transformation, under the perspective of User-Centered Design. We also analyse the interactivity and creative behavior of the three platforms in terms of Wiggins' formalization of creativity as a search. We arrive at the conclusion that adapting creative software for supporting human-computer co-creation requires redesigning some major aspects of the software, which guides our on-going project of building an interactive poetry composition tool.

Introduction

Machine creativity and *support for human creativity* are two complementary goals of computational creativity research. The role of the machine in supporting human creativity has been classified by Lubart (2005) into four categories: computer as a managment aid, computer as a communication enabler, computer as a creativity enhancer, and computer as a co-creator in the creative act. It is easy to see how advancements in machine creativity systems could support the role of the computer as a creativity enhancer, or even as a co-creator: A creative system in a certain domain, say poetry, could be used as a creative assistant for a human poet, producing draft poems that the poet could use as inspiration or raw material. This relationship could be taken even further to create a real partnership in which the computer and the user could take turns writing and editing a jointly authored poem.

Such co-creative systems have great potential for transforming the lives of professionals and laymen alike by increasing their creative potential. To aid the development of future co-creative systems and their integration to everyday lives of people, it is important to gather and analyse knowledge on the design and use of existing co-creative systems.

We use the term human-computer co-creation to refer to collaborative creativity where both the human and the computer take creative responsibility for the generation of a creative artefact. The term co-creation refers here to a social creativity process "leading to the emergence and sharing of creative activities and meaning in a socio-technical environment" (Fischer et al. 2005), but with the emphasis that the computer is, instead of only providing the socio-technical environment, also an active participant in the creative activities. This is similar to the definition of mixed-initiative co-creativity (MI-CC) by Yannakis et al. (2014), who define it as the creation of artefacts with the interaction of a human and a computational initiative. They note that the two participants do not need to contribute to the same degree, and we do not demand symmetric contributions from human-computer co-creative systems neither.

The focus of this paper is on investigating the design processes for human computer co-creation systems. More specifically we investigate the transformation of machine creativity methods into co-creative ones, i.e., from batch methods to human-computer co-creation. Our goal is to shed light on the design process, key design decisions, and various issues in such transformation projects. We look at the process from two directions: a user-centered perspective and a computational creativity perspective based on Wiggins' (2006) model.

We first give a brief introduction to user-centered design and a brief description of Wiggins' model of computational creativity. We then carry out an investigation of three systems described in the literature. We discuss the observations, and then reflect our findings by comparing them to our ongoing work to produce interactive, educational poetry writing software for children.

User-Centered Design Perspective to Human-Computer Co-Creation

We are interested in methodologies and tools for supporting human-computer co-creation. The design of computer support for creativity has been studied both in the fields of interaction design (e.g. Carroll and Latulipe (2009)) and computational creativity (e.g. Yeap et al. (2010)). Interaction design and especially user-centered design can provide us with a well defined design process and a selection of documented methods, which have been demonstrated useful in designing real-life interactive software. Therefore we adopt user-centered design as the methodological framework for examining the work presented in this paper.

User-centered design (UCD) can be considered as "the active involvement of users for a clear understanding of user and task requirements, iterative design and evaluation, and

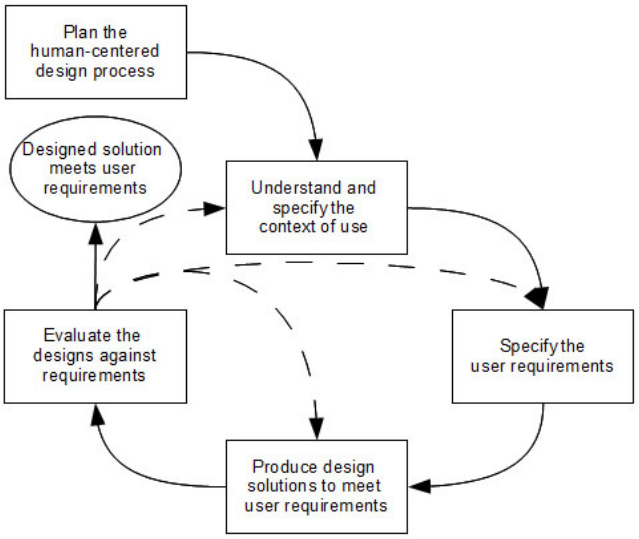


Figure 1: The user-centered design process as specified in (ISO/IEC 2010)

a multi-disciplinary approach" (Vredenburg et al. 2002). UCD methods have been developed since the 1980s and are today "generally considered to have improved product usefulness and usability" (Vredenburg et al. 2002). UCD can also be viewed more broadly as a part of Interaction Design — an umberella term covering multiple disciplines emphasising different design perspectives in and outside of Human Computer Interaction (Rogers, Sharp, and Preece 2011, p. 9-11).

The UCD process (ISO/IEC 2010) contains six steps (Figure 1): (1) Plan the human-centered design process, (2) Understand and specify the context of use, (3) Specify the user requirements, (4) Produce design solutions to meet user requirements, (5) Evaluate designs against requirements, and (6) Designed solution meets user requirements. Steps 2 to 5 form an iterative circle in which step 5 can be followed again by steps 2, 3, or 4 until the requirements have been satisfied as presented.

Methods in UCD vary in level of user involvement, need of resources and type of gathered data as well as in which part of the design process they are most commonly utilised. Some of the methods are developed specifically by human-computer interaction specialists, and some are used by other human-oriented fields such as antrophology, as well. Usually each UCD team chooses methods suitable for the study of their users in the set context according to their own resources and expertise. The most used methods include iterative design, usability evaluation and informal expert review (Vredenburg et al. 2002). Many more exist and we encourage the interested reader to consult a handbook.

A Search Perspective to Creativity

From a computational creativity perspective, we can study creative behaviour supported by software in the light of Wiggins' formalization of creativity as search (Wiggins 2006).

Wiggins' model attempts to clarify and formalize some concepts in Margaret Boden's (1992) descripive hierarchy of creativity. This model represents creative systems with a septuple $\langle \mathcal{U}, \mathcal{L}, [\![.]\!], \langle\!\langle ., ., . \rangle\!\rangle, \mathcal{R}, \mathcal{T}, \mathcal{E} \rangle$. Here Universe $\mathcal{U}$ refers to an abstract set of all possible artefacts, for instance poems. $\mathcal{R}$ refers to a set of rules, expressed in the language $\mathcal{L}$, which defines a subset of the universe $\mathcal{U}$ i.e. the conceptual space of the creative system in question. Traversal function $\mathcal{T}$ defines how search in the universe is performed and the evaluation function $\mathcal{E}$ assigns a value for (some) elements of the universe. This formalization allows describing exploratory creativity as search (primarily) in the conceptual space defined by $\mathcal{R}$ via traversal funtion $\mathcal{T}$ and evaluation function $\mathcal{E}$, whereas transformational creativity may be achieved, e.g., by modifying the rules $\mathcal{R}$ defining the conceptual space.

Wiggins' model provides one way to look at the co-creative process between the user and the computer and to study interaction in the process. For instance, issues arising from conflicts between the rules, evaluation functions, and traversal functions of the computer and the user can now be clearly described in Wiggins' formalism. The (transformative) actions the user and the computer take when such conflicts appear decide what the rules, evaluation function, and traversal function of the larger system consisting of both the computer and the user are.

It has to be noted that many other theories, for instance the work by Csikszentmihalyi (1997), could be used as a viewpoint to look at co-creativity. However, we selected Wiggins' model for its rigorous nature and popularity in the field of computational creativity.

Case Studies

In this section we review three case studies of interactive software supporting human-computer co-creation. We first describe the criteria used for selecting these systems and then proceed to give a brief overview of the systems. We then analyse these three systems, in terms of design processes, user interactions and changes to the underlying machine creativity methods, which provides suggestions for developing future co-creative systems.

Since there are few descriptions of the design processes of human-computer co-creative systems in literature, we have used somewhat loose criteria to select software for this study:

1. The project utilises established methods of computational creativity.
2. The end result of the project is interactive with a human user.
3. Design decisions taken in the project are described.
4. Quantitative or qualitative feedback is available for the interactive software.

The above criteria emphasize projects drawing influences from both disciplines, computational creativity and human-computer interaction. Based on the criteria, we selected three systems: STANDUP (Ritchie et al. 2007; Waller et al. 2009), Scuddle (Carlson, Schiphorst, and Pasquier 2011),

and Evolver (DiPaola et al. 2013). Our focus on the design process excludes some otherwise interesting examples of human-computer co-creative software, such as the Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013) and Tanagra (Smith, Whitehead, and Mateas 2011).

Overview of the selected systems STANDUP is a pun generating "language playground" developed for children with complex communication needs (CCN) (Ritchie et al. 2007; Waller et al. 2009). It is built on the basis of the JAPE system (Binsted 1996; Binsted and Ritchie 1997; 1994), which generates different classes of punning riddles using symbolic rules and a large, general purpose lexicon. The evaluation of the system with its target users suggested some restrictions in the capacity of the program but an increased facility with words and apparent enjoyment from its users (Waller et al. 2009). In addition, anecdotal evidence supported a positive effect on the communication of the children (Ritchie et al. 2007).

Scuddle is a movement exploration tool for choreographers to use in the early stages of their choreographic creation process (Carlson, Schiphorst, and Pasquier 2011). It is based on a genetic algorithm used to generate diverse combinations of movements. The evaluation of the program yelded positive results: users found the movements presented by the program non-habitual and creative and it prompted them to re-examine their own approaches to movement construction.

Evolver is a tool designed to help interior designers to explore design options based on the initial design elements provided by the designers themselves (DiPaola et al. 2013). Its focus is on helping the labor intensive early stages of a design project and offering novel designs outside the capabilities of its users. It is based on the autonomous creative genetic programming system called DarwinsGaze (DiPaola and Gabora 2009). Evolver was well received by its target audience who reported it supporting their creative processes, suggesting novel alternatives, easing manual work, and enabling communication. Interestingly some of the interior designers involved in the evaluation also considered the program as a collaborative partner in design instead of a mere platform.

All three systems show some established methods of computational creativity used as part of an interactive system. All systems have also been fairly successfull tools in increasing the creative potential of their users: STANDUP made the creative process of joke invention more accessible to an audience restricted by communication ability, Scuddle prompted new lines of creative inquiry in its users, and Evolver was at best considered a creative partner.

Interaction The level of user interaction is quite varied among the three cases. Of the three examples, Scuddle has the lowest level of interactivity. It provides the users only with simple options of starting or continuing the evolutionary algorithm, re-starting the whole process, or viewing six results evaluated by the computer (Carlson, Schiphorst, and Pasquier 2011). Describing these interaction options in Wiggins' framework, the theoretical categorisations of dance movements and their value can be seen as the conceptual space of the creative system. Traversal in the conceptual space is performed via a genetic algorithm which can be restarted or continued by the user evaluating the computer's pre-evaluated results. The user's role in the interaction lies more in the final evaluation of the artefacts than in the traversal of the options.

STANDUP has a higher level of interactivity than Scuddle. It offers a dual mode of interaction: user control can be divided into (1) options for the end user — a child with CCN, and (2) options for his or her carers. The child can choose a specific word to be included in the joke, a topic for the joke, or a specific joke type to be generated. The carer can adjust the program to suit the child best by restricting joke types, adjusting the words used in jokes based on their familiarity, or banning offensive words (Ritchie et al. 2007). In Wiggins' terms, the STANDUP user participates in defining the rules $\mathcal{R}$ in addition to participating in the transition function $\mathcal{T}$ and the evaluation function $\mathcal{E}$. On the other hand the computer provides the general conceptual space by defining the classes of puns and the allowed vocabulary. These can be modified by the user, i.e., the users' set of rules for conceptual space changes the respective set of rules of the computer. The traversal function of the computer is supervised by the user. The evaluation function of the computer makes sure that similar jokes have not been presented to the user before. The user makes the final evaluation and decides which of the jokes are saved.

Evolver provides the highest level of user interaction. The user provides the evolutionary algorithm with seed material and can select candidates to be used for generating the next generation of candidates as well as adjust the color scheme used (DiPaola et al. 2013). Viewed through Wiggins' framework, Evolver's interaction capabilities make the user's actions an integral part of the creative system: Evolver uses the seed material provided by the user to define the conceptual space. Traversal in this space is then performed via an evolutionary algorithm interactively with the user so that the user decides the parents for the next generation. The evaluation function of the co-creative system is a combination of the fitness function of the computer system and the final evaluation by the user.

Mapping the systems into Wiggins' model reveals that the human and the computer participating in the creative act can be viewed as one human-computer co-creative system. The mapping shows how both parties take responsibility over the generation of the creative artefact, although roles of the computer and the human are different. These particular examples also seem to indicate that the more interactive the system, the more integral the part of the user is in the creative model.

Design processes Carlson et al. (2011) started their design process for Scuddle by studying other computer aided choreographic systems and used the theory of choreography to establish requirements for Scuddle. They then proceeded to construct a prototype, which was tested with seven coreographers in simulated work sessions between a coreographer

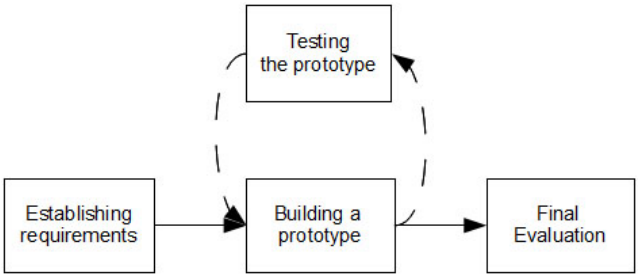


Figure 2: The design process of a co-creative tool described through the major design stages identified in the example projects

and a dancer. As evaluation methods they chose participant-observation and open ended interviews.

DiPaola et al. (2013) partnered with a design firm to develop Evolver. The design process started with establishing requirements by analysing the work processes of the employees of the partnering firm. The process continued with iterative prototyping and ended with a final evaluation conducted some months after the completion of the software.

Waller et al. (2009) relied on experts for gathering requirements for STANDUP. They continued iterative prototyping with the experts and adults with CCN and used typically developing children in testing graphics. The end product itself was evaluated with nine children with CCN during a ten week period including pre- and post-testing for the evaluation of learning effect, a training period for the children, and finally a scenario based observation of the users while using the software. The effects of the STANDUP software on the lives of the children beyond this period were studied with semistructured interviews and questionnaires directed at parents and other adults tightly involved with the children's learning progress.

All of the sample projects seem to follow a similar pattern in their design process (Figure 2). Each project starts by a requirement establishing stage and continues into prototype building. Two of the projects, Evolver and STANDUP continued this process iteratively by testing the prototype multiple times and adjusting it accordingly, while only one evaluation was conducted for Scuddle. The last iteration of this cycle can be called the final evaluation, a stage in which the final version of the prototype is evaluated more rigorosly, perhaps including assessment of usefullness or impact on the users.

When the process used in the studied cases is compared to the UCD process of Figure 1, we see that both processes share the stages of specifying requirements, producing solutions and evaluation. Both processes also have iterative properties, while the sample projects seem not to repeat the requirements setting stage. The stages of planning the process and understanding and specifying the context are missing from the case based description, but this may also be due to the result oriented reporting style of the papers, which may omit seemingly obvious details. Waller et al. (2009) report specifically having followed the UCD approach in designing STANDUP, and DiPaola et al. (2013) included researchers with a background in human-computer interaction in the design of Evolver.

Finally the processes differ in one important regard: If we categorise the processes by their starting points, Scuddle shows an example of applying a set of machine creativity methods directly into building interactive software, while Evolver and STANDUP both show an example of a process transforming existing autonomous creative systems into interactive products.

Changes to machine creativity methods To enable a higher level of interaction, the two projects using existing computational creativity prototypes had to conduct major changes in the machine creativity methods. These changes can be categorized into two rough categories: (1) changes done to facilitate interaction and (2) changes done to enhance the technical properties to better suit real-time use. The distinction between these classes can also be viewed through Wiggins' model. The first type of changes, driven by the goal of adding user interaction possibilities, increases the role of the user in Wiggins' model for the co-creative system, while the technical changes do not. However, the technical changes may support the quality of user interaction, which makes their categorisation without Wiggins' model difficult.

Ritchie et al. (2007) state that JAPE had multiple deficiencies which the STANDUP team had to account for by changing the system. The changes done to facilitate interaction in JAPE include keeping a record of jokes offered to a user to avoid too similar ones, the restriction of vocabulary to avoid obscene words and to focus on familiar ones, and possibilities to guide the search for jokes to a topic or specific words. The technical changes relate to adding better phonetic similarity measures and dropping some joke options to enhance the quality of jokes, as well as dropping some mechanisms to make the algorithm faster.

The DarwinsGaze algorithm underwent major changes in order to better suit the needs of Evolver's target audience as well (DiPaola et al. 2013). There is not as clear a distinction between interaction facilitating and technical changes on the surface, but viewed through Wiggins' model we see that giving the user control over the seed material and selection of candidates for pairing and adjusting the population both increase the user's role in the system. In addition, to emphasize gene linkage and user interpretability, the genetic algorithm was simplified by changing the gene structure to operate on a higher level of components called "design elements". The team also changed the internal format of pictures from bitmap to SVG to support layers in the generation and facilitate the import and export of pictures. Both of these modifications change the system in a way that can be seen in Wiggins' model. However, while the modifications increase the usability of the system, the user's role is not increased.

Building a Co-Creative Poetry Writer

We now move on to describe our on-going project developing an interactive poetry writing tool based on existing

poetry generation software.

We chose children in comprehensive education as our target user group, as they are learning to use language in creative ways and explore much of the similar structures such as rhyme and rhythm, which are addressed by the existing creative software. The following sections examine our process and compare it to the example cases.

Basis in Computational Creativity Methods The machine creativity elements in the interactive system under construction are based on the poetry generation work by Toivanen et al. (2012). This approach uses corpus-based methods to find associated words around a given topic word and then to write poetry about the topic by using these words to substitute words in a given piece of text. Poetic devices like rhyming and alliteration can be further controlled by using constraint-programming methods (Toivanen, Järvisalo, and Toivonen 2013). In addition to these approaches, the system includes methods which can provide poetic fragments in a certain meter (e.g. iambic pentameter) and contain certain words. These fragments have been automatically extracted from large masses of text and different combinations of them, possibly modified with the word substitution method, can be used as a building block of poetry writing.

Design Process After choosing school children as the target audience, we started establishing requirements by studying the users and the context. Restricted by time and targeting a very sensitive group of users, we decided, like Waller et al. (2009), to rely on indirect input from children in our early design phases and use their participation only in the evaluation.

We recruited five enthusiastic grammar school teachers to help us. They kindly allowed us to observe their classes. Four of the teachers were teaching a group of approximately 70 second grade students together. One teacher specialised in the Finnish language and literature, teaching multiple classes in the 7-9th grade. We observed one full day of education in the second grade classroom, as well as two ninth grade lessons. We focused on observing interactions between the teachers and the pupils, as well as between pupils and how they worked on creative writing tasks on computers. After the lessons we conducted semi-structured interviews with the teachers in charge. We also sent an internet based open-ended questionnaire on teaching materials to the teachers.

The observation revealed differences in the skills of children: Younger children were generally still honing their skills in basic writing, whereas older children were more focused on the subject matter. The younger children were also challenged by foreign language user interfaces but quick to learn by trying things out and learning from their neighbours. The observation also showed in real contexts the behavior and language used by the children when communicating peer-to-peer or with the teachers. This experience gave us inspiration for selecting suitable interaction metaphors — connections to real world situations or objects, which help designing insightful interfaces — as well as for reducing the level of complexity in the user interface of our application. — as well as We expanded our observations with a literature study on educational software, which revealed more suitable interaction patterns and methods. The interviews and questionnaires showed that teachers saw technology as a means to motivate and aid the learner. Some teachers, especially those working with younger children and children with special education needs, expressed a need for quality software to aid the learning of writing. In general, teachers emphasised poetry writing's role as a creative activity.

The interviews and observation indicated that the writing skills of children develop highly individually. Therefore our software needs to cater for writers capable of different levels of creative writing. We decided to develop a creative writing tool allowing for a varied level of computer assistance, to enable writers with different skillsets to try out poetry writing. We decided to use fridge magnets as a simple metaphor for the manipulation of text on screen. An interface for writing sentences using the magnet metaphor has previously been successfully developed by Kuhn et al. (2009).

To test the design, we developed a paper prototype which we evaluated with a specialist researching the use of information technology in education. Based on her feedback we simplified the interface further and revised some features in saving and exporting poetry. She also noted that more advanced writers would need more abstract topics for writing than those we offered in our paper prototype. We iterated the paper prototype development until both the specialist and we ourselves were confident in building a working prototype.

At the moment of writing this, we are completing the prototype implementation. Next, we will evaluate the prototype in two ways: (1) scenario-based evaluations with pairs of children in a laboratory setting and (2) testing in a classroom. The former is designed to catch the troubles children might have with the tool and in the latter we want to see how teachers manage a learning setting using the software.

The early decisions made about methodology and user involvement can be interpreted as the planning phase of our project viewed through the UCD process. The observation, interviews, questionnaires and the literature study conclude the second and the third phase of the UCD process, or they can be interpreted as the first stage of the general process seen in the examples. The paper prototyping shows some of the iterative prototyping of the general process, or one iterative cycle of the UCD process returning from phase five to phase three. Finally the planned evaluation fits the general process lifted from the examples very well, while also following the lines of the UCD process.

However there are some challenges to following the UCD process to the letter: we found it challenging to communicate the restrictions of the computational approach to our users for ideation. Similarly, we found that it is difficult to create extensive paper prototypes for testing with users in iterative prototyping. This is mainly because the use cases by definition involve creative input from the user, and it is hard to imitate quick responses to creative inputs. This reduces the feasibility to include users in the early stages of design.

Interaction In a typical use case, the user can give the computer inspirational keywords, around which the computer generates a few lines of draft poetry which the user can then start to modify and extend. This should help the user past the "blank page" stage. The user may additionally ask for more lines, or just new words for a specific place in the poem to help find suitable rhyming pairs for example. Any new fragments of text produced by the system adapt automatically to the modifications and additions done by the user. To enable more symmetric human-computer co-creation, we are also experimenting with different ways to show editing suggestions to the user.

From the perspective of Wiggins' model, the user and the software share the same universe $\mathcal{U}$ and language $\mathcal{L}$, and they produce a poem together by editing it in turns. Traversal in the conceptual space can thus be performed both by the computer (e.g. providing a line of poetry or proposals for rhyming words) and by the user (e.g. adding more text or changing existing words). They both aim to satisfy (or modify and satisfy) their own rules $\mathcal{R}$ and evaluation function $\mathcal{E}$. This shows that our system can also be interpreted as a co-creative system with the user and the computer both sharing responsibility over the creative artefact.

Changes into the machine creativity methods The methods by Toivanen et al. (2012; 2013) were designed to compose poetry autonomously and certain changes were needed to modify them to work in an interactive system.

The interactive poetry writing process supports turns of word substitution and moving by the user and the computer. The grammar template needs to be updated when the user moves the words around.

The user may ask for suggestions for certain words and here the constraint-based methods need to be modified so that they can provide, for instance, suggestions for rhyming or alliterating words which satisfy some additional constraints like having a certain part-of-speech and grammatical case. Finally, the computer also needs to be able to update its vocabulary and keep record of the changes made by the user.

In Wiggins' terms, the rules $\mathcal{R}$ and the transition function $\mathcal{T}$ are defined in collaboration by the user and the computer as they both can change the contents of the poem. On the other hand, the evaluation is mainly done by the user as the computer evaluates only some things such as metric structure, and the final evaluation is always done by the user.

Conclusions

We have looked at the re-design of machine creativity methods into interactive human-computer co-creation tools. Based on the small sample of design processes that we studied, UCD methods seem to be common in creating interactive software on the basis of machine creativity methods. All of the cases we studied follow a similar process that can be viewed as an instance of the UCD process. However, the principles of user involvement, iterative design and a multidisciplinary approach are fulfilled to different extent in each project.

Computational creativity methods also set some boundaries for the software to be designed. However, as two of the case studies and our own project show, the methods can be re-negotiated for interactivity, transformationally changing the boundaries of interaction. This again can permit new designs making the re-negotiation process also iterative.

When characterised in Wiggins' framework, the observations are that for a high level of interactivity the re-negotiation of the methods must include interaction facilitating changes, which give the user a larger role in the system, and that only usability factors can be enhanced without expanding the role of the user in the re-negotiation. However, our sample is small and the search for other ways to increase interactivity demands further research.

The re-negotiation of computational creativity methods and the role of the user in them is an important part of defining the nature of creative interaction in the software. The design choices taken in the re-negotiation further define the extent to which we can achieve human-computer co-creation. These design choices may include questions such as whether the interactions are always human initiated, or if the computer may also spontaneously offer new creative perspectives, whether the interaction is done by exchanging creative artefacts, is instruction oriented, or is carried out in a more conversational manner creating a socio-technical environment resembling that of human-human co-creation.

UCD is focused on the human user. However, if we want to create more balanced human-computer co-creation, we may also need to account for the input the computer needs from the user to be able to participate in the process more extensively. Thus, it might be useful to look into collaborative creativity tools and remote presence to see if the computer can take a role similar to another human being as a creative collaborator. The roles of the user and the computer in co-creation should also be connected to the roles considered by Maher (2012).

Finally, interesting insight into human-computer co-creation could be gained by using Wiggins' framework to characterise interactions and their effects. Assume that the human and computer agents both apply their own traversal functions $\mathcal{T}$ on a shared (partial) artefact, based on their own rules $\mathcal{R}$ and evaluations $\mathcal{E}$. This can result, for instance, (1) in immediate synergy, such as reaching good areas in the search space that neither one can reach alone ("increasing generative inspiration"), (2) in pressure/possibility for transformational creativity (e.g. "productive aberration"), as well as (3) in conflicts where one agent takes the search into an area where the other one is not able to operate in a meaningful way ("generative uninspiration"). An analysis of such cases could provide guidance for issues that one should be able to deal with in human-computer co-creation.

Acknowledgments

This work has been supported by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland and the Helsinki Doctoral Program in Computer Science (HECSE).

References

Binsted, K., and Ritchie, G. 1994. An implemented model of punning riddles. In *AAAI*, 633–638.

Binsted, K., and Ritchie, G. 1997. Computational rules for generating punning riddles. *Humor* 10(1):25–76.

Binsted, K. 1996. *Machine humour: An implemented model of puns*. Ph.D. Dissertation, University of Edinburgh, Edinburgh, Scotland.

Boden, M. 1992. *The Creative Mind*. London: Abacus.

Carlson, K.; Schiphorst, T.; and Pasquier, P. 2011. Scuddle: Generating movement catalysts for computer-aided choreography. In *Proceedings of the Second International Conference on Computational Creativity*.

Carroll, E. A., and Latulipe, C. 2009. The creativity support index. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, 4009–4014. New York, NY, USA: ACM.

Csikszentmihalyi, M. 1997. Flow and the psychology of discovery and invention. *HarperPerennial, New York*.

DiPaola, S., and Gabora, L. 2009. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming and Evolvable Machines* 10(2):97–110.

DiPaola, S.; McCaig, G.; Carlson, K.; Salevati, S.; and Sorenson, N. 2013. Adaptation of an autonomous creative evolutionary system for real-world design application based on creative cognition. In *Proceedings of the Fourth International Conference on Computational Creativity*, 40.

Fischer, G.; Giaccardi, E.; Eden, H.; Sugimoto, M.; and Ye, Y. 2005. Beyond binary choices: Integrating individual and social creativity. *International Journal of Human-Computer Studies* 63(4):482–512.

ISO/IEC. 2010. Iso 9241-210 ergonomics of human-system interaction – part 210: Human-centered design for interactive systems.

Kuhn, A.; Quintana, C.; and Soloway, E. 2009. Storytime: a new way for children to write. In *Proceedings of the 8th International Conference on Interaction Design and Children*, IDC '09, 218–221. New York, NY, USA: ACM.

Liapis, A.; Yannakakis, G.; and Togelius, J. 2013. Sentient sketchbook: Computer-aided game level authoring. In *Proceedings of ACM Conference on Foundations of Digital Games*.

Lubart, T. 2005. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies* 63(4):365–369.

Maher, M. L. 2012. Computational and collective creativity: Whos being creative?. In *Proc. 3rd Int. Conf. on Computational Creativity*.

Ritchie, G.; Manurung, R.; Pain, H.; Waller, A.; Black, R.; and OMara, D. 2007. A practical application of computational humour. In *Proceedings of the 4th International Joint Conference on Computational Creativity*, 91–98.

Rogers, Y.; Sharp, H.; and Preece, J. 2011. *Interaction Design: Beyond Human Computer Interaction.* Wiley, 3rd edition edition.

Smith, G.; Whitehead, J.; and Mateas, M. 2011. Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *Computational Intelligence and AI in Games, IEEE Transactions on* 3(3):201–215.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *International Conference on Computational Creativity*, 175–179.

Toivanen, J. M.; Järvisalo, M.; and Toivonen, H. 2013. Harnessing constraint programming for poetry composition. In *International Conference on Computational Creativity*, 160–167.

Vredenburg, K.; Mao, J.-Y.; Smith, P. W.; and Carey, T. 2002. A survey of user-centered design practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, 471–478. New York, NY, USA: ACM.

Waller, A.; Black, R.; O'Mara, D. A.; Pain, H.; Ritchie, G.; and Manurung, R. 2009. Evaluating the standup pun generating software with children with cerebral palsy. *ACM Trans.Access.Comput.* 1(3):16:1–16:27.

Wiggins, G. A. 2006. Searching for computational creativity. *New Generation Computing* 24(3):209–222.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity. In *Proceedings of the ACM Conference on Foundations of Digital Games*.

Yeap, W. K.; Opas, T.; and Mahyar, N. 2010. On two desiderata for creativity support tools. In *Proc. of the Intl. Conference on Computational Creativity*, 180–189.

The Social Impact of Self-Regulated Creativity on the Evolution of Simple versus Complex Creative Ideas

Liane Gabora
University of British Columbia
Department of Psychology, Okanagan campus, Arts Building, 3333 University Way
Kelowna BC, V1V 1V7, CANADA
liane.gabora@ubc.ca

Simon Tseng
University of British Columbia
Department of Engineering, 5000-2332 Main Mall
Vancouver BC,V6T 1Z4, CANADA
s.tseng@alumni.ubc.ca

Abstract

Since creative individuals invest in unproven ideas at the expense of propagating proven ones, excess creativity can be detrimental to society; moreover, some individuals benefit from creativity without being creative themselves by copying creators. This paper builds on previous studies of how societies evolve faster by tempering the novelty-generating effects of creativity with the novelty-preserving effects of imitation. It was hypothesized that (1) this balance can be achieved through self-regulation (SR) of creativity, by varying how creative one is according to the value of one' creative outputs, and (2) that the social benefit of SR is affected by the openness of the space of possible ideas. These hypotheses were tested using EVOC, an agent-based model of cultural evolution in which each agent self-regulated its invention-to-imitation ratio as a function of the fitness of its inventions. We compared SR to non-SR societies, and compared societies in which the space of possible ideas was open-ended because agents could chain simple ideas into complex ones, to societies without chaining, for which the space of possible ideas was fixed. Agents in SR societies gradually segregated into creators and imitators, and changes in diversity were rapider and more pronounced than non-SR. The mean fitness of ideas was higher in SR than non-SR societies, but this difference was temporary without chaining whereas it was permanent with chaining. We discuss limitations of the model and possible social implications of the results.

Keywords: Agent-based model; creativity; imitation; individual differences; self regulation; cultural evolution EVOC.

Introduction

It is commonly assumed that creativity is desirable, and the more creative one is, the better. Our capacity for self-expression, problem solving, and making aesthetically pleasing artifacts, all stem from our creative abilities. However, individuals often claim that their creativity is stifled by social norms, policies, and institutions. Moreover, our educational systems do not appear to prioritize the cultivation of creativity, and in some ways discourage it.

Perhaps there is an adaptive value to these seemingly mixed messages that society sends about the social desirability of creativity. Perhaps what is best for society is that individuals vary widely with respect to how creative they are, so as to ensure that the society as a whole both generates novel variants, and preserves the best of them. This paper provides a computational test of the following hypotheses. The first hypothesis is that society as a whole benefits when individuals can vary how creative they are in response to the perceived effectiveness of their ideas. In theory, if effective creators create more, and ineffective creators create less, the ideas held by society should collectively evolve faster. The second hypothesis is that the space of possible ideas has to be open-ended in order to benefit from this self-regulation mechanism. In theory, the effectiveness of such a self-regulation should vary with the extent to which some ideas are fitter or more effective than others.

Definition and Key Features of Creativity

There are a plethora of definitions of creativity in the literature; nevertheless, it is commonly accepted that a core characteristic of creativity is the production of an idea or product that meets two criteria: originality or *novelty*, and appropriateness, adaptiveness, or *usefulness, i.e.,* relevance to the task at hand (Guilford 1950; Moran 2011). Not only are humans individually creative, but we build on each other's ideas such that over centuries, art, science, and technology, as well as customs and folk knowledge, can be said to evolve. This cumulative building of new innovations on existing products is sometimes referred to as the ratchet effect (Tomasello, Kruger, and Ratner 1993). Creativity has long been associated with personal fulfillment (May 1975; Rogers 1959), self-actualization (Maslow 1959), and maintaining a competitive edge in the marketplace. Thus it is often assumed that more creativity is necessarily better.

However, there are significant drawbacks to creativity

(Cropley et al. 2010; Ludwig 1995). Generating creative ideas is difficult and time consuming, and a creative solution to one problem often generates other problems, or has unexpected negative side effects that may only become apparent after much effort has been invested. Creativity is correlated with rule bending, law breaking, and social unrest (Sternberg and Lubart 1995; Sulloway 1996), aggression (Tacher and Readdick 2006), group conflict (Troyer and Youngreen 2009), and dishonesty (Gino and Ariely 2012). Creative individuals are more likely to be viewed as aloof, arrogant, competitive, hostile, independent, introverted, lacking in warmth, nonconformist, norm doubting, unconscientious, unfriendly (Batey and Furnham 2006; Qian, Plucker, and Shen 2010; Treffinger et al. 2002). They tend to be more emotionally unstable, and more prone to affective disorders such as depression and bipolar disorder, and have a higher incidence of schizophrenic tendencies, than other segments of the population (Andreason 1987; Eysenck 1993; Flaherty 2005). They are also more prone to drug and alcohol abuse, as well as suicide (Jamison 1993; Goodwin 1998; Rothenberg 1990; Kaufman 2003). This suggests that there is a cost to creativity, both to the individual and to society.

Balancing Novelty with Continuity

Given the correlation between creativity and personality traits that are potentially socially disruptive, it is perhaps fortunate that in a group of interacting individuals, not all of them need be particularly creative for the benefits of creativity to be felt throughout the group. The rest can reap the rewards of the creator's ideas by copying them, buying from them, or simply admiring them. Few of us know how to build a computer, or write a symphony, but they are nonetheless ours to use and enjoy. Of course, if everyone relied on the strategy of imitating others rather than coming up with their own ideas, the generation of cultural novelty would grind to a halt. On the other hand, if everyone were as creative as the most creative amongst us, the frequency of the above-mentioned antisocial tendencies of creative people might be sufficiently high to interfere with cultural stability; *i.e.,* the perpetuation of cultural continuity. It is well known in theoretical biology that both novelty and continuity are essential for evolution, that is, for cumulative, open-ended, adaptive change over time.

This need for both novelty and continuity was demonstrated in an agent-based model of cultural evolution (Gabora 1995). Novelty was injected into the artificial society through the invention of new actions, and continuity was preserved through the imitation of existing actions. When agents never invented, there was nothing to imitate, and there was no cultural evolution at all. If the ratio of invention to imitation was even marginally greater than 0, not only was cumulative cultural evolution possible, but eventually all agents converged on optimal cultural outputs. When all agents always invented and never imitated, the mean fitness of cultural outputs was also sub-optimal because fit ideas were not dispersing through society. The society as a whole performed optimally when the ratio of creating to imitating was approximately 2:1. Although results obtained with a simple computer model may have little bearing on complex human societies, the finding that extremely high levels of creativity can be detrimental to the society suggests that there may be an adaptive value to society's ambivalent attitude toward creativity.

This suggested that society as a whole might benefit from a distinction between the conventional workforce and what has been called a "creative class" (Florida 2002) This was investigated in the model by introducing two types of agents: imitators, that only obtained new actions by imitating neighbors, and creators, that obtained new actions either by inventing or imitating (Gabora and Firouzi 2012). It was possible to vary the probability that creators create versus imitate; thus, whereas a given agent was either a creator or an imitator throughout the entire run, the proportion of creators innovating or imitating in a given iteration fluctuated stochastically. The mean fitness of ideas across the artificial society was highest when not all agents were creators. Specifically, there was a tradeoff between C, the proportion of creators to imitators in the society, and p, how creative the creators were). This provided further support for the hypothesis that society as a whole functions optimally when creativity is tempered with continuity.

We then hypothesized that society as a whole might perform even better if individuals are able to adjust how creative they are over time in accordance with their perceived creative success. For example, this could result from mechanisms such as selective ostracization of deviant behaviour unless accompanied by the generation of valuable novelty, and encouragement or even adulation of those whose creations are successful. In this way society might self-organize into a balanced mix of novelty generating creators and continuity perpetuating imitators, both of which are necessary for cumulative cultural evolution. A first step in investigating this hypothesis was to determine whether it is algorithmically possible to increase the mean fitness of ideas in a society by enabling them to self-regulate how creative they are, and investigate the conditions under which this is possible.

The Computational Model

We investigated this using an agent-based model of cultural evolution referred to as "EVOlution of Culture", abbreviated EVOC (Gabora 2008)[1]. It uses neural network based agents that (1) invent new ideas, (2) imitate actions implemented by neighbors, (3) evaluate ideas, and (4) implement successful ideas as actions. EVOC is an elaboration of Meme and Variations, or MAV (Gabora 1995), the earliest computer program to model culture as an evolutionary process in its own right, as opposed to modeling the interplay of cultural and biological evolution[2]. The goal behind MAV, and also behind EVOC, was to distil the underlying logic of cultural

[1]The code is freely available; to gain access please contact the first author by email at liane.gabora@ubc.ca.

[2]The approach can thus be contrasted with computer models of how individual learning affects biological evolution (Best 1999; Higgs 1992; Hinton and Nowlan 1992; Hutchins and Hazelhurst 1991).

evolution, *i.e.,* the process by which ideas adapt and build on one another in the minds of interacting individuals. Agents do not evolve in a biological sense, as they neither die nor have offspring, but do in a cultural sense, by generating and sharing ideas for actions. In cultural evolution, the generation of novelty takes place through invention . EVOC was originally developed to compare and contrast the processes of biological and cultural evolution, but has subsequently been used to address such questions as how does the presence of leaders or barriers to the diffusion of ideas affect cultural evolution.

We now summarize the architecture of EVOC in sufficient detail to explain our results; for further details we refer the reader to previous publications (Gabora 2008; Leijnen and Gabora 2009).

Agents

Agents consist of (1) a neural network, which encodes ideas for actions and detects trends in what constitutes a fit action, (2) a 'perceptual system', which observes and evaluates neighbours' actions, and (3) a body, consisting of six body parts which implement actions.

The neural network is composed of six input nodes and six corresponding output nodes that represent concepts of body parts (LEFT ARM, RIGHT ARM, LEFT LEG, RIGHT LEG, HEAD, and HIPS), and seven hidden nodes that represent more abstract concepts (LEFT, RIGHT, ARM, LEG, SYMMETRY, OPPOSITE, and MOVEMENT). Input nodes and output nodes are connected to hidden nodes of which they are instances (*e.g.,* RIGHT ARM is connected to RIGHT.) Each body part can occupy one of three possible positions: a neutral or default position, and two other positions, which are referred to as active positions. Activation of any input node activates the MOVEMENT hidden node. Same-direction activation of symmetrical input nodes (*e.g.,* positive activation – which represents upward motion – of both arms) activates the SYMMETRY node. The entire reason for the neural network is to enable agents to learn trends over time concerning what general types of actions tend to be valuable, and use this learning to invent new actions more effectively. Without the neural network agents invent at random and the fitness of their inventions increases much more slowly (Gabora, 2008).

Invention

An idea for a new action is a pattern consisting of six elements that dictate the placement of the six body parts. Agents generate new actions by modifying their initial action or an action that has been invented previously or acquired through imitation. During invention, the pattern of activation on the output nodes is fed back to the input nodes, and invention is biased according to the activations of the SYMMETRY and MOVEMENT hidden nodes. We emphasize that were this not the case there would be no benefit to using a neural network. To invent a new idea, for each node of the idea currently represented on the input layer of the neural network, the agent makes a probabilistic decision as to whether the position of that body part will change, and if it does, the direction of change is stochastically biased according to the learning rate. If the new idea has a higher fitness than the currently implemented idea, the agent learns and implements the action specified by that idea. When "chaining" is turned on, an agent can keep adding new sub-actions and thereby execute a multi-step action, so long as the most recently-added sub-action is both an optimal sub-action and different from the previous sub-action of that action (Gabora, Chia, and Firouzi 2013).

Imitation

The process of finding a neighbour to imitate works through a form of lazy (non-greedy) search. The imitating agent randomly scans its neighbours, and adopts the first action that is fitter than the action it is currently implementing. If it does not find a neighbour that is executing a fitter action than its own current action, it continues to execute the current action.

Evaluation: The Fitness Function

Following (Holland 1975), we refer to the success of an action in the artificial world as its *fitness*, with the caveat that unlike its usage in biology, here the term is unrelated to number of offspring (or ideas derived from a given idea). The fitness function used in these experiments rewards activity of all body parts except for the head, symmetrical limb movement, and positive limb movement. Fitness of a single-step action F_n is determined as per Eq. 1. Total body movement, m, is calculated by adding the number of active body parts, *i.e.,* body parts not in the neutral position.

$$F_n = m + 5(s_a + s_t) + 2(p_a + p_t) + 10 * a_h + 2 * a_p \quad (1)$$

$s_a = 1$ if arms move symmetrically; 0 otherwise
$s_t = 1$ if legs move symmetrically; 0 otherwise
$p_a = 1$ if both arms move upwards; 0 otherwise
$p_t = 1$ if both legs move upwards; 0 otherwise
$a_h = 1$ if head is stationary; 0 otherwise
$a_p =$number of body parts moving upwards;

Note that there are multiple optima. (For example an action can be optimal if either both arms move up or if both arms move down.) The fitness F_c of a multi-step action with n chained single-step actions (each with fitness F_n) is calculated by Eq. 2.

$$F_c = \sum_{k=1}^{n} \frac{F_n}{1.2^{n-1}} \quad (2)$$

Learning

Invention makes use of the ability to detect, learn, and respond adaptively to trends. Since no action acquired through imitation or invention is implemented unless it is fitter than the current action, new actions provide valuable information about what constitutes an effective idea. Knowledge acquired through the evaluation of actions is translated into educated guesses about what constitutes a successful action by updating the learning rate. For example, an agent may

learn that more overall movement tends to be either beneficial (as with the fitness function used here) or detrimental, or that symmetrical movement tends to be either beneficial (as with the fitness function used here) or detrimental, and bias the generation of new actions accordingly.

The Artificial World

These experiments used a default artificial world: a toroidal lattice with 1024 cells each occupied by a single, stationary agent, and a von Neumann neighborhood structure. Creators and imitators were randomly dispersed.

A Typical Run

Fitness and diversity of actions are initially low because all agents are initially immobile, implementing the same action, with all body parts in the neutral position. Soon some agent invents an action that has a higher fitness than immobility, and this action gets imitated, so fitness increases. Fitness increases further as other ideas get invented, assessed, implemented as actions, and spread through imitation. The diversity of actions increases as agents explore the space of possible actions, and then decreases as agents hone in on the fittest actions. Thus, over successive rounds of invention and imitation, the agents' actions improve. EVOC thereby models how "descent with modification" occurs in a purely cultural context.

Method

To test the hypothesis that the mean fitness of cultural outputs across society increases faster with social regulation (SR) than without it, we increased the relative frequency of invention for agents that generated superior ideas, and decreased it for agents that generated inferior ideas. To implement this the computer code was modified as follows. Each iteration, for each agent, the fitness of its current action relative to the mean fitness of actions for all agents at the previous iteration was assessed. Thus we obtained the relative fitness (RF) of its cultural output. The agent's personal probability of creating, $p(C)$, was a function of RF. It was calculated as follows:

$$p(C)_n = \begin{cases} 1, & \text{if } p(C)_{n-1} \times RF_{n-1} > 1 \\ p(C)_{n-1} \times RF_{n-1}, & \text{otherwise} \end{cases} \quad (3)$$

The probability of imitating, $p(I)$, was 1 - $p(C)$. Thus when SR was on, if relative fitness was high, the agent invented more, and if it was low the agent imitated more. $p(C)$ was initialized at 0.5 for both SR and non-SR societies. We compared runs with SR to runs without it, both with and without the capacity to chain simple ideas into more complex ones.

Results

All data are averages across 250 runs. We first present the results of experiments in which chaining was turned off and thus only simple inventions were possible. Second we present the results of experiments with chaining turned on such that simple ideas could be combined into increasingly complex inventions.

The Effect of Social Regulation with No Chaining

With chaining turned off, the mean fitness of the cultural outputs of societies with SR (the ability to self-regulate inventiveness as a function of inventive success) was higher than that of societies without SR, as shown in Figure 1. However, the difference between SR and non-SR societies is only temporary; it lasts for the duration that the space of possible ideas in being explored. In both SR and non-SR societies mean fitness of actions plateaued when all agents converged on optimally fit ideas. Thus the value of segregating into creators and imitators is short-lived.

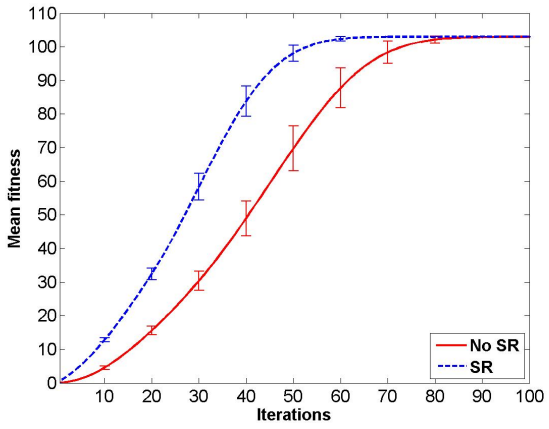


Figure 1: This graph plots the mean fitness of implemented actions across all agents over the duration of the run without chaining, with and without social regulation.

The diversity, or number of different ideas, exhibited an increase as the space of possibilities is explored followed by a decrease as agents converge on fit actions, as shown in Figure 2.

This pattern is typical in evolutionary scenarios where outputs vary in fitness. What is of particular interest here is that this pattern occurred earlier, and was more pronounced, in societies with SR than in societies without it Inferior creators were evidently inventing the same ideas so decreasing their creativity had little effect on diversity. On the other hand, superior creators were diverging variety of different directions, so making them more creative did increase diversity.

As illustrated in Figure 3, in societies with SR, while all agents initially invented and imitated with equal frequency, encouraging effective creators to create and discouraging ineffective creators did eventually cause them to segregate into two distinct groups: one that invented, and one that imitated. Thus whereas any point along the pareto frontier was optimal behaviour from an individual standpoint, they all piled up at the extreme ends, and the society as a whole benefited from this division of labour.

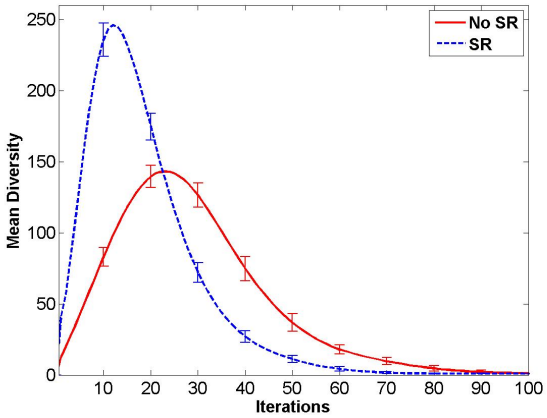


Figure 2: This graph plots the mean diversity of implemented actions across all agents over the duration of the run without chaining, with and without social regulation.

Thus the observed increase in fitness can indeed be attributed to increasingly pronounced individual differences in degree of creativity over the course of a run; agents that generated superior cultural outputs had more opportunity to do so, while agents that generated inferior cultural outputs became more likely to propagate proven effective ideas rather than reinvent the wheel.

The Effect of Social Regulation with Chaining

With chaining turned on, cultural outputs got increasingly fitter over the course of a run, as shown in Figure 4. This is because a fit action could always be made fitter by adding another sub-action. Note that with chaining turned on, although the number of different actions decreases, the agents do not converge on a static set of actions; the set of implemented actions changes continuously as they find new, fitter actions.

As was the case without chaining, the diversity of ideas with chaining turned on exhibited an increase as the space of possibilities is explored followed by a decrease as agents converge on fit actions, and once again this pattern was more pronounced in societies with SR than in societies without it, as shown in Figure 5. Interestingly, however, diversity no longer peaks later for non-SR than SR. Because with the capacity to chain simple ideas into increasingly complex ideas, the pool of possible ideas is now unconstrained, it no longer makes sense to converge quickly on optimal ideas. Indeed, there no longer is a fixed set of optimal ideas.

As was the case in the experiments without chaining, societies with SR ended up separating into two distinct groups: one that primarily invented, and one that primarily imitated.

Discussion

The goal of this paper was not to develop a realistic model of creativity but to investigate whether, with respect to creativity, can there be too much of a good thing. Are the needs

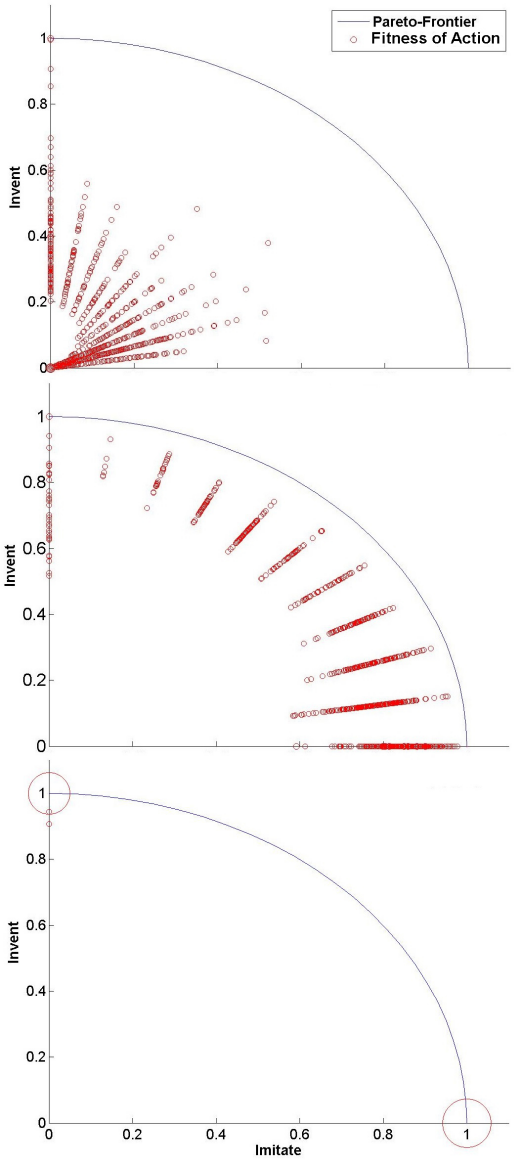


Figure 3: This graph plots the fitness of actions obtained through invention on the y axis and through imitation on the x axis. Fitness values are given as a proportion of the fitness of an optimally fit action. The pareto frontier indicates the range of possible ways an agent can behave optimally, either by always inventing optimally (upper left corner) or always implementing an optimal action obtained by imitating a neighbour (bottom right corner) or by implementing optimal actions obtained through some combination of inventing and imitating (all other points along the curve). Each small red circle shows the mean fitness of an agent's actions obtained through invention and imitation averaged across ten iterations: iterations 1 to 10 in the top graph, 25 to 35 in the middle graph, and 90 to 100 in the bottom graph. Since by iteration 90 all values were piled up in two spots – the upper left and the bottom right – they are indicated by large red circles at these locations.

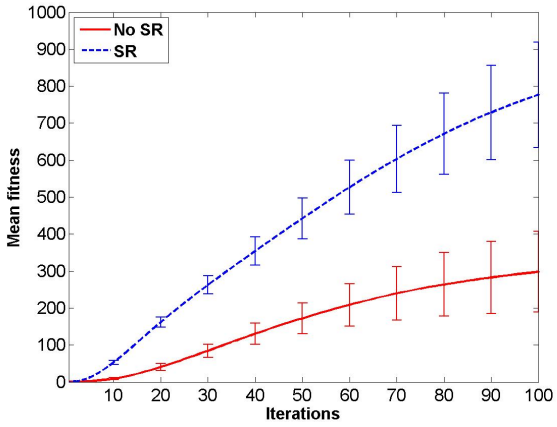


Figure 4: This graph plots the mean fitness of implemented actions across all agents over the duration of the run with Chaining turned on, with and without social regulation.

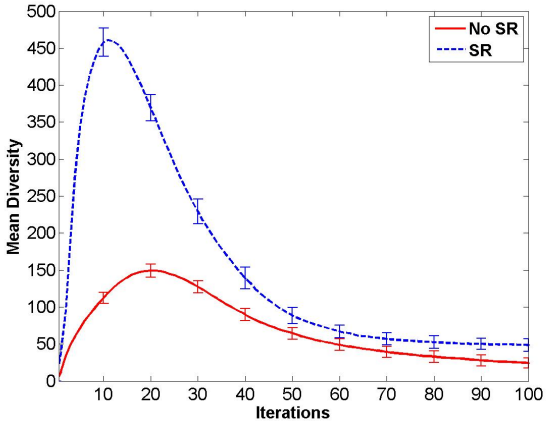


Figure 5: This graph plots the mean diversity of implemented actions across all agents over the duration of the run with Chaining, with and without social regulation.

of the individual for creative expression at odds with society's need to reinforce conventions and established protocols? EVOC agents are too rudimentary to suffer the affective penalties of creativity but the model incorporates another drawback to creativity: time spent inventing is time not spent imitating. Because creative agents spend their time inventing new ideas at the expense of social learning of proven ideas, effectively rupture the fabric of the artificial society; they act as insulators that impede the diffusion of proven solutions. Imitators, in contrast, serve as a "cultural memory" that ensures the preservation of successful ideas. When effective inventors created more and poor inventors created less, the society as a whole could capitalize on the creative abilities of the best inventors and capitalize on efforts of the rest to disseminate fit cultural outputs. This effect was temporary when agents were limited to a finite set of simple ideas; in other words, when the set of possible ideas was finite, the benefits of self-regulated creativity were short-lived. However, when agents were able to chain simple ideas into complex ideas and thus the space of possible ideas was open-ended, the benefits of self-regulation of creativity increased throughout the duration of a run. The results suggest that it can be beneficial for a social group if individuals are allowed to follow different developmental trajectories in accordance with their demonstrated successes, but only if the space of possible ideas is open-ended enough that there are always avenues for new creative ideas to explore.

It has been suggested that the capacity to chain together ideas for simple actions to generate ideas for complex actions such that the space of possible ideas was open-ended emerged some 1.7 million years ago, around the time of the transition from *Homo habilis* to *Homo erectus* (Donald 1991). This hypothesis is supported by mathematical (Gabora and Aerts 2009; Gabora and Kitto 2013) and computational (Gabora and Saberi 2011; Gabora and DiPaola 2012; Gabora, Chia, and Firouzi 2013) modelling. The fact that self-regulation of creativity was only found to be of lasting value in societies composed of agents capable of chaining suggests that there may have been insufficient selective pressure for self-regulation of creativity before this. Thus, prior to this time there would have been little individual variation across individuals in a social group with pronounced individual differences in creativity emerging after this time.

These results do not prove that in real societies successful creators invent more and unsuccessful creators invent less; they merely show this kind of self-regulation is a feasible means of increasing the mean fitness of creative outputs. However, the fact that strong individual differences in creativity exist (Kaufman 2003; Wolfradt and Pretz 2001) suggests that this occurs in real societies. Whether prompted by individuals themselves or mediated by way of social cues, families, organizations, or societies may spontaneously self-organize to achieve a balance between creative processes that generate innovations and the imitative processes that disseminate these innovations. In other words, they evolve faster by tempering novelty with continuity. A more complex version of this scheme is that individuals find a task at which they excel, such that for each task domain there exists some individual in the social group who comes to be best equipped to explore that space of possibilities.

The social practice of discouraging creativity until the individual has proven him- or herself may serve as a form of social self-regulation ensuring that creative efforts are not squandered. Individuals who are tuned to social norms and expectations may over time become increasingly concerned with imitating and cooperating with others in a manner that promotes cultural continuity. Their thoughts travel more well-worn routes, and they are increasingly less likely to innovate. Others might be tuned to the demands of creative tasks, and less tethered to social norms and expectations, and thereby more likely to see things from unconventional perspectives. Thus they are more likely to come up with solutions to problems or unexpected challenges, find new avenues for self-expression, and contribute to the generation of cultural novelty. In other words, what Cropley *et al.* (2010) refer to as the "dark side of creativity" may reflect that the creative individual is tuned to task needs at expense

of human needs. Although in the long run this benefits the group as a whole because it results in creative outputs, in the short run the creative individual may be less likely to obey social norms and live up to social expectations, and to experience stigmatization or discrimination as a result, particularly in his/her early years (Craft 2005; Scott 1999; Torrance 1963). Once the merits of such individuals' creative efforts become known, they may be supported or even idolized.

Limitations of this work include that the fitness function was static throughout a run, and agents had only one action to optimize. In real life, there are many tasks, and a division of labor such that each agent specializes in a few tasks, and imitates other agents to carry out other tasks. It may be that no one individual is an across-the-board "creator" or "imitator" but that different individuals find different niches for domain-specific creative outputs.

Another limitation is that currently EVOC does not allow an agent to imitate some features of an idea and not others. This would be useful because cultural outputs both in EVOC and the real world exhibit a version of what in biology is referred to as epistasis, wherein what is optimal with respect to one component depends on what is going on with respect to another. Once both components have been optimized in a mutually beneficial way (in EVOC, for example, symmetrical arm movement), excess creativity risks breaking up co-adapted partial solutions. In future studies we will investigate the effects of enabling partial imitation.

Acknowledgments

This work was supported by grants from the Natural Sciences and Engineering Research Council of Canada and the Flemish Fund for Scientific Research, Belgium.

References

Andreason, N. 1987. Creativity and mental illness. prevalence rates in writers and their first degree relatives. *American Journal of Psychiatry* 144:1288–1292.

Batey, M., and Furnham, A. 2006. Creativity, intelligence, and personality: A critical review of the scattered literature. *Genetic and Social General Psychology Monographs* 7:355–429.

Best, M. 1999. How culture can guide evolution: An inquiry into gene/meme enhancement and opposition. *Adaptive Behavior* 132:289–293.

Craft, A. 2005. *Creativity in schools: Tensions and dilemmas*. London: Routledge.

Cropley, D.; Cropley, A.; Kaufman, J.; and Runco, M. 2010. *The dark side of creativity*. Cambridge UK: Cambridge University Press.

Donald, M. 1991. *Origins of the modern mind*. Cambridge MA: Cambridge University Press.

Eysenck, H. 1993. Creativity and personality: Suggestions for a theory. *Psychological Inquiry* 4:147–178.

Flaherty, A. 2005. Frontotemporal and dopaminergic control of idea generation and creative drive. *Journal of Computational Neurology* 493:147–153.

Florida, R. 2002. *The rise of the creative class*. London: Basic Books.

Gabora, L., and Aerts, D. 2009. A model of the emergence and evolution of integrated worldviews. *Journal of Mathematical Psychology* 53:434–451.

Gabora, L., and DiPaola, S. 2012. How did humans become so creative? In *Proceedings of the International Conference on Computational Creativity*, 203–210. Dublin, Ireland: Association for the Advancement of Artificial Intelligence.

Gabora, L., and Firouzi, H. 2012. Society functions best with an intermediate level of creativity. In *Proceedings of the 34th Annual Meeting of the Cognitive Science Society*, 1578–1583. Houston TX: Cognitive Science Society.

Gabora, L., and Kitto, K. 2013. Concept combination and the origins of complex cognition. In Swan, E., ed., *Origins of mind: Biosemiotics Series, Vol. 8*. Berlin: Springer. 361–382.

Gabora, L., and Saberi, M. 2011. How did human creativity arise? an agent-based model of the origin of cumulative open-ended cultural evolutiony. In *Proceedings of the ACM Conference on Cognition and Creativity*, 299–306. New York: Association for Computing Machinery.

Gabora, L.; Chia, W.; and Firouzi, H. 2013. A computational model of two cognitive transitions underlying cultural evolution. In *Proceedings of the 35th Annual Meeting of the Cognitive Science Society*, 2344–2349. Houston TX: Cognitive Science Society.

Gabora, L. 1995. Meme and variations: A computational model of cultural evolution. In Nadel, L., and Stein, D., eds., *1993 Lectures in Complex Systems*. Reading MA: Addison-Wesley.

Gabora, L. 2008. Evoc: A computational model of cultural evolution. In *Proceedings of the 30th Annual Meeting of the Cognitive Science Society*, 1466–1471. New York: Sheridan Publishing.

Gino, F., and Ariely, D. 2012. The dark side of creativity: Original thinkers can be more dishonest. *Journal of Personality and Social Psychology* 102:445–459.

Goodwin, D. 1998. *Alcohol and the Writer*. New York: Penguin.

Guilford, J. 1950. Creativity. *American Psychologisty* 5:444–454.

Higgs, P. 1992. The mimetic transition: a simulation study of the evolution of learning by imitation. *Proceedings of the Royal Society B - Biological Sciences* 267:1355–1361.

Hinton, G., and Nowlan, S. 1992. How learning can guide evolution. *Complex Systems* 267:495–502.

Holland, J. 1975. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

Hutchins, E., and Hazelhurst, B. 1991. Learning in the cultural process. In Langton, C.; Taylor, J.; Farmer, D.; and Rasmussen, S., eds., *Artificial life II*. Redwood City: Addison-Wesley.

Jamison, K. 1993. *Touched by fire: Manic-depressive illness and the artistic temperament*. New York: Free Press.

Kaufman, J. 2003. The cost of the muse: Poets die young. *Death Studies* 27:813–822.

Leijnen, S., and Gabora, L. 2009. How creative should creators be to optimize the evolution of ideas? a computational model. *Electronic Proceedings of Theoretical Computer Science* 9:108–119.

Ludwig, A. 1995. *The Price of Greatness*. New York: Guilford Press.

Maslow, A. 1959. Creativity in self-actualizing people. In Harper, and Brothers., eds., *Creativity and its cultivation*. New York: McGraw-Hill.

May, R. 1975. *The courage to create*. New York: Bantam.

Moran, S. 2011. The roles of creativity in society. In Kaufman, J., and Sternberg, R., eds., *Cambridge handbook of creativity*. Cambridge UK: Cambridge University Press.

Qian, M.; Plucker, J.; and Shen, J. 2010. A model of chinese adolescents creative personality. *Creativity Research Journal* 22:62–67.

Rogers, C. 1959. Toward a theory of creativity. In Anderson, H., ed., *Creativity and its cultivation*. New York: Harper & Row.

Rothenberg, A. 1990. Creativity, mental health, and alcoholism. *Creativity Research Journal* 3:179–201.

Scott, C. 1999. Teachers biases toward creative children. *Creativity Research Journal* 12:321–337.

Sternberg, R., and Lubart, T. 1995. *Defying the crowd: Cultivating creativity in a culture of conformity*. New York: Free Press.

Sulloway, F. 1996. *Born to rebel*. New York: Pantheon.

Tacher, E., and Readdick, C. 2006. The relation between aggression and creativity among second graders. *Creativity Research Journal* 18:261267.

Tomasello, M.; Kruger, A.; and Ratner, H. 1993. Cultural learning. *Behavioral and Brain Sciences* 16:495–552.

Torrance, E. 1963. *Guiding creative talent*. Englewood Cliffs, NJ: Prentice-Hall.

Treffinger, D.; Young, G.; Selby, E.; and Shepardson, C. 2002. *Assessing creativity: A guide for educators (RM02170)*. Storrs CT: University of Connecticut Press and The National Research Center on the Gifted and Talented.

Troyer, L., and Youngreen, R. 2009. Conflict and creativity in groups. *Journal of Social Issues* 65:409–413.

Wolfradt, U., and Pretz, J. 2001. Individual differences in creativity: Personality, story writing, and hobbies. *European Journal of Personality* 15:297–310.

A Four Strategy Model of Creative Parameter Space Interaction

Robert Tubb and Simon Dixon
Centre for Digital Music
Queen Mary University of London
London, E1 4NS, UK
r.h.tubb@qmul.ac.uk
simon.dixon@eecs.qmul.ac.uk

Abstract

This paper proposes a new theoretical model for the design of creativity-enhancing interfaces. The combination of user and content creation software is looked at as a creative system, and we tackle the question of how best to design the interface to utilise the abilities of both the computer and the brain. This model has been developed in the context of music technology, but may apply to any situation in which a large number of feature parameters must be adjusted to achieve a creative result. The model of creativity inspiring this approach is Wiggins' Creative Systems Framework. Two further theories from cognitive psychology motivate the model: the notion of creativity being composed of divergent and convergent thought processes, and the "dual process" theory of implicit vs. explicit thought. These two axes are combined to describe four different solution space traversal strategies. The majority of computer interfaces provide separate parameters, altered sequentially. This theory predicts that these one-to-one mappings encourage a particular navigation strategy ("Explicit-Convergent") and as such may inhibit certain aspects of creativity.

Introduction

Although enhancing creativity is often the implied goal, researchers in music technology seem wary of attacking the question of what manner of tools may augment the creativity of the musician. This is perhaps understandable: being one of the most mysterious products of our immensely complex brains, creativity is a great challenge to research. Individuals can vary enormously in how they go about being creative, and results from cognitive neuroscience are still rather contradictory (Dietrich and Kanso 2010). Therefore theoretical guidelines are scarce, and measuring success is difficult. This paper attempts to tie in some findings of cognitive psychology, computational creativity and digital musical instrument (DMI) research, to propose a simple four strategy model of creative interaction. A model that may explain many of the subjective experiences of computer musicians, and assist the design of creativity enhancing interfaces.

Creative Cognition

Guilford (1967) characterised the creative process as a combination of "convergent" and "divergent" thinking. Divergent production is the generation of many provisional candidate solutions to a problem, whereas convergence is the narrowing of the options to find the most appropriate solution. Most modern theories have similar processes present in some form, sometimes referred to by different names such as "Generative" and "Evaluative". Campbell (1960) and Simonton (1999) have considered creativity as a Darwinian process, and propose a process of idea variation and selection.

Another interesting process model of creativity is the incubation-illumination model (Wallas 1926). Illumination is more or less synonymous with "insight". Insight problems are a tool that psychologists have used to study this phenomenon. These are puzzles that no amount of step by step reasoning can solve. They often involve setting up some functional fixedness (commonly known as a "mental block"). Insight occurs when the problem is suddenly seen from a different angle. One claim is that conceptual combination processes can yield insight, but are beneath the level of consciousness. The "special process" model holds that these problems require completely different brain processes from logical or verbal problems (Schooler, Ohlsson, and Brooks 1993).

Wiggins' Creative Systems Framework (CSF) (Wiggins 2006) is a more formal descendent of Boden's theories of artificial creativity (Boden 1992). It describes creativity in terms of the exploration of conceptual space. It consists of the universe of all possible concepts $\mathscr{U}$, an existing conceptual space (for example domain knowledge) $\mathscr{C}$, rules (constraints) that define this conceptual space $\mathscr{R}$, a set of techniques to traverse the space $\mathscr{T}$, and an evaluation method $\mathscr{E}$: a way to assign value to a location c that yields a "fitness function". Exploratory creativity is said to proceed as follows: if traversal takes us outside the space of existing concepts this results in an "aberration". If the aberration proves valuable according to $\mathscr{E}$, then the new point is included in the domain, and the conceptual space is extended. Wiggins claims that transformational creativity (a fundamental shift in the rules of the domain) can be viewed as no different from exploratory creativity but on a meta-level. This is to say that a transformation of conceptual space can be achieved by exploring the conceptual space of conceptual spaces. Later we attempt to adapt this model to apply to a parameter space, to propose what creativity might mean in the (very reduced) case of adjusting continuous controls of a sound synthesis engine.

System 1 / Implicit	System 2 / Implicit
associative	rule-based
holistic	analytic
automatic	controlled
relatively undemanding	demanding
fast acquisition by biology + experience	slow acquisition by cultural and formal tuition
evolved first	evolved recently
short term reactions	long term planning
parallel	serial
large associative memory	limited working memory

Table 1: Contrasts between implicit and explicit thinking (Stanovich and West 2000).

Dual Process Models of Cognition

The formal definition of intuition states that it is the ability to acquire knowledge without the use of reason. This is a rather negative definition, and inspires the question: what mechanisms are present in the brain *apart* from reason? A more positive approach to nailing down intuitiveness is to make use of the "dual process theory" of reasoning (Evans 2003; Kahneman 2011). The dual process hypothesis is that two systems of different capabilities are present in the brain. The first (System 1) is fast, parallel and associative, but can suffer from inflexibility and bias. The second (System 2) is more rational and analytical, but is slower, requires intentional effort, and has limited working memory. In this paper we shall use the more illustrative terms "Implicit" and "Explicit" to refer to System 1 and 2 respectively. Table 1 lists descriptions of the two systems, taken from Stanovich and West (2000). This portrayal is often used by social psychologists to explain why many decisions that humans take (under, for example, time constraints) seem to be irrational (De Martino et al. 2006). The theory, however, is also relevant to a great deal of other human behaviour, including problem solving, human-computer interaction, and surely creativity. It should be noted that both these systems are extremely broad high-level categorisations. Implicit processing, for instance, encompasses a whole host of perceptual, motor, linguistic and emotional systems. For this reason Stanovich (2009) proposes that implicit system should be called TASS (The Autonomous Set of Subsystems), and also suggests the explicit system breaks down into two subsystems: the "reflective" and the "algorithmic".

How might the two processes relate to creativity? Holistic thinking has historically been associated with the right brain, and also with creativity. However, whilst left/right asymmetries can be dramatic (McGilchrist 2009), creativity is unlikely to be an exclusively right-brain phenomenon (Dietrich 2007). One might also conflate divergent thinking with the fast-unconscious system, and convergent thinking with the slow-conscious. However, tacit thinking is mostly quick-access default behaviour, and can be stubbornly inflexible, exactly the *opposite* of novel idea generation.

It is also clear that explicit thinking can create wildly divergent ideas. That is, by asking new questions, intentionally avoiding the obvious by imposing constraints, or redesigning the creative process itself, a point in the solution space may be reached that is very distant from existing concepts (Joyce 2009). This nonetheless relies on a conscious, symbolic, and often systematic approach. Therefore a particularly important aspect of the explicit system's abilities is reflection, or meta-cognition: the ability to inspect one's own thoughts (Buchanan 2001). In Pearce and Wiggins' cognitive model of the composition process, at least three out of the five processes relate to reflective abilities (Pearce and Wiggins 2002). So associating artistic creativity with intuitive thinking misses this fact that transformations can result from using analytical symbolic thought to intentionally change the rules, strategies and even value systems of the creative domain. Next we shall investigate the ramifications of both fast and slow systems being able to conduct both divergent and convergent strategies, and try to define them in terms of solution space traversal mechanisms. This model then prompts consideration of how the interface may help or hinder creative work.

Creative Interaction with Synthesis Parameters

The CSF terminology becomes useful for asking what creativity might mean when navigating a finite, continuous parameter space, such as that provided by a music synthesiser. Whilst the complete CSF is not yet rigorously applied, the main components map well onto the various elements of the human-computer system. As the musician is interacting with the parameter space, and is constrained by it, it is ostensibly a space of viable compositions $\mathscr{C}_{param}$, and the interface provides $\mathscr{T}_{param}$: the mechanisms to navigate the space. Obviously there are cultural and emotional associations that sounds may possess that are not represented in this very reduced domain. Parameters such as pitch, filter cut-off frequency, and amplitude envelopes only represent the lowest levels in the hierarchical conceptual space of music. Nevertheless, for this work we assume that the the higher level concepts mainly influence $\mathscr{E}$. By assuming that the evaluation of the fitness of a given point in parameter space is carried out by the user, difficult questions such as the cultural associations of particular sounds can be side stepped. The interface designer can assume some complex fitness function is being optimised, without needing to know its exact form (though interesting work has been done both tracking users paths through solution space and obtaining value ratings (Jennings, Simonton, and Palmer 2011)). However this does not mean that the navigation of solution space is entirely carried out within the brain. The constraints and "affordances" (Norman 1999) of the tools, notations and abstractions used for composition have a significant effect on the finished product (Mooney 2011; Magnusson 2010). For example, the following situations may arise:

1. The composer will sometimes have a idea in mind, and will therefore need to optimise parameters such that the idea is realised.
2. The composer will, at other times, not have anything specific in mind, and is looking to engage in an exploratory

process that may produce inspiration.

These two scenarios map very well to notions of convergent and divergent thinking. In the first case the creative act has already occurred in the brain of the composer, and all that is necessary is an interface that enables the user to adjust parameters such that the data converges to the idea. Such would be the case in live performance of a score: the piece exists, but should be realised accurately, and according to the performers expressive intent. This is of course a great design challenge. But the second scenario is just as important: the composer embarks on an interactive journey, and unpredictability is a key ingredient. Accidents and surprise are often seen as key components of the creative process (Kronengold 2005; Fiebrink et al. 2010). Therefore would appear that some of the divergent thought can be outsourced to the technology. These technological flukes are analogous to the aberrations in the CSF. Thus the design of the instrument affects creativity, not just in the surface sense that different instruments have varying timbres, but in a deep sense of the interface frames and guides the process, similar to the way language guides thought, or that unconscious priming may change behaviour. A previous experiment has shown that divergent and convergent stages can be best served by different types of interfaces (Tubb and Dixon 2014).

Divergent and convergent modes seem also to have a different relationship to $\mathscr{E}$. Many musicians and sound designers intentionally put themselves into states of mind where they temporarily suspend criticism[1]. This implies that it is useful to disengage evaluation, in order that local minima in that fitness function may be escaped.

The mapping of physical controllers to sound synthesis parameters has been an active research topic for at least twenty years (Winkler 1995; Wanderley and Depalle 2004). Mapping has a significant effect not only on what sounds are easy or difficult to create, but also the subjective experience of the user.

The principal distinctions between types of mappings are as follows (Hunt, Wanderley, and Kirk 2000).

- One-to-many: one control dimension is mapped to many synthesis parameters.
- Many-to-one: many control parameters affect one synth parameter.
- Many-to-many: a combination of the above.

Research has shown (Hunt and Kirk 2000) that complex many-to-many mappings appear to be more effective for expressive performance, and may lead to greater performance improvements with practice. This seems to imply that if a mapping is multi-dimensional, and confounds the users' attempts to analyse and manipulate the dimensions separately, then implicit learning cognitive systems are employed. Dimensions that are amenable to being bound together perceptually are termed "integral" (Jacob et al. 1994). For example

[1]It is unlikely that musicians turn off *all* judgement. It could be that they switch to assessment using fast "gut feeling" assessments (Implicit), rather than more demanding evaluations using analytical, art-theoretical evaluations or a theory of other minds (Explicit).

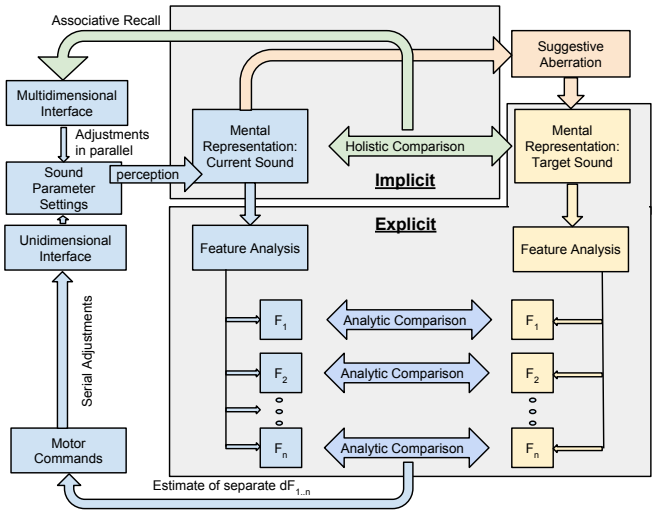


Figure 1: A cognitive model of altering synthesis parameters to match a desired goal. With the use of complex multidimensional controllers (the upper action-perception loop), implicit processes are hypothesised to compute mappings from multidimensional feature sets to motor movements beneath conscious awareness.

colour space is formed of 3 integral dimensions, however colour and position are mutually "separable". Timbre space is large;y integral, therefore one may question the approach of providing dimensions separately. Practice of a complex controller is less like carrying out a series of commands, and more like learning to ride a bike. Eventually this leads to increased processing bandwidth in the action-perception loop. Hunt also suggests that implicit learning frees up explicit resources to work on other things.

A tentative cognitive model of how the implicit and explicit systems navigate parameters is shown in figure 1. This applies to the case when the composer has a specific target in mind, although there is always the possibility that a chance discovery will produce an aberration and an alternative target may be suggested. On the left is the technology, the sound parameters, synthesis engine. Two interfaces are shown, the lower one a unidimensional (slider or WIMP interface) interface and a multi-dimensional (physical controller with complex mapping). If the multi-dimensional interface is well learned, then automatic, holistic processing can process in parallel a large number of features that must otherwise be sequentially adjusted, whilst the goal and its features are held in working memory. The drawbacks of the fast action-perception cycle are firstly, that to become accurate it requires large amounts of practice, and secondly, that it will be poor at adapting to unencountered target sounds or interface mappings. It is worth noting that these drawbacks only apply in the convergent case. For divergence, even an unlearned multidimensional interface may be beneficial (Tubb and Dixon 2014).

A Four-Strategy Model of Creative Interaction

This theory details how a simple two stage model of creativity (divergence vs. convergence) and dual process theory (implicit vs. explicit) can be combined to inform the design of creative composition interfaces. It is worth setting out the exact scope of this model. It is not intended to be a model of separate systems within the brain. It is not intended to have any predictive power outside the domain of interaction with a parameter space, though it may prove useful in other areas, and we speculatively propose how these four strategies may interact to produce insight. Furthermore, important cultural, personality and emotional considerations have been ignored. It only addresses what Boden (Boden 1992) terms P-creativity, rather than the H-creativity found in culturally significant achievements. Specifically, it is intended to be a categorisation of parameter search strategies, a summary of how those strategies work together (or not) to create novelty and value, and how parameters should be mapped to gestures to assist each of these processes. This design methodology should prevent the designer forcing the user into the wrong creative problem solving strategy at the wrong time.

Divergent and Convergent Solution-Space Traversal

First of all we attempt to define divergent and convergent processes with reference to the CSF (Wiggins 2006).

Convergent processes are traversal mechanisms that improve the fitness of solutions. These could be a series of discrete options, for example selecting the best sound from a number of candidates, or they could be a continuum, for example finding the "best" setting for a synthesis parameter is a convergent process. Convergence requires both a fitness evaluation $\mathscr{E}$, and some prediction of what change will increase value, which yields a parameter traversal strategy $\mathscr{T}$. $\mathscr{E}$ is therefore actively employed in guiding $\mathscr{T}$. This is analogous to a gradient descent algorithm (these algorithms are said to "converge" on a solution). So whilst some models of creativity postulate generative and evaluative stages, where convergence is just evaluation and selection, in our model convergence can still change the solution (i.e. incremental improvement rather than just evaluation or selection c.f. the "honing" theory of creativity (Gabora 2005)). A second method of convergence is more analytical: where $\mathscr{E}$ can be broken down into smaller individual success criteria, each of which requires a non-creative solution.

Divergent processes are different in that they set aside questions of improving any fitness value, and generate candidate solutions distant from the current ones, e.g. creating lots of more or less randomly scattered points. $\mathscr{E}$ may still operate in the background in order to spot promising new ideas, but is disengaged from directly determining $\mathscr{T}$, in order to prevent it revisiting unoriginal ideas. An alternative divergent approach can be carried out on the meta-level: deliberately transforming the fitness function or the constraints.

Convergence by itself will rarely produce novelty, as multiple runs will settle in the same local minimum. Divergence by itself will produce useless noise. It is the careful blending of these processes that yields progress. Examples abound from machine learning that combine both divergent and convergent behaviours, such as random forests, genetic algorithms and particle swarm optimisation. Balancing the two tendencies is also known as the exploration-exploitation trade-off (Barto 1998). Often such algorithms progressively reduce the diversity component as the search progresses.

So by defining divergence and convergence in this way, we see that by strategically connecting and disconnecting judgements of fitness from the parameter navigation strategy, the musician can produce both novelty and value.

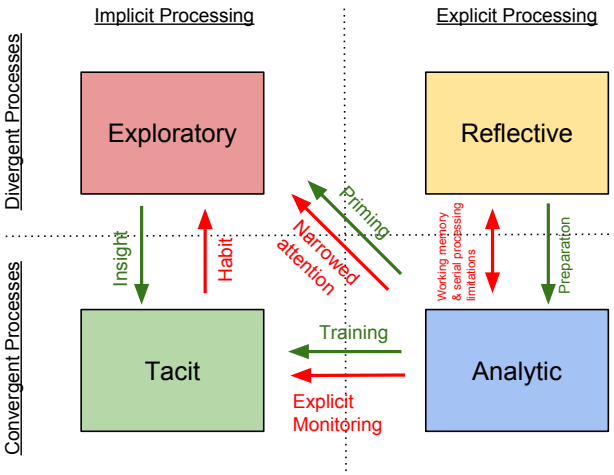


Figure 2: The four quadrants of implicit vs. explicit thinking (left/right) and divergent and convergent thinking (top/bottom). Examples of useful information transfer are shown in green. Examples of detrimental interference effects shown in red.

The Four Quadrant Model

The central hypothesis in this section is that both fast and slow brain systems may conduct convergent or divergent searches. This results in four distinct parameter space traversal strategies.

Figure 2 shows the four categories: divergent-implicit (exploratory), divergent-explicit (reflective), convergent-implicit (tacit) and convergent-explicit (analytic). These may be strategies carried out within the brain (conceptual space traversal), or actual manipulations of the controls of an instrument (parameter space traversal). Below, each quadrant is described in more detail, both in terms of cognitive processes and interfaces that may augment them.

Exploratory (implicit-divergent) refers to stochastic, associative, combinatorial or transformational processes that can quickly generate a large number of points across a solution space. Examples may be the unconscious process of conceptual recombination, techniques such as brainstorming, or simple playfulness. Computers effectively generate random, transformed and recombined data, therefore exploration is easily augmented.

Tacit (implicit-convergent) is intended to refer to those instinctive or learned techniques that quickly produce a valuable, but probably unoriginal local solution to a problem. These could be instinctive, or learned well enough to become automatic. The appropriate interface is a well learned complex, multi-dimensional, space-multiplexed interface such as a traditional musical instrument, but could also be interaction metaphor such as a physical model that makes use of instinctive understanding of the physical world.

Analytic (explicit-convergent) processes break a problem down into separate components, and solve them in a sequential way. In the solution space it would proceed in a city-block fashion, one dimension at a time. An analytic interface is one such as a DAW[2] that provides individual parameters as knobs and sliders, and sequential, time-multiplexed input devices such as the mouse. These tend to rely on the perceptual aspects of the parameters themselves being fairly independent and separable. The great advantage of this mode is that complex problems can be broken down into simpler parts. With well defined goals and predictably behaved parameters, accurate location of desired solutions can be achieved in linear time, despite the exponential increase in the size of the space.

Reflective (explicit-divergent) refers to meta-cognitive analytical methods that can take existing conceptual spaces and infer new ones: proposing entirely new problem spaces by asking questions or generating hypotheses. One mechanism is that the analytic system *transforms* the solution space, the constraints and/or the fitness function, deliberately forcing converged points out of their local solution finding complacency[3]. Other reflective strategies may be use of metaphor and analogy. For truly transformational creativity this meta-exploration ability is essential. A reflective musical interface might be one that offers the ability to create new musical abstractions, for example a musical programming language (Blackwell and Collins 2005; Bresson, Agon, and Assayag 2011).

The final component to add to this model regards the evaluation process. Judgement too can be divided into implicit and explicit manifestations. Implicit judgement is fast and affective ("I like this" or "I don't like that"). Explicit judgement is more demanding but it is more of a sighted process i.e. also providing the value function gradient ("I like this because..." or "I don't like that, it needs the following...").

All four quadrants play a part in creativity. Take the incubation-illumination model as a, highly speculative, illustration, purely in the cognitive domain. Preparation is the process of asking a new question, or finding a new problem (reflective), and attempting to solve it, consciously via the (methodical) solutions of the past. Applying methods based on past rules and concepts leads to repeated failure, but this process is both activating concepts in the subconscious for recombination (a process known as priming), and tacitly learning how to quickly select a solution (constructing a neural fitness landscape that will function as an unconscious solution recogniser). At some point one of the many divergent (exploratory) subconscious combinations will be implicitly recognised, and then "miraculously" provided to the conscious mind[4]. In this way implicit parallelism can be set to work exploring large regions of a complex solution space.

Insight may be an example of when these strategies gel, however there may also be inhibition effects (some are shown as red arrows in figure 2) when they work against one other. Probably the single most important inhibition effect is that explicit processing is serial, with limited working memory. Therefore if it is fully engaged with analytic processing, e.g. dealing with many separate musical parameters, there will be less resources available for meta-cognition and high level reasoning. Tasks such as critical listening have been shown to suffer under interface-induced higher cognitive load (Mycroft, Reiss, and Stockman 2013). Other inter-quadrant interference effects include "explicit monitoring", also known as "analysis paralysis": a phenomenon where if an attempt is made to consciously control an automatic action, performance suffers (Masters 1992; Wan and Huon 2005). Habit naturally inhibits exploration: an automatic action will tend to be repetitive and inflexible (Barrett 1998).

One final inhibition effect is that analytic thought involves narrowed attention: users may be less open to peripheral cues and remote associations emerging from exploratory processes (Ansburg and Hill 2003). This prediction seems to align with many users' reports of using computers to make music: the fact they can get hung up on details, lose perspective and miss the big picture of what they are attempting to express. Evaluation of one's own work requires taking a step back to get a "perspective" of structure at longer time scales (Nash and Blackwell 2012). Lack of perspective can be a problem when manipulating complex interfaces:

> Participants voiced strong feelings that computer-music systems encouraged endless experimentation and fine-tuning of the minutiae of sound design, in conflict with pushing forward and working on higher-level compositional decisions and creating finished works. (Duignan, Noble, and Biddle 2010)

Unfortunately the reflective attention monitoring system may itself be inhibited, therefore preventing the realisation that perspective has been lost. So, in summary, there seems to be a high risk that explicit-convergent interfaces may inhibit high level transformational creativity.

[2]The Digital Audio Workstation. Effectively a software reconstruction of an entire recording studio.

[3]A useful analogy would be tipping the surface of a "tilt maze" in order to extract a ball from a hole, and help its progress to the final goal.

[4]Wiggins proposes that the criterion for admission into consciousness is not only the certainty of the idea as a good solution, but also an information theoretic measure of surprise: implying that novelty generation is practically hard-wired into the threshold between implicit and explicit thought (Wiggins 2012).

Discussion

The principal application of the above framework is to generate a number of guidelines by which to design and evaluate creative interfaces. Some of these will already correspond with those put forward within the HCI and DMI literature, some may be novel. However, we propose one underlying principle: just as the dimensional structure of the interface (how the parameters are presented and mapped) must match the perceptual nature of the task (Jacob et al. 1994), so also the structure of the interface must be able to match the current creative strategy of the artist. The computer interface should follow the human thought process as closely as possible, not only in terms of the steps required to *render* a final product, but also in terms of the different geometries of the search strategies employed to *discover* that final product. Therefore the interface must support exploratory, reflective, tacit and analytic modes.

We propose that the incubation-illumination cycle outlined in the previous section is already somewhat mirrored in creative technological interaction. However, to date this has not been specifically designed for, so there is surely room for improvement. Technologies exist that augment each individual quadrant, but principally lacking are easy *transitions between* strategies. For example switching between instrumental play to computer based editing to designing one's own musical abstractions is currently quite demanding, and generally stalls any creative flow. How could all four modes be provided without merely increasing the cognitive load? How, specifically, are these twelve possible[5] transitions to be carried out? This is our topic of further research.

Almost all user interfaces for creative software provide parameters such that features are edited in a separate, serial fashion. These interfaces are used to create music, animation, industrial design, architecture and computer games. They find their way into almost every aspect of 21st century digital culture. If this interaction paradigm really does change the way that people are creative, this seemingly innocent and logical arrangement may already have had significant consequences for the quality of artistic innovations. Will new multidimensional interaction devices encourage a different approach?

Currently, this is just a speculative model, albeit informed by and retrodicting other research and experiences in the electronic music community. Further work will attempt to find evidence for the efficacy of this approach via experiments, interaction data analysis and interviews regarding the artists own strategies of using computers to be creative.

[5]Enumeration of all of these is beyond the scope of this paper. However one illustrative example would be to start by improvising with a complex tacit interface, but then abstract major themes (perhaps automatically) from that improvisation. These themes would be then gathered in a reduced space, to be explored, recombined and performed using the *same* multi-dimensional interface. Themes in the explorations in this new space could again be extracted, producing a recurrent exploratory/reflective process that also leverages tacit skill.

References

Ansburg, P. I., and Hill, K. 2003. Creative and analytic thinkers differ in their use of attentional resources. *Personality and Individual Differences* 34(7):1141 – 1152.

Barrett, F. J. 1998. Coda - creativity and improvisation in jazz and organizations: Implications for organizational learning. *Organization Science* 9(5):605–622.

Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT press. 25–49.

Blackwell, A., and Collins, N. 2005. The programming language as a musical instrument. *Proceedings of PPIG05 (Psychology of Programming Interest Group)*.

Boden, M. A. 1992. *The creative mind: Myths and mechanisms*. Abacus.

Bresson, J.; Agon, C.; and Assayag, G. 2011. Openmusic: Visual programming environment for music composition, analysis and research. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, 743–746. New York, NY, USA: ACM.

Buchanan, B. G. 2001. Creativity at the metalevel: Aaai-2000 presidential address. *AI magazine* 22(3):13–28.

Campbell, D. T. 1960. Blind variation and selective retentions in creative thought as in other knowledge processes. *Psychological review* 67(6):380.

De Martino, B.; Kumaran, D.; Seymour, B.; and Dolan, R. J. 2006. Frames, biases, and rational decision-making in the human brain. *Science* 313(5787):684–687.

Dietrich, A., and Kanso, R. 2010. A review of EEG, ERP, and neuroimaging studies of creativity and insight. *Psychological bulletin* 136(5):822.

Dietrich, A. 2007. Whos afraid of a cognitive neuroscience of creativity? *Methods* 42(1):22–27.

Duignan, M.; Noble, J.; and Biddle, R. 2010. Abstraction and activity in computer-mediated music production. *Computer Music Journal* 34(4):22–33.

Evans, J. S. B., and Frankish, K. E., eds. 2009. *In two minds: Dual processes and beyond.* Oxford University Press.

Evans, J. S. B. 2003. In two minds: dual-process accounts of reasoning. *Trends in cognitive sciences* 7(10):454–459.

Fiebrink, R.; Trueman, D.; Britt, C.; Nagai, M.; Kaczmarek, K.; Early, M.; Daniel, M.; Hege, A.; and Cook, P. 2010. Toward understanding human-computer interaction in composing the instrument. In *Proc. of the International Computer Music Conference*.

Gabora, L. 2005. Creative thought as a non darwinian evolutionary process. *The Journal of Creative Behavior* 39(4):262–283.

Guilford, J. P. 1967. *The nature of human intelligence.* McGraw-Hill (New York).

Hunt, A., and Kirk, R. 2000. Mapping strategies for musical performance. *Trends in Gestural Control of Music* 21.

Hunt, A.; Wanderley, M.; and Kirk, R. 2000. Towards a model for instrumental mapping in expert musical interaction. In *Proceedings of the 2000 International Computer Music Conference*, 209–212.

Jacob, R. J.; Sibert, L. E.; McFarlane, D. C.; and Mullen Jr, M. P. 1994. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction (TOCHI)* 1(1):3–26.

Jennings, K. E.; Simonton, D. K.; and Palmer, S. E. 2011. Understanding exploratory creativity in a visual domain. In *Proceedings of the 8th ACM conference on Creativity and cognition*, 223–232. ACM.

Joyce, C. 2009. *The blank page: effects of constraint on creativity*. Ph.D. Dissertation, Haas School of Business, University of California, Berkeley.

Kahneman, D. 2011. *Thinking, fast and slow*. Farrar, Straus and Giroux.

Kronengold, C. 2005. Accidents, hooks and theory. *Popular Music* 24(3):381.

Magnusson, T. 2010. Designing constraints: Composing and performing with digital musical systems. *Computer Music Journal* 34(4):62–73.

Masters, R. S. 1992. Knowledge, knerves and know-how: The role of explicit versus implicit knowledge in the breakdown of a complex motor skill under pressure. *British journal of psychology* 83(3):343–358.

McGilchrist, I. 2009. *The master and his emissary: The divided brain and the making of the western world*. Yale University Press.

Mooney, J. 2011. Frameworks and affordances: Understanding the tools of music-making. *Journal of Music, Technology and Education* 3(2-3):2–3.

Mycroft, J.; Reiss, J. D.; and Stockman, T. 2013. The influence of graphical user interface design on critical listening skills. In *Sound and Music Computing (SMC), Stockholm*.

Nash, C., and Blackwell, A. 2012. Liveness and flow in notation use. In Essl, G.; Gillespie, B.; Gurevich, M.; and O'Modhrain, S., eds., *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*. Ann Arbor, Michigan: University of Michigan.

Norman, D. A. 1999. Affordance, conventions, and design. *interactions* 6(3):38–43.

Pearce, M., and Wiggins, G. A. 2002. Aspects of a cognitive theory of creativity in musical composition. In *Proceedings of the ECAI02 Workshop on Creative Systems*.

Schooler, J. W.; Ohlsson, S.; and Brooks, K. 1993. Thoughts beyond words: When language overshadows insight. *Journal of experimental psychology: General* 122(2):166.

Simonton, D. K. 1999. *Origins of genius: Darwinian perspectives on creativity.* Oxford University Press.

Stanovich, K. E., and West, R. F. 2000. Individual differences in reasoning: Implications for the rationality debate? *Behavioral and brain sciences* 23(5):645–665.

Stanovich, K. E. 2009. *Distinguishing the reflective, algorithmic, and autonomous minds: Is it time for a tri-process theory?*, In Evans and Frankish (2009). 55–88.

Tubb, R., and Dixon, S. 2014. Sonic zoom: A zoomable mapping of a musical parameter space using hilbert curves. *Computer music journal* 38(3):forthcoming.

Wallas, G. 1926. *Art of Thought*. Verlag.

Wan, C. Y., and Huon, G. F. 2005. Performance degradation under pressure in music: An examination of attentional processes. *Psychology of Music* 33(2):155–172.

Wanderley, M. M., and Depalle, P. 2004. Gestural control of sound synthesis. *Proceedings of the IEEE* 92(4):632–644.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Wiggins, G. A. 2012. The mind's chorus: Creativity before consciousness. *Cognitive Computation* 4(3):306–319.

Winkler, T. 1995. Making motion musical: Gesture mapping strategies for interactive computer music. In *Proceedings of the International Computer Music Conference*.

Autonomously Managing Competing Objectives to Improve the Creation and Curation of Artifacts

David Norton, Derrall Heath, Dan Ventura
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
dnorton@byu.edu, dheath@byu.edu, ventura@cs.byu.edu

Abstract

DARCI (Digital ARtist Communicating Intention) is a creative system that we are developing to explore the bounds of computational creativity within the domain of visual art. As with many creative systems, as we increase the autonomy of DARCI, the quality of the artifacts it creates and then curates decreases—a phenomenon Colton and Wiggins have termed the latent heat effect. We present two new metrics that DARCI uses to evolve and curate renderings of images that convey target adjectives without completely obfuscating the original image. We show how we balance the two metrics and then explore various ways of combining them to autonomously yield images that arguably succeed at this task.

Introduction

There has been a recent push in computational creativity towards fully autonomous systems that are perceived as creative in their own right. One of the most significant problems facing modern creative systems is the level of curation that is occurring in these systems. If a system is producing dozens, hundreds, or even thousands of artifacts from which a human is choosing a single valued artifact, then is the system truly fully autonomous? Colton has argued that for a system to be perceived as creative, it must demonstrate appreciation for its own work (Colton 2008). A strong implication of this is that the system must be able to do its own curation by autonomously selecting an artifact for human judgment.

DARCI (Digital ARtist Communicating Intention) is a creative system that we are developing to explore the bounds of computational creativity within the domain of visual art. DARCI is composed of several subsystems, each with its own creative potential, and each designed to perform an integral step of image creation from conception of an idea, to design, to various phases of implementation, to curation. The most complete subsystem, and the one that is the focus of this paper, is called the *image renderer*. The image renderer uses a genetic algorithm to discover a sequence of image filters that will render an image composition (produced by another subsystem) so that it will reflect a list of adjectives (selected from yet another subsystem). After evolving a population of candidate renderings, the image renderer must select an interesting candidate that reflects both the original image *and* the given adjectives—in other words, it must curate the finished artifacts.

Historically, DARCI has been successful at producing such images when curation is a joint effort between DARCI and a human (Norton, Heath, and Ventura 2011b; Heath, Norton, and Ventura 2013). In these cases, DARCI selects a number of artifacts, and a human chooses their favorite from that selection. When DARCI curates on its own, the results have been significantly less successful. This decrease in quality is to be expected and is a phenomenon Colton and Wiggins call the *latent heat effect*—"as the creative responsibility given to a system increases, the value of its output does not (*initially*) increase ..." (emphasis added) (Colton and Wiggins 2012). Since we know DARCI is capable of producing interesting images, we are interested in *increasing* the value of the artifacts the system produces when curating alone, thus decreasing the latent heat effect.

DARCI's image renderer uses a combination of two conflicting metrics as a fitness function to evaluate and assign fitness scores to candidate artifacts. The fitness score not only drives the evolution of artifacts using a genetic algorithm, it is also used to curate the population of candidate artifacts when evolution is complete. For this paper we have made improvements to the fitness function in order to improve the quality of artifacts DARCI produces.

Previously, the fitness function has been the combined average of an ad-hoc interest metric and an adjective matching metric. In this paper, we will abandon the interest metric in favor of a new similarity metric, and combine it with an improved adjective matching metric. While we take measures to ensure that both metrics output real values in a similar range, experience has shown that the two metrics are not measuring attributes of equal quality. This has led to the observation that if combining metrics with an average, the algorithm will give disproportionate weight to the metric that is easier to maximize. Thus, we will investigate different means of combining these two metrics in an attempt to more effectively balance the requirements put upon the image rendering subsystem and decrease the latent heat effect. We show the results of these new fitness functions in figures curated strictly by DARCI.

Image Rendering

The image rendering subsystem uses a series of image filters to render pre-existing images which we refer to as *source images*. The subsystem has access to Photoshop-like filters

with varying parameters. It uses a genetic algorithm to discover the configuration and parameter settings of these image filters so that candidate artifacts will reflect target adjectives without over or under-filtering the source image (Norton, Heath, and Ventura 2011b; 2013). A genetic algorithm is used because evolutionary approaches elegantly facilitate the creation of artifacts through both combination and exploration, two processes described by Boden for generating creative products (Boden 2004). Gero has also outlined how the processes underlying evolution are ideal for producing novel and unexpected solutions, a crucial part of creativity (Gero 1996). Finally, we have shown how evolutionary algorithms approximate some aspects of the creative process in human artists (Norton, Heath, and Ventura 2011a).

In this section we will describe in detail the two metrics used in this paper: adjective matching and similarity.

Adjective Matching

The adjective matching metric is the output of a learning subsystem of DARCI called the Visuo-Linguistic Associator (VLA). The VLA is a collection of artificial neural networks (ANN) that learns to associate image features with adjectives through backpropagation training. The original VLA has been described in detail previously (Norton, Heath, and Ventura 2010). Here we introduce an improved VLA.

While DARCI is designed to function as an online system, the original VLA required subsystem resets whenever it was time to introduce new training data, essentially learning in batch. Thus, in order for DARCI to adapt, human intervention was needed at regular intervals. The new VLA uses an approach closer to incremental learning to better facilitate the desired autonomous online functionality. Additionally, the new VLA uses a more accurate and complete approach to predicting additional training data. In this section we will describe the new VLA without any assumptions that the reader is familiar with the previous system.

Training Data Training data for DARCI is contained in a database. Each data point consists of an adjective (the label), the sentiment toward the adjective (positive or negative), the image features associated with the adjective (the image), and a time stamp. In our research, the term *adjective* always refers to a unique adjective synset as defined in WordNet (Fellbaum 1998). Hence, different senses of the same word will belong to different synsets, or adjectives.

Data points are added to the database as they are submitted by volunteers using a training website (Heath and Norton 2009). Whenever the training algorithm is invoked, new *relevant* data points are introduced to the learner one at a time in the submitted order. The learner consists of a series of binary ANNs, one for each *relevant* adjective. An adjective, and any corresponding data point, is considered *relevant* once there are at least ten *distinct* positive and ten *distinct* negative instances of the adjective in the database. Here, *distinct* means occurrences of the adjective with unique sets of image features (i.e. if an adjective is used to label the same image multiple times it only counts as one occurrence). At the moment the learner is invoked, a new neural network is created for any new adjectives that have become relevant.

Table 1: Image features used to train neural networks.

Color & Light:
1. Average red, green, and blue
2. Average hue, saturation, and intensity
3. Saturation and intensity contrast
4. Unique hue count (from 20 quantized hues)
5. Hue contrast
6. Dominant hue
7. Dominant hue image percent

Shape:
1. Geometric moment
2. Eccentricity
3. Invariant moment (5x vector)
4. Legendre moment
5. Zernike moment
6. Psuedo-Zernike moment
7. Edge direction histogram (30 bins)

Texture:
1. Co-occurrence matrix (x4)
   1. Maximum probability
   2. First order element difference moment
   3. First order inverse element difference moment
   4. Entropy
   5. Uniformity
2. Edge frequency (25x vector)
3. Primitive length
   1. Short primitive emphasis
   2. Long primitive emphasis
   3. Gray-level uniformity
   4. Primitive uniformity
   5. Primitive percentage

The reason we only create and train the learner on relevant data points is a matter of practicality. There are over 18000 adjective synsets in WordNet, and at the time of this writing more than 6000 adjective synsets in DARCI's database. However, most of the adjectives in DARCI's database are rare with only one or two positive data points. This is not enough data to successfully train any learner in a complex domain such as image annotation. Since performance speed is important for DARCI, accessing 6000 neural nets, most of which would be insufficiently trained, to annotate an image is impractical. As of this writing, DARCI has 237 relevant adjectives, a much more useful and manageable number. Taking synonyms into consideration, these relevant adjectives cover most standard adjectives.

The learner's neural networks are trained using standard back propagation with 102 image features as inputs. These image features are widely accepted global features for content based image retrieval, and most of them are available through the DISCOVIR (DIStributed COntent-based Visual Information Retrieval) system (King, Ng, and Sia 2004; Gevers and Smeulders 2000). A summary of the features we use can be found in Table 1. These features describe the color content, lighting, textures, and shape patterns found in images. Specific to the art domain, several researchers have shown that such features are useful in classifying images according to aesthetics (Datta et al. 2006), painting genre (Zujovic, Gandy, and Friedman 2007), and emotional semantics (Wang, Yu, and Jiang 2006). As many of these researchers have found color to be particularly useful in classifying images, we added four color-based features inspired by Li's own colorfulness features (Li and Chen 2009) to those contained in DISCOVIR. In Table 1 these colorfulness features are "Color & Light" numbers 4-7.

When training neural networks in batch, back propagation requires many epochs of training to converge. During each epoch, all of the training data is presented to the neural network in a random order. To imitate this with incremental learning, each new data point is introduced to the appropriate neural network along with a selection of previous data points. Along with this *recycled* data, additional data points

are *predicted* from the co-occurrences of adjectives with images. By including predicted data we are able to augment the limited data we do have. Similar, but less complete, approaches to augmenting training data have been successful in the past (Norton, Heath, and Ventura 2010).

Recycling Data For each new data point presented to a neural network for a given adjective, a, n positive data points from the set of all previous positive data points for the given adjective, D_{a+}, and n negative data points from the set of all previous negative data points for the given adjective, D_{a-}, are selected. The data points are selected with replacement according to the probability $P(rank(d))$ where $d \in D_{as}$, s is the sentiment of the set ($-$ or $+$), and $rank(d)$ is the temporal ordering of element d in D_{as}. The most recent element has a rank of $|D_{as}|$ and the oldest element has a rank of 1. The equation for $P(rank(d))$ is as follows:

$$P(rank(d)) = \frac{rank(d)}{\sum_{i=0}^{|D_{as}|} i} \quad (1)$$

The value for the number of previous data points chosen, n, is defined by $n = min(r, |D_{a+}|, |D_{a-}|)$ where r is a parameter setting the maximum number of data points to recycle each time a new data point is introduced. For the experiments in this paper, this value is set to 100.

Informally, every time a new data point is presented to a neural network, an equal number of positive and negative data points are selected from the previous data points for that neural network. These are selected randomly but with a higher probability given to more recent data.

Predicting Data To augment the training data we collect from DARCI's website, we analyze the co-occurrence of relevant adjectives to predict additional data points. Here we say that two adjectives co-occur whenever the same image is labeled with both adjectives at least once—these labels can be negative or positive. As each new data point is introduced to the learner, co-occurrence counts (distinct images) are updated for all pairings of relevant adjectives across all four combinations of sentiment. For example, as of this paper, 'scary' has 26 co-occurrences with 'disturbing' (or 'scary' co-occurs with 'disturbing' in 26 distinct images) and 0 co-occurrences with 'not disturbing', while 'not scary' has 5 co-occurrences with 'disturbing' and 32 co-occurrences with 'not disturbing'.

Once the co-occurrence counts have been updated, they are used to predict m positive and m negative data points to augment the new data point. m is calculated as $\lfloor pn \rfloor$ where p is a prediction coefficient and n is defined above. For this paper, p is set to 0.3. These predicted data points are not added to the database.

To predict new data points for the given adjective, a, the system first calculates each of the likelihoods that an image will be labeled with a or $\neg a$ given that the image is labeled positively or negatively with each of the adjectives, a_i, in A, the set of all relevant adjectives. Likelihood is calculated as:

$$L(a|a_i) = \frac{co(a, a_i)}{supp(a_i)} \quad (2)$$

where $co(a, a_i)$ is the co-occurrence count for a and a_i, and $supp(a_i)$ is the support of a_i (i.e. number of distinct images labeled with a_i).

Predicted data points for a are chosen using two *probability distributions* created from the above likelihoods, one for positive data points and the other for negative. The positive probability distribution is created by choosing the set of likelihoods, Λ_+, that is the set of all likelihoods described with $L(a|a_i)$ and $L(a|\neg a_i)$ that are greater than some threshold, γ, and less than 1. In this paper, γ is set to 0.4. A likelihood of 1 is omitted because it is guaranteed that there will be no new images to predict with label a. The positive probability distribution is then created by normalizing Λ_+. The negative probability distribution is created in the same way except using the set of all likelihoods, Λ_-, described with $L(\neg a|a_i)$ and $L(\neg a|\neg a_i)$ satisfying the same conditions.

For each data point to be predicted, a likelihood distribution from either Λ_+ or Λ_- is selected using the above probability distributions. Then an image is selected, using a uniform distribution, from all those images with the likelihood's label (either a_i or $\neg a_i$) that are not labeled with a. The label for the new predicted data point is a, the sentiment is the sentiment of the distribution Λ, and the features are the image features of the selected image.

Informally, data points are predicted by assuming that images labeled with adjectives that frequently co-occur with a given adjective, can also be labeled with the given adjective.

Artificial Neural Networks Once recycled and predicted data points for a particular incoming data point are selected, they are shuffled with the incoming data point and given as inputs into the appropriate neural network. The incoming data point then immediately becomes available as historical data for subsequent training data. This process is repeated for each new data point introduced to the learner. Assuming that there is sufficient data, each new data point will be accompanied by a total of $2n + 2m$ data points. In the case of this paper, that's 260 recycled or predicted data points evenly balanced between positive and negative sentiments.

As previously mentioned, one binary artificial neural network is created for each relevant adjective. These neural networks have 102 input nodes for the image features previously described. For this research, based on preliminary experimentation, the neural networks have 10 hidden nodes, a learning rate of 0.01, and a momentum of 0.1.

When the VLA is accessed for the adjective matching metric, the candidate artifact being evaluated is analyzed by extracting the 102 image features. These features are then presented to the appropriate neural network and the output is used as the actual metric. Thus, as Baluja and Machado et al. have done previously, we essentially build and use a model of human appreciation to guide the creation process so that we will hopefully produce images that humans can value (Baluja, Pomerleau, and Jochem 1994; Machado, Romero, and Manaris 2007). Unlike Baluja and Machado however, our model associates images with language and meaning (adjectives), an important step in building a system that communicates intention with its artifacts.

Similarity

The similarity metric borrows from the growing research on *bag-of-visual-word* models (Csurka et al. 2004; Sivic et al. 2005) to analyze local features rather than global ones as we have done previously (Norton, Heath, and Ventura 2011b). Typically, these local features are descriptions of points in an image that are the most surprising, or said another way, the least predictable. After such an interest point is identified, it is described with a vector of features obtained by analyzing the region surrounding the point. *Visual words* are quantized local features. A dictionary of visual words is defined for a domain by extracting local interest points from a large number of representative images and then clustering them (typically with k-means) by their features into k clusters, where k is the desired dictionary size. With this dictionary, visual words can be extracted from any image by determining to which clusters the image's local interest points belong. A bag-of-visual-words for the image can then be created by organizing the visual word counts for the image into a fixed vector. This model is analogous to the bag-of-words construct for text documents in natural language processing. These fixed vectors can then be compared to determine image similarity.

For the similarity metric used in this paper, we use the standard SURF (Speeded-Up Robust Features) detector and descriptor to extract interest points and their features from images (Bay et al. 2008). SURF quickly identifies interest points using an approximation of the difference of Gaussians function, which will often identify corners and distinct edges within images. To describe each interest point, SURF first assigns an orientation to the interest point based on surrounding gradients. Then, relative to this orientation, SURF creates a 64 element feature vector by summing both the values and magnitudes of Haar wavelet responses in the horizontal and vertical directions for each square of a four by four grid centered on the point.

We build our visual word dictionary by extracting these SURF features from more than 2000 images taken from the database of images we've collected to train DARCI. The resulting interest points are then clustered into a dictionary of 1000 visual words using Elkan k-means (Elkan 2003).

Similarity is determined by comparing candidate artifacts with the source image. We create a normalized bag-of-visual-words for the source image and each candidate artifact using our dictionary, and then calculate the *angular similarity* between these two vectors. Angular similarity between two vectors, A and B, is calculated as follows:

$$similarity = 1 - \frac{cos^{-1}(\frac{A \cdot B}{\|A\|\|B\|})}{\pi} \quad (3)$$

This metric effectively measures the number of interest points that coincide between the two images by comparing the angle between vectors A and B. In text analysis, *cosine similarity* (the parenthetical expression contained in Equation 3) is typically used to compare the similarity of documents. With this metric, as the sparseness of vectors increases, the similarity between arbitrary vectors approaches 0. In our case, as vectors are quite sparse, artifacts that are even slightly different from the source would have low scores using this measure. Nevertheless, creating renderings that are very similar to the source image is trivial as it requires simply using fewer and less severe filters. Thus, despite encountering low scores from only small differences, the genetic algorithm would be able to easily converge to near perfect or even perfect scores. This interplay between a harsh similarity metric and relative ease of convergence would place too much weight on the similarity metric. In fact, auxiliary experiments have shown that when using cosine similarity, the adjective matching metric is almost ignored in artifact production.

Since the bag-of-visual-word vectors can only contain positive values, using angular similarity instead of cosine similarity naturally constrains the output to between 0.5 and 1.0. This smaller spread in potential scores significantly reduces the negative impact of sudden jumps in similarity score due to small changes in the candidate renderings. It should be noted that in cases where a candidate artifact has no detected interest features ($\|B\| = 0$), the similarity will default to 0. This is the only case where the similarity score can be below 0.5 as the metric cannot make a comparison.

Experimental Design

Six fitness functions are explored in this paper. They are referred to as *similarity*, *adjective*, *average*, *minimum*, *alternate*, and *converge*. *Similarity* and *adjective* are the similarity and adjective matching metrics in isolation. The other four combine these two conflicting metrics in different ways. *Average* is the approach we have used in the past. With this approach, the two metrics are averaged together with equal weight. With *minimum*, the fitness function is the minimum of the metrics. *Alternate* uses one metric at a time for the fitness function, but it alternates between the two every generation beginning with adjective matching. Finally, *converge* also uses one metric at a time; however, it alternates every 20 generations also beginning with adjective matching.

The two conflicting metrics result in a process that is arguably transformational in nature, at least to a limited degree. Boden describes transformational creativity as that which transforms the conceptual space of a domain (Boden 1999). While the space of possible artifacts cannot change (the filters available for rendering images do not change), the evaluation of the artifacts does change through the interplay of the two metrics. This interplay occurs organically in the *minimum* fitness function by forcing the system to emphasize the metric that it is struggling to optimize at any given epoch during the evolutionary algorithm. The interplay of divergent metrics occurs more mechanically in the *alternate* and *converge* fitness functions by scheduling the emphasis; however, the sudden shift in metric could result in more *unexpected* results, a criterion of creativity emphasized by Maher (Maher 2010; Maher, Brady, and Fisher 2013). The scheduled approaches were inspired by Dipaola and Gabora's work with "Evolving Darwin's Gaze", an installation that also evolves images under two shifting criteria (DiPaola and Gabora 2009). Their criteria are a pixel matching metric comparing artifacts to a specific portrait of Charles Darwin, and an artistic heuristic. We anticipate that our less

Figure 1: The three source images used in all experiments. Images A and C have resolutions of 1600x1200. Image B has a resolution of 1920x1200.

restrictive metrics will ultimately allow for even more surprise and variation in artifacts, while also communicating meaning (adjectives).

Each of the above fitness functions except for *similarity* was run on three source images across five adjectives for a total of fifteen experiments per approach. *Similarity* was only run once for each source image since no adjective was needed. For algorithmic efficiency, the artifacts produced in the experiments were scaled down to a maximum width of 800 pixels. Each experiment ran for 100 generations.

The five adjectives used were 'happy', 'sad', 'fiery', 'wet', and 'peaceful'. These were chosen because they were well represented in our adjective matching training data and because they depict a range of distinct meanings and emotional valence. The three source images (referred to as images A, B, and C) are shown in Figure 1 with their corresponding resolutions.

As mentioned previously, optimizing to the similarity metric alone is trivial for the genetic algorithm since it need only remove filters to do so. However, there is no such trivial approach to optimize to the adjective metric. Historically, near perfect similarity scores are common, while near perfect adjective matching scores are non-existent. In order to balance the quality of the two metrics in our experiments, the source images were not scaled down to match the resolution of the artifacts. A source image and its otherwise unaltered counter part will yield similar but not identical visual-bags-of-words when analyzed for the similarity metric. This means that the genetic algorithm will no longer be able to trivially achieve perfect similarity. The similarity scores of each source image compared to the scaled down version of itself are, for images A, B, and C respectively: 0.826, 0.739, and 0.843 with an average score of 0.803. This means that for our experiments, the range of similarity is now more or less between 0.5 and 0.803—with a now soft ceiling. This is much closer to the range we have seen from adjective matching in auxiliary experiments: 0.144 to 0.714.

Results

In this section we will discuss DARCI's artifact selection for each experiment. While all interpretations of the images themselves are clearly subjective, we attempt to be conservative and consistent in our observations. We will discuss the artifacts in terms of the objectives of the image rendering subsystem: to depict the source image and adjective together in an *interesting* way. By *interesting* we specifically

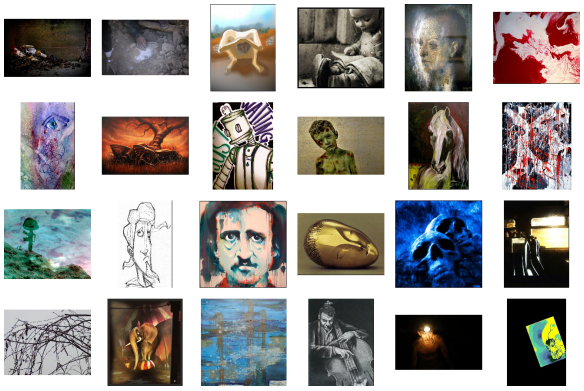


Figure 2: Sample 'sad' images from training data.

mean that extensive filtering (more than basic color filtering or use of inconspicuous filters) has occurred without removing all trace of the source image. Any hint of the source image will be considered acceptable in attributing *interest* to an artifact.

This definition of *interesting* is derived from two commonly proposed requirements for creativity applied to the specific goal of DARCI's image rendering subsystem. These two requirements are, as defined by the American Psychological Association, functionality and originality; or, as Boden described them for the domain of computation, quality and novelty (Boden 1999). Since the purpose of the image renderer is to alter a source image, elimination of the source image would not be functional. Ritchie describes a related requirement that is also applicable here—that of typicality (Ritchie 2007). Ritchie defines typicality as the extent to which an artifact is an example of its intended class. In our case this would be a rendering of a source image as opposed to an entirely new image. The second requirement, novelty, requires that the image renderer produce renderings that are distinctive. Thus, minor or no changes to a source image would clearly suggest a failure at novelty. In an attempt to reduce the amount of subjectivity in our analysis, DARCI's artifacts are either *interesting* by this definition or not. There is no attempt to rate the degree of *interest*.

In addition to being *interesting*, DARCI's artifacts must match the intended adjective. In order to be as objective as possible, we will compare DARCI's artifacts to images from the VLA training data for each given adjective. These images are representative of the types of images one would find if searching google images for a specific adjective. Examples of these images can be found in Figures 2-6. Since DARCI is rendering, as opposed to composing, and due to the limitations of DARCI's image analysis features (and indeed the limitations of the entire field of computer vision), we will be looking for similarities in color, light, and texture as opposed to similar object content.

The 'sad' training images (Figure 2) tend to be desaturated, even black and white, and/or dark with an emphasis on dull colors. The 'happy' training images (Figure 3) trend towards bright and colorful, often containing a full spectrum of colors. The 'fiery' training images (Figure 4)

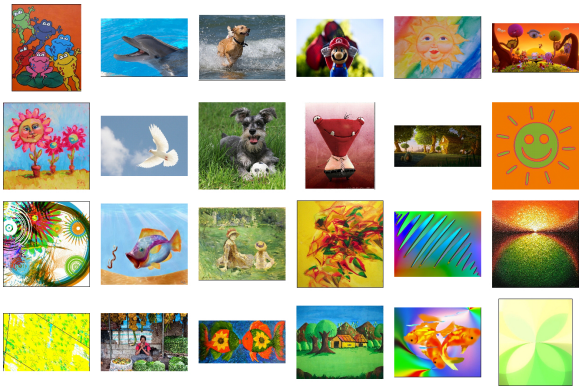


Figure 3: Sample ‘happy’ images from training data.

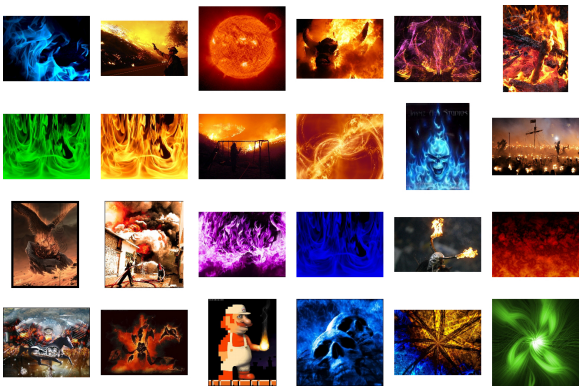


Figure 4: Sample ‘fiery’ images from training data.

usually have distinct flame textures, are bright, and most are monochromatic—typically orange. The ‘wet’ training images (Figure 5) consist of cool colors, usually blue, and have frequent specular highlights and/or wavy patterns. Finally, the ‘peaceful’ training images (Figure 6) contain a variety of soft or pastel colors with a lot of smooth textures.

Ideally, the most fit artifact discovered by the genetic algorithm should be the one that best satisfies the objectives for object rendering outlined above. Thus, for most of the fitness functions, we used this method of selection. However, we anticipated that for two of the fitness functions, *alternate* and *converge* this would not be an appropriate approach. The reason for this is that both of these fitness functions only use one metric at a time, meaning that the most fit artifact discovered could only have been optimized for a single metric. The expected result would be the same as a selection from one of the control fitness functions—not an ideal balance of metrics.

We will first discuss the results of the fitness functions that use the most-fit selection process: *similarity*, *adjective*, *average*, and *minimum*. Later we will discuss *alternate* and *converge* using a different selection criteria. We will evaluate each selection process by the proportion of artifacts that meet the *interest* and adjective matching requirements.

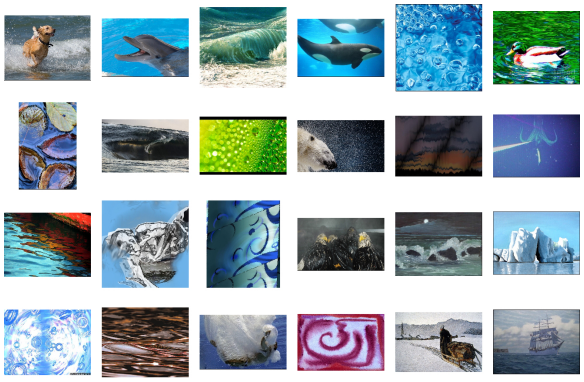


Figure 5: Sample ‘wet’ images from training data.

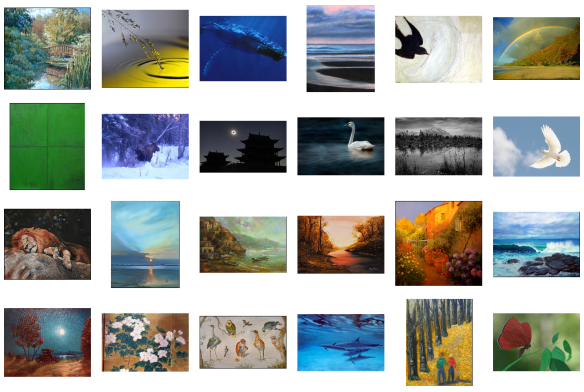


Figure 6: Sample ‘peaceful’ images from training data.

Most Fit Selection

The most fit artifact discovered for each source image in the *similarity* control experiments is shown in Figure 7. The most fit artifact discovered in each of the other experiments is shown in Figures 8-12.

First looking at the *similarity* results (Figure 7), we see that with the exception of image A, DARCI did not select nearly identical images as we might have expected. This illustrates the effect of not scaling the source images. The chosen artifacts actually had slightly higher fitness scores than the strictly scaled down source images demonstrated earlier. For comparison, the fitness score of each of these artifacts is, for artifacts produced from images A, B, and C respectively: 0.836, 0.762, and 0.860 with an average score

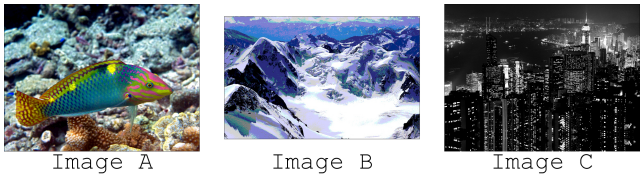


Figure 7: The most fit artifacts for each indicated source image discovered using the *similarity* fitness function.

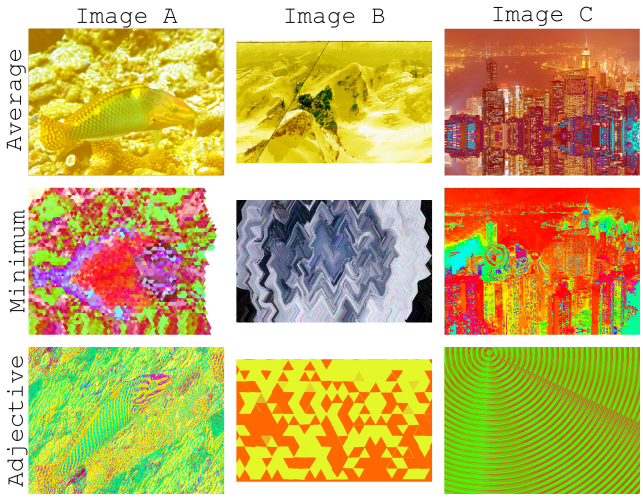


Figure 8: The most fit artifacts for each indicated source image and fitness function for the adjective 'happy'.

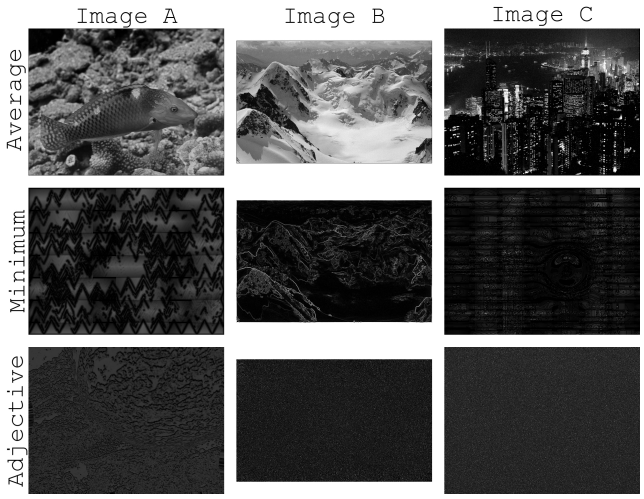


Figure 9: The most fit artifacts for each indicated source image and fitness function for the adjective 'sad'.

of 0.820. That being said, these artifacts are still quite close to the source images, and any resemblances to any of the specified adjectives are obviously happenstance.

For the *average* fitness function, arguably all three of the 'happy' images convey their adjective by applying bright colored filters (Figure 8). All three of the 'sad' images are made more sad by converting them to dark black and white images (Figure 9). Two out of the three 'fiery' images are fiery by primarily coloring with oranges and reds (Figure 10). Image B also looks bright and molten in texture, and some of the buildings in the background of image C almost look on fire. All three 'wet' images are debatably wet, mostly by implementing blue filters (Figure 11). Although, Image B actually looks like it is being viewed through a window soaked during a downpour. None of the 'peaceful' images look any more peaceful than their sources; and very little if anything has changed (Figure 12). With the odd exception of the 'peaceful' images, *average* does quite well at conveying adjectives; however, most of the images don't use much more than simple color filters to do so. In our estimation, for the *average* artifacts, 'happy' B and C, 'fiery' B and C, and 'wet' B satisfy the objectives for object rendering as outlined earlier.

For the *minimum* fitness function, two of the 'happy' images, A and C, are made happy by incorporating many bright colors. Image A looks kaleidoscopic and image C has some rainbow effects. Image B seems out of place, though close inspection will reveal that it may have received a high fitness because of many bright colors as well. While perhaps difficult to notice at first, both image A and B maintain the presence of the source image. All of the 'sad' images are quite dark, suggesting sadness. Image A and C may look like they have eliminated the source images, but the vague shape of the fish is visible within the squiggles of image A, and close inspection of image C will reveal many of the city lights behind the heavy distortion. The three 'fiery' images could be considered 'fiery'. Image A literally looks on fire and image C looks molten. All three 'wet' images appear wet; as with *average*, this is primarily accomplished by making the images blue. Image B does look like the image is now reflected off of a lake, and image C is a bit bleary and wavy giving it ever so slightly the look of being underwater. With the exception of image A, the 'peaceful' images aren't even recognizable, nor do they look peaceful in the way 'peaceful' is reflected in the training images. We're beginning to get a sense of how DARCI interprets 'peaceful' though. In our estimation, of the *minimum* images, 'happy' A and C, all 'sad' and 'fiery' images, and 'wet' B and C satisfy the objectives for object rendering. While 'happy' B and 'peaceful' A are *interesting* representations of the source image, they do not convey the adjective properly.

In the case of the *adjective* fitness function, we see that with three exceptions ('happy' A, 'sad' A, and peaceful 'C'), the source image is undetectable. 'Happy' A and 'sad' A do fit their adjectives, but 'peaceful' C does not. Interestingly, in our estimation *adjective* does not depict the given adjectives as well as *average* or *minimum*. This can be attributed in part to the system exploiting the VLA's neural networks with extreme and unnatural image features.

With all three of these fitness functions, we have seen unsatisfactory performance with 'peaceful'. However, this poor performance goes beyond DARCI's strange interpretation of what makes an image 'peaceful' (apparently being purple and noisy). That can be attributed to inadequate learning by the VLA, perhaps because of limited available training data. One could even make the case for it being a creative expression of 'peaceful'. The other problem here is the fact that for 'peaceful' artifacts, the three *average* artifacts were virtually unmodified from the source image, and that two of the *minimum* artifacts completely obfuscated the source image. This issue can be explained by a problematic interaction between the similarity and adjective matching metrics for 'peaceful'.

The 'peaceful' neural network output has very low vari-

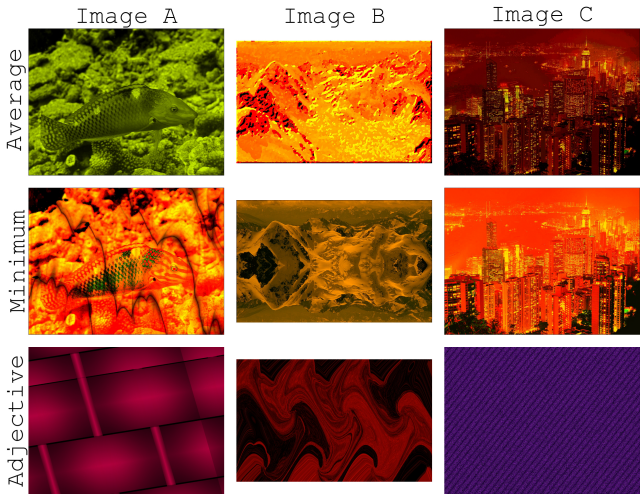


Figure 10: The most fit artifacts for each indicated source image and fitness function for the adjective 'fiery'.

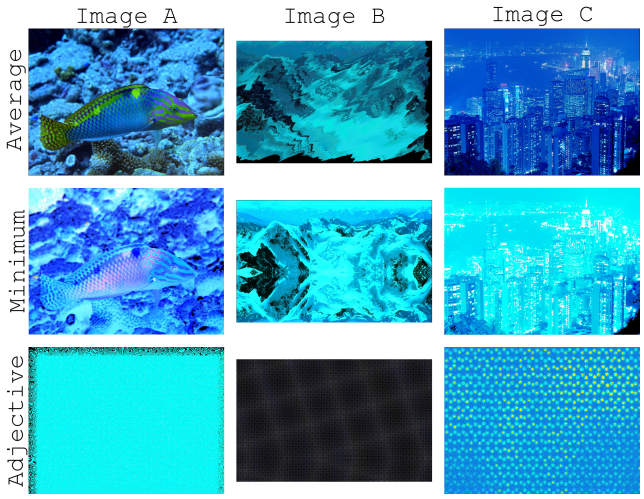


Figure 11: The most fit artifacts for each indicated source image and fitness function for the adjective 'wet'.

ance compared to the other neural networks, and a mean slightly under 0.5. The variance is so low that the highest 'peaceful' neural network outputs encountered are not much higher than the lowest similarity score possible (0.5). Thus, the *minimum* fitness function is effectively acting like the *adjective* fitness function for 'peaceful'. In the case of *average*, the variance is so low that the smallest changes in similarity still overshadow any changes in adjective matching. This example illustrates that despite our best efforts to balance the two metrics, incongruities between the two can still occur. Thus, for future work, a dynamic solution that takes into consideration certain statistics about each metric may be in order.

Selection After Last Shift

As indicated earlier, the *alternate* and *converge* fitness functions need a different selection method than that used above. As suspected, using most-fit selection resulted in artifacts that were either similar to those in Figure 7 or completely abstract like the images produced with *adjective*. The assumption with *alternate* and *converge* is that even though only a single metric is in effect at each generation, the genetic algorithm will not be able to converge to either because of constant shifts in the metric, and will instead find an interesting and unexpected solution.

With this in mind, the selection criteria that we use here is to pick the most fit artifact from the last *shift* in metric. This is the point at which we would expect to find the most surprising artifacts. We define a *shift* in the metric as the changing from the similarity metric to the adjective matching metric or vice versa. For *alternate* this is the shift from similarity to adjective matching at generation 100 which we will call *alternate-adjective*, and for *converge* it is the shift from adjective matching to similarity also at generation 100 which we will call *converge-similarity*. Since the direction of the shift may strongly effect the outcome, we have also selected the most fit artifact from generation 99 for *alternate* (adjective matching to similarity) and generation 80 for *converge* (similarity to adjective matching). We will call these two approaches respectively *alternate-similarity* and *converge-adjective*.

The results of these experiments are in Figures 13 to 15. In the interest of space, we do curate these images by only showing those artifacts that are neither over nor under-filtered (i.e. *interesting*) based on observations similar to those made for the earlier experiments. In the case of *alternate-similarity*, there were no artifacts produced that weren't under-filtered. Most had tinting or small distortions, but none were *interesting*.

Figure 13 shows *interesting* artifacts that were selected with *alternate-adjective*. This particular fitness function and selection criteria yielded the most numerous *interesting* artifacts of the four configurations. In this case, all but one of the not-shown artifacts were too abstract. Of the remaining *interesting* artifacts, all but the unusual 'peaceful' images arguably convey the intended adjectives.

Next, Figure 14 shows *interesting* artifacts selected with *converge-adjective*. Most of the other artifacts selected obfuscated the source image too much. Here, with the exception of 'fiery' A and perhaps 'fiery' B, the images convey the intended adjectives.

Finally, Figure 15 shows the *interesting* artifacts selected with *converge-similarity*. While the images shown are adequately *interesting*, we don't consider them as distinguished as those in the previous two examples. All of the other artifacts were too similar to the source image to warrant display. All of the displayed artifacts do convey the given adjectives.

Filter Sequence Length

Functionally, much of the quality of an artifact can be attributed to the length of the artifact's genotype. The genotype is the "genetic" encoding of the artifact, and in the image rendering subsystem is a sequence of image filters. The more filters used to render a source image, the more likely

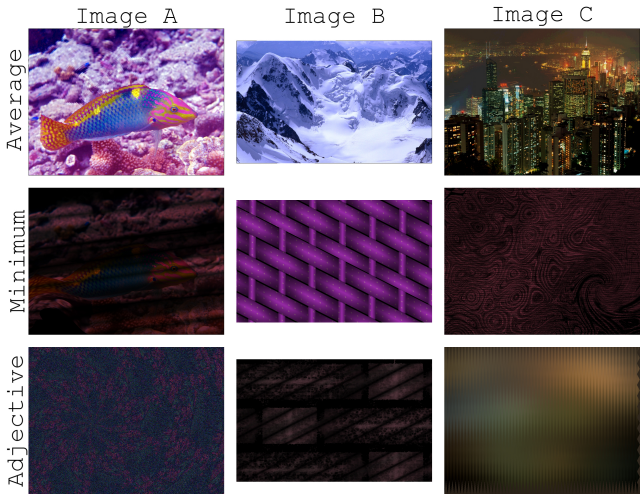


Figure 12: The most fit artifacts for each indicated source image and fitness function for the adjective 'peaceful'.

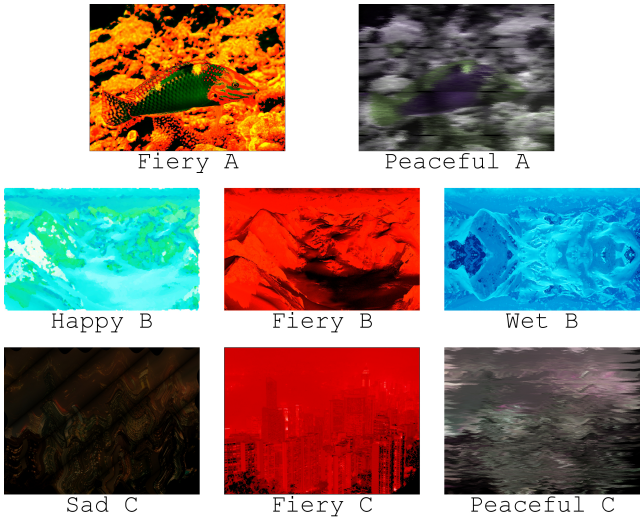


Figure 13: Artifacts selected for the indicated source images and adjectives for the *alternate-adjective* fitness function.

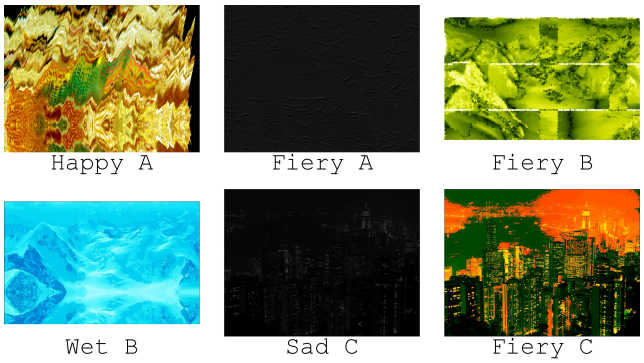


Figure 14: Artifacts selected for the indicated source images and adjectives for the *converge-adjective* fitness function.

the artifact will become abstract. The fewer filters used, the more likely the artifact will not deviate from the source image. Figure 16 shows the average genotype length (in number of filters) for each fitness function explored in this paper over the 100 epochs of evolution. The top performing fitness functions show a comfortable balance between too many and too few filters. *Minimum* does this the best.

Conclusions

The motivation behind this work has been to improve DARCI's ability to independently curate its own artifacts. All of the artifacts displayed in this paper were fully curated by DARCI under various selection criteria, with only a few indicated exceptions for space.

We show that DARCI is autonomously able to consistently create and select images that reflect the requested adjective with four out of five adjectives. This demonstrates the quality of the new adjective matching metric. We also demonstrate that the similarity metric functions as intended.

We explored a variety of fitness functions combining two metrics with varying degrees of success. Each method of combining the metrics had its own biases but, from our analysis, the *minimum* fitness function performed the best. Over half of the artifacts selected with this fitness function satisfied the goals of the image rendering subsystem—arguably a significant step in decreasing the latent heat effect in DARCI. We attribute the success of *minimum* to the fact that it allows the genetic algorithm to naturally shift evolutionary focus to the metric that is suffering the most.

We are confident that the improvements made to the image rendering subsystem in this paper will significantly decrease the latent heat effect in DARCI. We intend to test this theory in the future by conducting a thorough online survey comparing this improved version of DARCI to other versions, and perhaps even to humans. To further improve the image rendering subsystem described in this paper, we also intend to pursue more adaptable variations of the metrics outlined here. Metrics that will adapt their output in response to the features of other metrics.

References

Baluja, S.; Pomerleau, D.; and Jochem, T. 1994. Towards automated artificial evolution for computer-generated images. *Connection Science* 6:325–354.

Bay, H.; Ess, A.; Tuytelaars, T.; and Gool, L. V. 2008. Speeded-up robust features (SURF). *Computer Vision and Image Understanding* 110:346–359.

Boden, M. A. 1999. *Handbook of Creativity*. Press Syndicate of the University of Cambridge. chapter 18.

Boden, M. A. 2004. *The Creative Mind: Myths and Mechanisms (second edition)*. Routledge.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier. In *20th European Conference on Artificial Intelligence*, 21–26.

Colton, S. 2008. Creativity versus the perception of creativ-

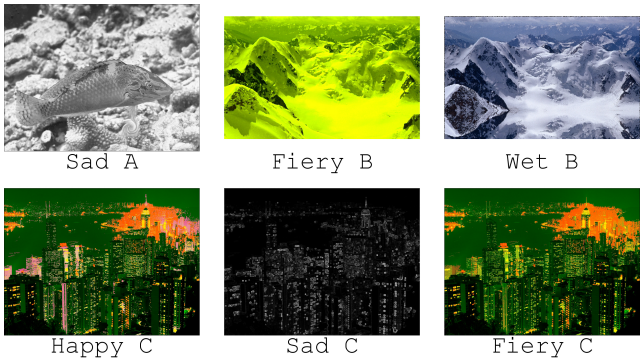


Figure 15: Artifacts selected for the indicated source images and adjectives for the *converge-similarity* fitness function.

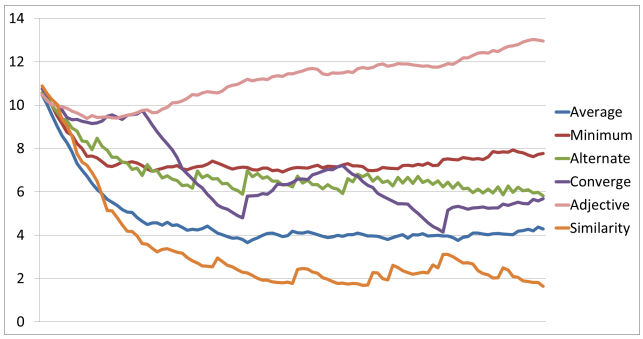


Figure 16: The average lengths of genotypes across all fitness functions over 100 epochs of evolution.

ity in computational systems. *Creative Intelligent Systems: Papers from the AAAI Spring Symposium* 14–20.

Csurka, G.; Dance, C. R.; Fan, L.; Willamowski, J.; and Bray, C. 2004. Visual categorization with bags of keypoints. In *Proceedings of the Workshop on Statistical Learning in Computer Vision*, 1–22.

Datta, R.; Joshi, D.; Li, J.; and Wang, J. Z. 2006. Studying aesthetics in photographic images using a computational approach. *Lecture Notes in Computer Science* 3953:288–301.

DiPaola, S., and Gabora, L. 2009. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming and Evolvable Machines* 10(2):97–110.

Elkan, C. 2003. Using the triangle inequality to accelerate *k*-means. In *Proceedings of the Twentieth International Conference on Machine Learning*, 147–153.

Fellbaum, C., ed. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Gero, J. S. 1996. Creativity, emergence, and evolution in design. *Knowledge-Based Systems* 9:435–448.

Gevers, T., and Smeulders, A. 2000. Combining color and shape invariant features for image retrieval. *IEEE Transactions on Image Processing* 9:102–119.

Heath, D., and Norton, D. 2009. DARCI (Digital ARtist Communicating Intention). http://darci.cs.byu.edu.

Heath, D.; Norton, D.; and Ventura, D. 2013. Autonomously communicating conceptual knowledge through visual art. In *Proceedings of the 4th International Conference on Computational Creativity*, 97–104.

King, I.; Ng, C. H.; and Sia, K. C. 2004. Distributed content-based visual information retrieval system on peer-to-pear network. *ACM Transactions on Information Systems* 22(3):477–501.

Li, C., and Chen, T. 2009. Aesthetic visual quality assessment of paintings. *IEEE Journal of Selected Topics in Signal Processing* 3:236–252.

Machado, P.; Romero, J.; and Manaris, B. 2007. Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. In Romero, J., and Machado, P., eds., *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Berlin: Springer. 381–415.

Maher, M. L.; Brady, K.; and Fisher, D. H. 2013. Computational models of surprise as a mechanism for evaluating creative design. In *Proceedings of the 4th International Conference on Computational Creativity*, 147–151.

Maher, M. L. 2010. Evaluating creativity in humans, computers, and collectively intelligent systems. In *DESIRE '10 Proceedings of the 1st DESIRE Network Conference on Creativity and Innovation in Design*, 22–28.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing appreciation in a creative system. In *Proceedings of the 1st International Conference on Computational Creativity*, 26–35.

Norton, D.; Heath, D.; and Ventura, D. 2011a. An artistic dialogue with the artificial. In *Proceedings of the 8th ACM Conference on Creativity and Cognition*, 31–40. New York, NY, USA: ACM.

Norton, D.; Heath, D.; and Ventura, D. 2011b. Autonomously creating quality images. In *Proceedings of the 2nd International Conference on Computational Creativity*, 10–15.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *Journal of Creative Behavior* 47(2):106–124.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.

Sivic, J.; Russell, B. C.; Efros, A. A.; Zisserman, A.; and Freeman, W. T. 2005. Discovering objects and their location in images. *International Journal of Computer Vision* 1:370–377.

Wang, W.-N.; Yu, Y.-L.; and Jiang, S.-M. 2006. Image retrieval by emotional semantics: A study of emotional space and feature extraction. *IEEE International Conference on Systems, Man, and Cybernetics* 4:3534–3539.

Zujovic, J.; Gandy, L.; and Friedman, S. 2007. Identifying painting genre using neural networks. *miscellaneous*.

Automated Daily Production of Evolutionary Audio Visual Art — An Experimental Practice

Tatsuo Unemi
Department of Information Systems Science
Soka University
Hachiōji, Tokyo 192-8577 Japan
unemi@iss.soka.ac.jp

Abstract

Evolutionary computing based on computational aesthetic measure as fitness criteria is one of the possible methods to let the machine make art. The author developed and set up a computer system that produces ten short animations consisting sequences of abstract images and sound effects everyday. The produced pieces are published on the internet using three methods, movie files, HTML5 + WebGL, and a special application software. The latter two methods provides viewers experiences of a high resolution lossless animation. Their digest versions are also uploaded on a popular web service of movie sharing. It started October 2011. It is still in an experimental level that we need to brush up, but it has not always but often succeeded to engage the viewers.

Introduction

As similarly as the evolutionary process in the nature has produced huge number of complex variations of unique species on the earth, evolutionary computing has a capability to produce unpredictable designs by the computer. As the nature often provides us experiences of beautiful audio visual stimuli, the computer has a potential capability to produce beautiful images and sounds if we set up a combinatorial search space in the machine that contains masterpieces. We can find a lot of technical variations for such approach under the name of "Generative art (Galanter 2003; Pearson 2011)." The design of computational aesthetic measures is very important to realize an efficient search in the huge space. It would act as a skill of a genius photographer who can find amazing scenery in the nature to be captured by his/her camera. It is easy for the computer to generate huge number of audio visual patterns by exhaustive search, but almost all of the products would be trash without an appropriate measure.

Though development of the computable models for aesthetic measures comparable with the human artists is on the long way of challenge in the research field of computational creativity, some of the methods has already been examined in the experimental activities by a number of researchers, such as (Machado and Cardoso 2002; Ross, Ralph, and Zong 2006; den Heijer and Eiben 2010). The author has also developed an experimental system of evolutionary computing that automatically produces art pieces, combining ideas of preceding researches and his own ideas (Unemi 2012a). Owing to the recent improvement of computational power of graphical processing unit (GPU) on the personal computer, it became possible to use this type of system for realtime production of non-stop sequence of short animations on site (Unemi 2013). At the same time, it is also possible to set up a machine to make automatic production everyday without any assistance by human.

This paper introduces the author's project named "Daily Evolutionary Animation" that started October, 2011. The following sections describe a summary of evolutionary process, aesthetic measures employed, daily production process, and a public showing on the internet. In the final section, we discuss future extensions along this project.

Summary of evolutionary process

The author developed SBArt (Unemi 2009) originally as a tool to breed a visual evolutionary art using a mechanism of interactive evolutionary computation (Takagi 2001). The first version that runs on UNIX workstation was released in the public domain on the internet in 1993. It is based on a similar mechanism of the pioneering work by (Sims 1991) that uses a tree structure of mathematical expression as the genotype. The expression is a function that maps (x, y, t) coordinate of spaciotemporal space to a color space of hue, saturation and brightness. The spacial coordinate (x, y) is used to indicate the pixel in the image, and the temporal coordinate t indicates the frame position in the movie. Each expression is organized by the terminal symbols and non-terminal symbols. A terminal symbol expresses a value of three dimensional vector. It is a constant containing three scalar values or a permutation of three variables, x, y and t. A non-terminal symbol is a unary or binary operator that takes three dimensional vector for each argument and result value. We prepared nine unary functions including minus sign, absolute value, trigonometric functions, exponential functions, and so on; and ten binary functions including addition, subtraction, multiplication, division, power, and so on. Two selective functions that return one of two arguments choosing by comparison of the first elements are effective to compose a collage of different patterns. Each genotype is used to draw the phenotype by determining the color values distributed in a volume of movie data. The computational cost depends on

the resolution of both space and time because it must calculate a three dimensional value for each pixel.

As an extension of the system, an automated process of evolution was implemented as described in (Unemi 2012a). Evolutionary process is conducted in a manner of minimal generation gap method (Satoh, Ono, and Kobayashi 1997) that produces only two offsprings from randomly selected parents in each computing step. The genetic reproduction is done in a style of genetic programming (Koza 1992) using subtree exchange for crossover and symbol replacement for mutation. To prevent infinite extension of the length of genotype through the iteration of genetic operations, the maximum number of symbols in a single genotype is restricted within 120. The fitness values are calculated based on aesthetic measures described in the next section.

It used to take some seconds to render a single frame image for movie production in 2001, but it became possible to render an animation in realtime by using the parallel processing of GPU. We revised the software so that it uses Core Image Framework by compiling the expression into Core Image Kernel Language to take advantage of GPU's power (Unemi 2010). It is a dialect of shading language GLSL in OpenGL working on MacOS X.

Aesthetic measures

It might be an ultimate goal of the research on computational creativity to implement a computable procedure that evaluates how a pattern is beautiful as a delegate of human critics. Many artists and scientists have been struggling with this difficult and interesting theme from several points of views as summarized in (Galanter 2012). It is obvious that the human's decision on aesthetics is depending on his/her own both private and social experiences, but it is also affected by physical functionalities of our sensory organs and fundamental signal processing in the brain widely shared among humans beyond the differences in cultures and races. Some of these measures in a level of perception should match with a mathematical theory of complexity and fluctuation.

We implemented three for each measure on geometric arrangement and on distribution of micro features for a still image, that is,

1. pseudo complexity measure utilizing JPEG compression,
2. global contrast factor in color image,
3. distribution of gradient angles of brightness,
4. frequency distribution of hue values,
5. frequency distribution of brightness and
6. average and variance of saturation values.

The detail of procedure for each measurement and auxiliary normalization are described in (Unemi 2012a). All of these procedures are relatively easy to implement utilizing well known technics of image processing.

The method 1. is a convenient approximation of complexity originally used in (Machado, Romero, and Manaris 2007). The evaluation is done by calculating a ratio between the compression ratio and the ideal value the user specified. 2. is a modified version of the factor proposed by (Matkovic et al. 2005). The original version takes a gray scale image to calculate the differences of brightness between each pair of adjacent pixels in multiple resolution, but we extended it to be applicable for a color image by replacing the difference of brightness with the distance in the color space.

For 4. and 5., there are a number of hypotheses and investigations on a frequency distribution of different types of features observed in phenomena happened in both nature and human society, such as pressure of natural wind, sound frequencies from a stream, populations of cities, note pitches of music, and so on. One of the well-known hypothesis is *power law* on which we can find a number of samples in (Newman 2006), for example. (den Heijer and Eiben 2010) is employing *Benford's law*, a similar shape of distribution with the power law, as one of the factors to measure the aesthetic value. We use a distribution extracted from one thousand snap photos of portraits and natural sceneries as the ideal distribution, that is approximately similar to the power law. 6. is a subject to be adjusted following the user's preference, colorful or monotone. We used a parameter setup for relatively psychedelic results at the start time, but changed it for more grayish results some months later, in order to make the results give the viewer weaker visual stimuli. The geometric mean among these measures is taken as the total evaluation of a single frame image.

To evaluate a movie, the aesthetic measure should be calculated from all of the pixels contained in the three dimensional volume of space and time of colors. However, it is still difficult to complete the calculation within an acceptable time for all of the data in the final product even using parallel processing on GPU. For example, half a minute of hi-definition movie contains approximately 2 giga pixels. To reduce the computational cost, we uses reduced resolution of 512×384 pixels for each frame image, and picks up only ten frames as the samples. In total, the number of pixels to be calculated is $512 \times 384 \times 10 = 1,966,080$. It is also important to combine an aesthetic measure on motion in animation. We employed a simple method of taking average value of absolute differences between colors of two pixels in the same position of consecutive frames in order to estimate how fast or slow the picture is moving. The point of motion measure is the inverse value of absolute difference with the ideal speed specified by human. The final evaluation is a geometric mean between the average point of still images and the average point of motion measures among sampled frames.

Automated daily production

The functionality of automated evolution has enabled not only an installation of automatic art but also automated production without an assistance by human. From October 6th, 2011, the system has been automatically producing ten movies everyday. The production procedure starts in the morning of Japanese Standard Time, continuing the evolutionary process from a random population until the completion of 200 steps of generation alternation. Starting from 20 randomly generated genotypes, children are added to the population until the population size reaches 80, then replacement starts. To prevent a premature convergence that often

happens in search process in optimization, the population is refreshed by the following procedure for each 50 steps. It

1. picks up the best 15 individuals from the current population,
2. generates five random genotypes,
3. produces 20 individuals by crossover operation from individuals in 1. + 2., and then
4. starts the same process from these 20 individuals as conducted in the first step.

Throughout the process, $20+2\times200+20\times(200/50-1)=500$ (20 in the initial population, two children for each step and 20s in the refreshing procedure for each 50 steps) individuals are examined.

After the completion of 200 steps of evolutional process, the procedure selects the best ten individuals from the final population, and generates a source code of shading language for WebGL and 20-second movie files for each. A synchronized sound effect is also generated without any prerecorded sampled data but purely synthesized sound waves by combination of oscillation and modulation as described in (Unemi 2012b). The parameters of sound wave synthesis are the statistical factors extracted from frame images.

The main machine for the evolutionary production is an old MacPro 2006, equipped with two Intel Xeon dual core processors of 3 GHz, GeForce 7300 GT as GPU and MacOS X 10.6. as the operating system. The elapsed time necessary for the evolutionary process described above is approximately 90 minutes. It would be reduced in less than half if we could arrange it by a newer machine.

The entirety of the daily process is controlled by a program in AppleScript that accesses to application softwares, SBArt4 for evolutionary production, QuickTime Player 7 and X to convert the movie file format and to organize a digest movie, *curl* to submit the digest to YouTube, and *"t"* to announce the completion on twitter. The process is launched as a startup procedure after the machine wakes up at the scheduled time everyday. If no error occurs, the machine shuts down automatically.

Public showing on the internet

To complete the fully automated process by exhibiting the products on the internet, we built three types of user interfaces for viewers based on movie files, HTML5 + WebGL, and a special application software. In all of these methods, the animations are automatically played back in a sequence following the viewer's choice from three alternatives, random, forward and backward. The viewer also allows to directly select the date from the calendar shown in the graphical user interface, and choose one of ten pieces listed as thumbnail images to be played back. Figure 1 shows a sample of the web page to watch the animations distributed as a form of movie files.

Movie files

Each of the produced movie files is compressed in both the H.264 and Ogg Vorbis formats in order to be adaptable for playback by popular web browsers, such as Safari, FireFox,

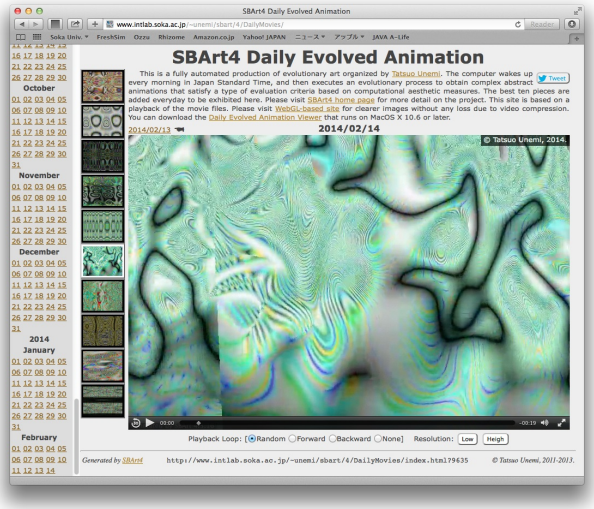


Figure 1: A sample of web page to watch the animations distributed in a form of movie files.

Google Chrome, and Opera. These movies are accessible from `http://www.intlab.soka.ac.jp/~unemi/sbart/4/DailyMovies/`. Reorganization of a web site to adapt to the newly generated movies is also performed automatically just after the compressed movie files are uploaded to the web server. The daily and weekly digests of these movies are also posted to a popular site for movie sharing. A daily digest is a sequence of six-seconds excerpts for each movie, for a total duration of one minute. A weekly digest is a sequence two-seconds excerpts for each of the 70 movies produced in the last seven days. These digests are accessible at `http://www.youtube.com/user/une0ytb/`.

The daily process consumes an average of 346 MB of the storage in the web server everyday, which means that storing all of the movies produced over a number of years on a hard disk drive is feasible, because 126 GB for one year's worth of movies is not unreasonable considering the HDD capacity of currently available consumer products.

HTML5 + WebGL

A drawback of movie file is dilemma between quality and size. Usual environment of an internet user has no enough capability to display an uncompressed sequence of raw images. If we try to transmit uncompressed movie data of VGA (640×480 pixels) in 30 frames per second, the required band width is $640\times480\times30\times3=27,648,000$ bytes per second. It is possible in a local area network with Giga bit channel, but difficult for usual connection beyond the continents toward a personal computer at home. The compression techniques widely used were designed for movies captured by the camera and/or cartoon animation. Because the evolutionary art might contains very complex patterns that is difficult to be compressed efficiently, such methods commonly used are not always effective for this project.

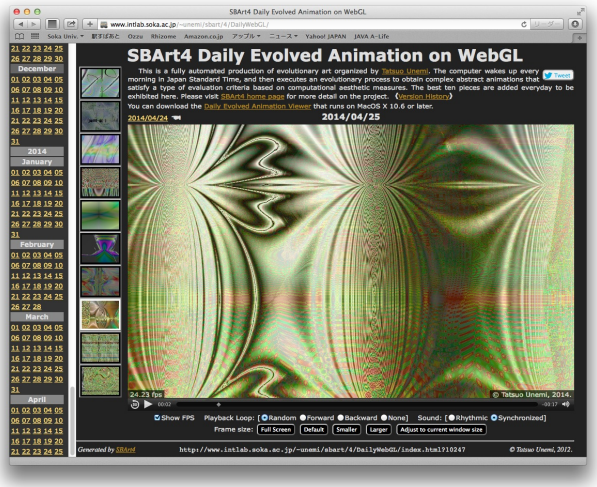


Figure 2: A sample of web page to watch the animations distributed in a form of shading code.

The latest web technology made it possible to let the browser render a complicated graphic image by downloading a script written in JavaScript. The newest specification of HTML5 includes some methods for interactive control of both graphics and network communication. In addition, WebGL is available to render a 3D graphics in a 2D rectangle area of *canvas* object utilizing shading language GLSL ES. It is possible to render an image without any loss by compression if the browser directly draw it based on functional expression produced through evolutionary production. Because SBArt4 is using Core Image Kernel Language to render each image as described above, it is relatively easy to generate a source code of GLSL ES from the genotype. An advantage of shading language is that it is possible to render arbitrary size of image without loss even if it's in the full screen mode of high DPI display. The fastest frame rate is depending on both the power of hardware and the efficiency of JavaScript execution on the browser.

An audio file in high quality is not so heavy in comparison with movie file. JavaScript controls the frame image alternation by checking the progress of audio playback. The average size of audio file in AAC compression is approximately 330 kbytes in 44.1 kHz as sampling rate, 16 bits as sample size, two channels and 20 seconds in duration, for each piece. Because the average size of shading code is 3 kbytes for each piece, the total amount of storage required for the web server is almost one 100th of the case in movie file. The service is available from `http://www.intlab.soka.ac.jp/~unemi/sbart/4/DailyWebGL/`. Figure 2 shows a sample web page to watch the animations distributed in a form of shading code.

Specific application software

In the method using WebGL described in the above section, it sometimes suffers computational bottleneck due to the hardware performance and browser's implementation for

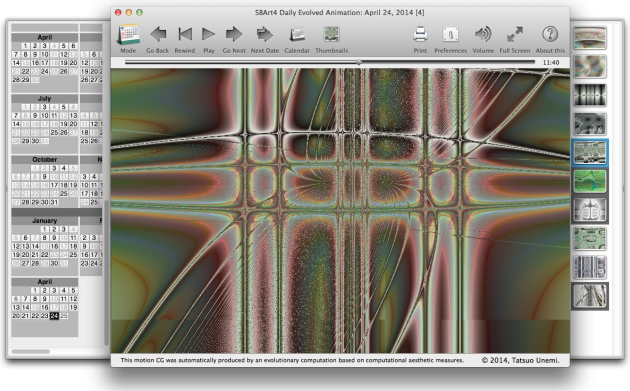


Figure 3: A sample image of a window of special application, DEAViewer, to watch the animations distributed in a form of shading code.

executable scripts. To take full advantage of the power of machine at viewer's side, it is the best way to distribute an application software optimized for viewing the products. We developed a software named *DEAViewer* runnable on OS X 10.6 or later, and are distributing it on Apple's App Store in free of charge. The basic mechanism is almost same with the case of WebGL, but the procedure of control part is directly executed on CPU by compiled machine code without any overhead of either compilation or interpretation of the code. It downloads the same information used in WebGL version, and slightly modifies the shading code to adapt to an efficient GLSL code. The more detail information is at `http://www.intlab.soka.ac.jp/~unemi/sbart/4/deaviewer.html`. It provides a viewer's experience of 30 fps lossless animation on 4K display. Figure 3 shows a sample image of a window of special application, DEAViewer, to watch the animations distributed in a form of shading code.

Future extension

Though it has already passed for two and half years and the number of produced pieces reached 9,500, but we have not conducted any analysis over them so far. In the author's intuitive reflection through those years, it often produces amazing pieces but sometimes not. Almost all of productions, except small number of erroneous failure, obtained higher fitness defined as a type of aesthetic measure we designed. This is a typical evidence why we need more research to pursue a human equivalent ability of evaluation even in a perception level for visual arts, because it suggests that the measures employed here might be necessary but not sufficient.

Of course, there are several candidates of aesthetic measures to be introduced, such as a composition based on golden ratio and/or rule of thirds. If we want to obtain an image that inspires something we know in the physical real world or in popular mythology, the composition is very important though it might be a long way to achieve. It is also

necessary to consider not only on the perception level but also deeper level of understanding by combination of memory retrieval and conceptual inference connected to emotional move. It is of course a big issue in computational creativity to make a machine that creates emotionally impressive piece inspiring something in human mind connected with viewer's private life or social affairs.

An easier extension is on the method of combination among different measures. The system introduced here is using geometric means because we thought all of the measures should be necessary conditions. We should examine another style of combination such as weighted summation, minimum and maximum among them. More complex combination of these logical operations might be effective. It might be also interesting to introduce some methods developed in the field of multi-objective optimization (Deb 2001), as (Ross, Ralph, and Zong 2006; den Heijer and Eiben 2014) examined. An effective method must be introduced to produce pieces of wider variation, if we use more generations, or the eternal evolutionary process, for production.

Another extension we should try in not far future is on the aesthetic measure of motion in the temporal sequence of pictures. We introduced very simple method to estimate the speed of motion in order to reduce the computational cost, but it must be replaced with some statistical analysis based on a type of optical flow. The techniques to extract distribution of 2D vectors of flow in the motion picture are originally developed for detection of the camera movement and an object moving in the captured scenery. But it must be useful to measure the interestingness of motion.

To provide a test bed for the research on computational aesthetic measures, it might be valuable to develop a mechanism of software plug-in to add a third party module for evaluation. It will make it easier to examine and compare the researchers' ideas.

Conclusion

Our experimental project of automated daily production of evolutionary audio visual art was introduced above. We have a lot of tasks to be conducted toward the machine that produces impressive art pieces. The author hopes this project inspires some ideas for the artists and researchers interested in creativity of human and/or machine.

References

Deb, K. 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons.

den Heijer, E., and Eiben, A. E. 2010. Using aesthetic measures to evolve art. In *WCCI 2010 IEEE World Congress on Computational Intelligence*, 4533–4540.

den Heijer, E., and Eiben, A. E. 2014. Investigating aesthetic measures for unsupervised evolutionary art. *Swarm and Evolutionary Computation* 16:52–68.

Galanter, P. 2003. What is generative art? complexity theory as a context for art theory. In *Proceedings of the 6th Generative Art Conference*, 76–99.

Galanter, P. 2012. Computational aesthetic evaluation: Past and future. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*. London, UK: Springer-Verlag. chapter 10.

Koza, J. R. 1992. *Genetic Programming: On The Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Machado, P., and Cardoso, A. 2002. All the truth about NEvAr. *Applied Intelligence* 16:101–118.

Machado, P.; Romero, J.; and Manaris, B. 2007. Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. In Romero, J., and Machado, P., eds., *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Berlin Heidelberg: Springer-Verlag. 381–415.

Matkovic, K.; Neumann, L.; Neumann, A.; Psik, T.; and Purgathofer, W. 2005. Global contrast factor – a new approach to image contrast. In *Computational Aesthetics 2005*, 159–168.

Newman, M. E. J. 2006. Power laws, Pareto distributions and Zipf's law. *arXiv.org* (cond-mat/0412004).

Pearson, M. 2011. *Generative Art: A practical guide using Processing*. Manning Publications.

Ross, B. J.; Ralph, W.; and Zong, H. 2006. Evolutionary image synthesis using a model of aesthetics. In *WCCI 2006 IEEE World Congress on Computational Intelligence*, 3832–3839.

Satoh, H.; Ono, I.; and Kobayashi, S. 1997. A new generation alternation model of genetic algorithms and its assessment. *Journal of Japanese Society for Artificial Intelligence* 12(5):734–744.

Sims, K. 1991. Artificial evolution for computer graphics. *Computer Graphics* 25:319–328.

Takagi, H. 2001. Interactive evolutionary computation: Fusion of the capacities of EC optimization and human evaluation. *Proceesings of the IEEE* 89(9):1275–1296.

Unemi, T. 2009. Simulated breeding: A framework of breeding artifacts on the computer. In Komosinski, M., and Adamatzky, A. A., eds., *Artificial Models in Software*. London, UK: Springer-Verlag, 2 edition. chapter 12.

Unemi, T. 2010. SBArt4 – breeding abstract animations in realtime. In *WCCI 2010 IEEE World Congress on Computational Intelligence*, 4004–4009.

Unemi, T. 2012a. SBArt4 for an automatic evolutionary art. In *WCCI 2012 IEEE World Congress on Computational Intelligence*, 2014–2021.

Unemi, T. 2012b. Synthesis of sound effects for generative animation. In *Proceedings of the 15th Generative Art Conference*, 364–376.

Unemi, T. 2013. Non-stop evolutionary art you are embedded in. In *Proceedings of the 16th Generative Art Conference*, 247–253.

Building Artistic Computer Colleagues with an Enactive Model of Creativity

Nicholas Davis[1], Yanna Popova[2], Ivan Sysoev[1],

Chih-Pin Hsiao[3], Dingtian Zhang[1], Brian Magerko[1]

[1]School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA USA
{ndavis35, ivan.sysoev, alandtzhang, magerko}@gatech.edu

[2]Department of Cognitive Science
Case Western Reserve University
Cleveland, OH USA
yanna.popova@case.edu

[3]College of Architecture
Georgia Institute of Technology
Atlanta, GA USA
chsiao9@gatech.edu

Abstract

This paper reports on the theory, design, and implementation of an artistic computer colleague that improvises and collaborates with human users in real-time. Our system, Drawing Apprentice, is based on existing theories of art, creative cognition, and collaboration synthesized into an enactive model of creativity. The implementation details of the Drawing Apprentice are provided along with early collaborative artwork created with the system. We present the enactive model of creativity as a potential theoretical framework for designing creative systems involving continuous improvisational collaboration between a human and computer.

Introduction

Creative technologies have come a long way in supporting human creativity in a variety of ways. Modern creativity support tools (CST) have been extremely effective at helping users produce higher quality products by allowing them to explore creative possibilities, perform complex simulations, and record and track ideas (Shneiderman 2007). However, with all their capabilities and features, popular creativity support tools like Adobe's Photoshop are not yet able to generate original artistic contributions, such as new lines or brush strokes that add to the user's artwork. Recent advances in artificial intelligence and computational creativity are beginning to change this by developing co-creative computer colleagues to enrich the human creative process in a completely new manner through collaboration with a creative computer (Lubart 2005).

Computer colleagues can bridge the gap between CSTs that support a creative person and computers that generate creative products autonomously (see Figure 1). We hypothesize collaboration with computer colleagues based on the enactive model of creativity can enrich the creative process like human collaboration (i.e. increase playful exploration, motivation, creative engagement) in open-ended creative domains such as non-representational visual art. We have designed and implemented a prototype of an artistic computer colleague using the enactive model of creativity (EMC) to test this hypothesis.

Our system, called Drawing Apprentice applies EMC to abstract improvisational art. This artistic domain was selected for its open-ended, flexible and emergent art patterns (Clouzot 1956). EMC synthesizes several cognitive science and creativity theories to model creativity as an enactive process that emerges through constant interaction with the environment and other agents within it. In this view, creative actions emerge through experimental interactions with the environment based on simulations and perceived artistic affordances rather than executing a fully formed plan and artistic goal.

In the following sections, we first introduce co-creativity in the context of computational creativity and improvisational abstract art. Next, we provide some background on enactive cognition. Then, we present our enactive model of creativity and show how it helped us design an improvisational drawing agent. Finally, we consider evaluation metrics and show early artwork created with the system.

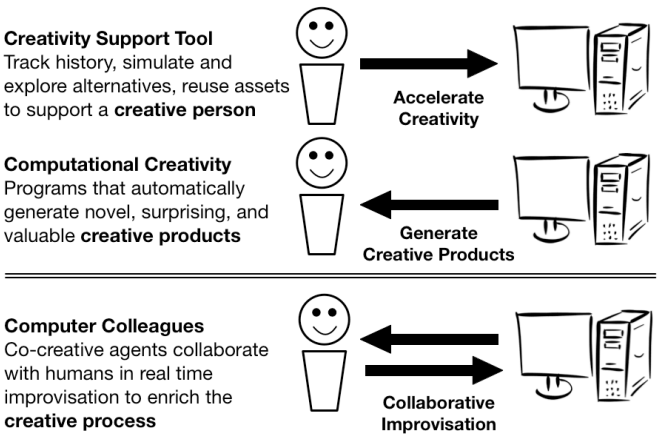


Figure 1: Computer Colleagues Bridge the Gap Between CSTs and Computational Creativity

Background

Computational Creativity

HCI researchers build creativity support tools that augment and extend the creative abilities of humans (Shneiderman 2007), while AI researchers develop computationally creative systems that implement and sometimes elaborate on cognitive theories of creativity (Boden 2003; Colton 2008; Li et al. 2012). Enormous progress has been made in these two complementary pursuits; however, there is a gap in the research literature about blending humans and computers in a continuous and collaborative co-creative process (Lubart 2005). The field of computational creativity does not yet have a guiding paradigm or set of design principles to structure creative systems involving continuous real time improvisational collaboration between creative humans and creative agents (Lubart 2005).

Co-creativity is classified as multiple parties contributing to the creative process in a blended manner (Candy et al. 2002). It arises through collaboration where each contributor plays an equal role. Cooperation, on the other hand can be modeled as a distribution of labor where the result only represents the sum of each individual contribution (Candy et al. 2002). Co-creativity allows participants to improvise based on decisions of their peers. Ideas can be fused, and built upon in ways that stem from the unique mix of personalities and motivations of the team members (Candy et al. 2002). Here, the creative product emerges through interaction and negotiation between multiple parties, and the sum is greater than individual contribution. These interaction principles can be extended to include a sufficiently creative agent that can collaborate with human users in a new kind of human-computer creativity.

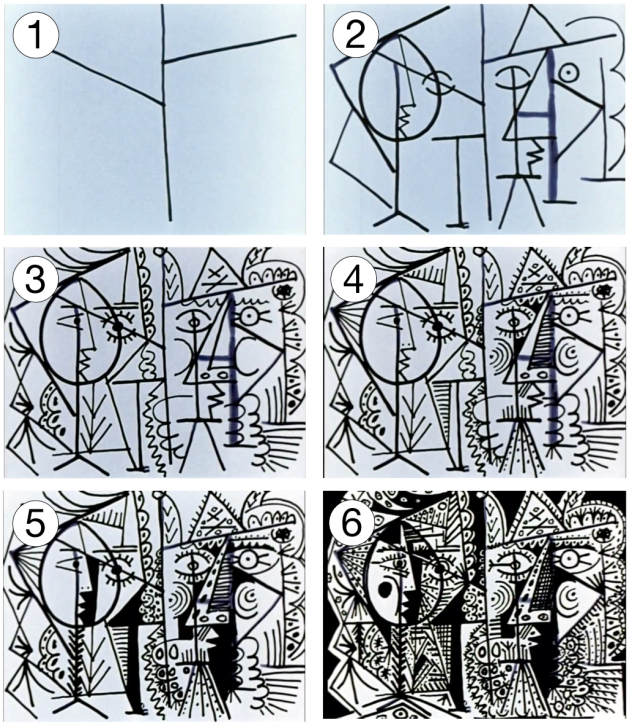


Figure 2: Time-lapse representation of Picasso's abstract art improvisation creative process reproduced from a film of Picasso painting in Clouzot (1956)

Some approaches that have yielded interesting examples of human-computer creativity include mimicry, structured improvisation, and using contextual clues to negotiate shared mental models. The improvisational percussion robot Shimon mimics human musicians by analyzing the rhythm and pitch of musical performances and generating synchronized melodic improvisations (Hoffman & Weinberg 2010). In practice, the human and robot develop a call-and-response interaction where each party modifies and builds on the previous contribution. Some co-creative agents use sensory input to construct mental models of agents, actions, intentions, and objects in the environment (Magerko et al. 2010). Mental models help agents effectively structure, organize, interpret, and act on sensory data in real time, which is critical for meaningful improvisation.

Abstract Improvisational Creativity

Pablo Picasso's work is the most well known example of the type of improvisational abstract art the system was designed for. One of the defining features of abstract art is its ability to morph and transform throughout the creative process as the artist discovers, assigns, and re-interprets meaning in the artwork (see Figure 2 and Clouzot's (1956) *Le mystère Picasso* for additional context).

In the cognitive science literature, this type of meaning re-assignment is referred to as a *conceptual shift* (Nersessian 2008). Colloquially termed the Eureka! or Aha! moment, conceptual shifts occur when two separate knowledge domains are connected in the mind (Boden 2003; Nersessian 2008). It is often partially or wholly responsible for insights that lead to creative discoveries and solutions.

Abstract art is particularly interesting for creativity research because conceptual shifts and flexible meanings are its cornerstones. Its fluidity makes abstract art ideal for collaboration, as collaborators quickly and easily negotiate common ground and construct shared meaning in an artwork. Abstract art contributions also cannot be 'wrong' in the same strict sense as representational art because accurate representations are not the goal, which helps lower the barrier of entry for novices (both human and computer).

Improvisational creativity more closely resembles a dialogue where each party makes contributions that feed into an interactive creative process (Sawyer 2012). Jazz improvisation exemplifies artists working together to experimentally negotiate creative strategies based on current musical themes, patterns, and the history of interaction (Sawyer 2012; Mendonça 2004).

Improvisational creativity is distinguished from other types of creativity because the product is usually ephemeral—*the process is the product* (Sawyer 2012). Computer colleagues can enrich the *creative process* by engaging artists in a fun and interesting collaborative art making experience. The final creative product could be thought of as merely a record of that collaborative experience.

Enactive Cognition

Enactive cognition is an outgrowth of the embodiment paradigm in cognitive science. Embodiment claims cognition is largely structured by the manner in which our bodies enable us to interact with the environment (Varela et al 1991). This approach is contrasted with earlier cognitive theories that conceptualized the mind as a machine and cognition as a complex but disembodied manipulation of symbolic representations (Newell 1959). In particular, enaction emphasizes the role that perception plays in guiding and facilitating emergent action (De Jaegher 2009). In the following sections, we describe how the enactive approach reframes perception into an active and dynamic process critical for participatory sense-making, i.e. negotiating emergent actions and meaning in concert with the environment and other agents. Next, we examine the role of goals and planning in the enactive perspective. Finally, we review some sketching and design research to show evidence that enaction plays a key role in the creative process when creative individuals 'think by doing.'

Enactive Perception In the enactive view, cognition is seen as a cycle of anticipation, assimilation and adaptation, all of which are embedded in and contributing to a continuous process of perception and action. Perception is not a passive reception of sensory data, but rather an active process of visually reaching out into the environment to understand how objects can be manipulated (Gibson 1986; Noë 2004). This type of enactive perception minimally involves a negotiation among the following factors: 1) The subject's intentional state; 2) The skills and bodily capabilities of the individual; and 3) Perceptually available features of the environment that afford different actions such as size, shape, and weight (e.g. is it graspable, liftable, draggable, etc. as elaborated in Norman (1999)). Sensory data enters the cognitive system and irrelevant data is suppressed and filtered (Gaspar 2014). Objects and details of the environment that relate to the subject's intentional goals appear to conscious perception as affordances, which can grab, direct, and guide attention and action (Norman 1999).

Each time the individual physically moves through the environment, or acts upon the environment, that action changes the perceptually available features of the environment, which can reveal new relationships and opportunities for interaction. For example, when a painter steps back from her painting, two things happen: (1) she disengages from her current painting activity, and (2) she changes the sensory input to her visual system. From this new perspective, the artist can evaluate global relationships between local regions in the painting and discover new themes and artistic goals that can guide her next artistic decisions once she re-engages the artwork.

Participatory Sense-Making The enactive view accentuates the participatory nature of meaning generation, often called participatory sense making. Cognitive systems generate meaning by active transformational and not mere informational interactions with the environment (Varela et al. 1991; Gapenne and Di Paolo 2010). Each interaction with the environment can (and often does) reveal new goals, which leads to a circuitous rather than a linear creative process. Creative individuals engage in a dialogue with the materials in their environment (and other agents) to define and refine creative intentions (Schon 1992). This view is helpful in open-ended domains where goals are often discovered rather than explicitly defined.

In human daily interactions, for example, there is evidence that some form of natural coordination takes place in the shape of movement anticipation and synchronization. A good example of participatory sense-making would be the familiar situation where you encounter someone coming from the opposite direction in a narrow passageway (De Jaegher 2009). While trying to negotiate a safe and quick passage, both participants look toward their intended path (providing a social cue) while also trying to assess the projected path of other agents. Interaction then, in the form of coordination of movements, is the decisive factor in how quickly the individuals achieve their goal of passing each other.

Rather than adopting a plan with a fixed and concrete goal state to control locomotion, an enactive analysis would posit that individuals remain flexible throughout the situated action by dynamically accommodating the choice of the other agent. If the interaction cannot be settled by subtle perceptual negotiation, more intentional gestures can be recruited to communicate intention more precisely. If collision seems unavoidable, even after clear gestures to communicate intention are made, language may be recruited to settle the navigational issue with a solid plan, usually followed by a brief period of uncomfortable laughter (because we usually manage these situations without such extreme measures).

Goals as Socially Negotiated, Dynamic, and Emergent Even at the level of social interaction with an intelligent agent, an enactive approach tries to avoid postulating high-level cognitive mechanisms at the core of our intersubjetive skills. Enaction breaks away from traditional cognitive science theories positing precisely formulated goals, detailed planning procedures, and robust internal representations of both (Newell 1959). The co-evolution of a communicative/creative process is seen here as a gradual unfolding in real time of a dynamic system spanning a human subject, the environment, and agents within it. In this view, intentions emerge but are also transformed in and through the interaction with other agents and the environment.

One argument against a naïve planning approach in AI is that it takes a significant amount of cognitive effort to construct mental simulations that provide the level of detail and granularity required to carefully plan every complex action humans engage in (De Jaegher 2009). There is considerable evidence that demonstrates humans do, in fact, have a keen skill for visual thinking, but it still takes cognitive resources to perform mental operations and inferences on images (Kosslyn 1980). It is often simply easier to act on the environment and experiment with how different interactions affect the system (Noë 2004).

Thinking By Doing The literature on creativity supports the enactive perspective with research on 'thinking by doing.' There is a multitude of evidence demonstrating how both representational and non-representational artists plan their artworks using sketches, studies, and other ways to simulate artistic alternatives (Mace 2002). Sketching reduces cognitive load and facilitates perceptually based reasoning (Schön 1992). Artists generate vague ideas and then use some form of sketch or prototyping activity to creatively explore, evaluate, and refine artistic intentions (Davis 2011). Sketching allows creative individuals to think by doing. When an action or idea is materialized in some way, the perceptual system is rewarded with richer data than pure mental simulations and abstract reasoning. Additionally, cognitive resources that would have been used to simulate the action (i.e. consciously visualizing the situation) are now freed for other tasks such as interpretation and analysis (Shneiderman 2007).

Enactive Model of Creativity

An enactive model of creativity proposes creativity as an emergent negotiation between agents with intelligent perceptual systems, exploratory interaction, and an environment rich with affordances. We first explain the visual conventions of the enactive model of creativity and describe how it can be applied to model creative cognition through time. Then, we introduce a new concept derived from our model called perceptual logic, which is a perceptual filter that highlights relevant affordances in the environment while suppressing irrelevant affordances.

Model Description

In the enactive model of creativity (see Figure 3), the awareness of the agent is represented by the vertical rectangle situated on a spectrum of cognition, which means that the agent is 'aware' of what is perceived and its current intention. Perception is constituted partly by the mental model the agent has constructed for the current situation (top-down cognition) as well as the sensory input coming from the environment (bottom-up cognition) (Gibson 1988; Glenberg 1997; Varela et al. 1999; Stewart et al 2010; Gabora 2010).

To get a sense of the intended dynamism of this model, imagine the entire 'awareness' rectangle (the central part of Figure 3) can shift to the left or right of the cognitive continuum as a function of the agent's concentration. Routine actions only require minimal thought and a limited amount of highly relevant sensory data. The enactive (and temporally extended) model of routine actions, such as driving, would by visually depicted by having the awareness rectangle resting at equilibrium in the center of the spectrum with small deviations to the left to update and revise strategy, and deviations to the right to interactively evaluate those ideas in a perceive-act cycle (see Figure 4).

If the agent is performing an unfamiliar task, however, cognitive resources are recruited to actively build a mental model of the situation, which requires performing experimental interactions, closely examining the results in the

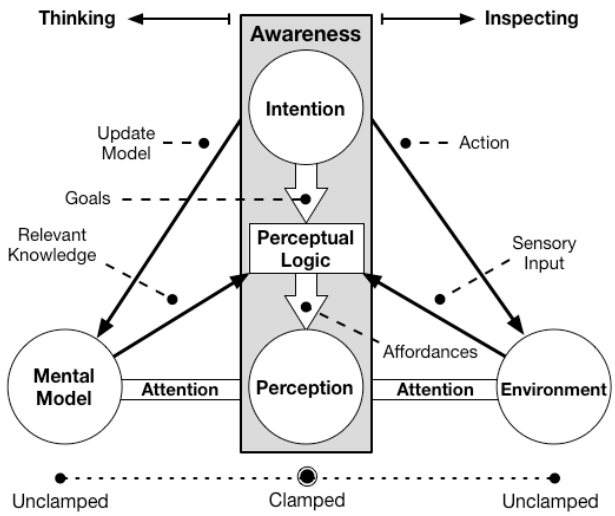


Figure 3: Enactive Model of Creativity

environment, and then updating the mental model in a slower perceive-think-act cycle. As novices learn to filter irrelevant sensory details and operate effectively with minimum conscious supervision of a task, the perceive-think-act cycle gradually tightens until expertise is achieved. Additionally, the agent can engage in pure reflection or pure interactive inspection, which would be described by tight cycles on either end of the spectrum (see Figure 4).

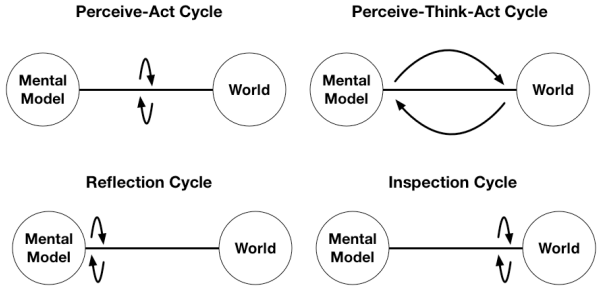


Figure 4: Cycles of cognition in the enactive model of creativity

To simulate working memory, the agent only has a limited amount of cognitive resources. These resources are used through a process of directed attention, i.e. concentration. During this simulated form of concentration, agents devote their attention to reflecting on the situation (building more detailed mental models, running complex mental simulations, etc.) and acting in a deliberate and interactive manner to inspect the world.

Perceptual Logic

The contents of perception vary based on an individual's position on this continuum of cognition (Glenberg 1997). As individuals deviate from the equilibrium in the center of the spectrum, perception becomes partially 'unclamped,' which loosens semantic constraints on sensory input and memory (Glenberg 1997). In our model, different points on the cognitive spectrum result in a unique *perceptual logic*

that is used to intelligently perceive affordances in the environment. The enactive approach in cognitive science describes the 'intelligence' of perception in a theoretical sense, but operationalizing the theory required explaining the implicit black box mechanism that makes perception 'intelligent.' The mechanism basically serves to to filter all possible affordances and present only relevant affordances to conscious perception. Perceptual logic is our proposed method for developing 'intelligent' perception in an agent.

The enactive approach proposes that perceptual intelligence arises through the formation of percept-action pairings that are chunked and internalized for quick retrieval (Noë 2004). Perceptual logic is a proposed cognitive mechanism that filters sensory data, identifies relevant percept-action pairings, and presents these percept-action pairings as affordances to perception. Perceptual logic performs a similar role as the 'simulator' in Perceptual Symbol Systems (Barsalou 1999). The simulator activates all the associated information related to a percept, including the various ways it can be interacted with based on experiential knowledge and physical characteristics.

Clamping Perception Research indicates that perception filters irrelevant sensory input to reduce distractions and facilitate everyday cognition (Gasper 2014). When the agent is engaged in a routine task and following well established affordances, sensory data is 'clamped' to filter out unnecessary details and un-conventional ways of seeing objects (Glenberg 1997). Everyday cognition is represented in EMC by situating the awareness rectangle in the center of the spectrum of cognition, creating a point of equilibrium. Shifting either to the left or right on this spectrum requires the agent to concentrate on either the details of her mental model or closely inspect details in the environment. At equilibrium, EMC proposes that perception is clamped to a combination of sensory input and cognitive input that optimizes routine interactions. When minor problems arise, such as small improvisational adjustments to the action based on environmental feedback, this equilibrium is slightly perturbed. The agent could generate various alternative actions by thinking (moving slightly left on the spectrum) and explore various ideas by interacting with the environment (moving slightly right on the spectrum).

Unclamping Perception If there is a severe disruption to the current task (e.g. a great new idea, distraction, or some kind of failure), it might become necessary to disengage from the current task to re-evaluate the situation (Dourish 2004). When an individual 'disengages' from a task, perception becomes 'unclamped' and attention shifts to thinking and simulating solutions (moving far left on spectrum) and closely examining the detail of the environment to discover new affordances (moving far right on the spectrum). The degree of concentration devoted to thinking about or acting on the environment determines how far, in either direction, awareness is situated on the spectrum of cognition. At the extreme left of the continuum (thinking) would be closing one's eyes to try to think deeply about a topic, which removes sensory input from perception altogether. At the extreme right of the continuum (inspecting) would be an individual fully concentrated on acting skillfully, carefully, and deliberately on the environment.

Modulating Semantic Constraints During these periods of disengaged evaluation, EMC proposes that the semantic constraints for recalling associated ideas from memory and interpreting elements in the environment become 'unclamped' to enable re-conceptualization. Unclamping semantic constraints helps overcome functional fixedness, which is a phenomenon where individuals have trouble dissociating objects from their entrenched meaning during insight problem solving (Adamson 1952).

Interestingly, this model identifies an important role for distraction in the creative process. Distraction is one way to prompt an individual to disengage from everyday cognition. In abstract art, for example, unfinished segments of the artwork (or unexpected contributions from a collaborator) may distract the artist while they are drawing. These newly discovered areas might not align with the artist's current intention. As a result, the artist might want to resolve that tension by drawing additional lines, which can catalyze the creative process. However, too many distractions might frustrate an artist.

EMC accounts for meaning negotiation by describing how perception employs different types of perceptual logic to filter affordances in the environment. Applying a different perceptual logic changes the manner in which sensory inputs are processed, organized, and made sense of. It therefore reveals different affordances in the environment, which can help the individuals discover new creative uses for objects that are relevant to goals.

Drawing Apprentice System Design

The enactive model of creativity informs the Drawing Apprentice's cognitive architecture, and collaborative drawing and jazz improv informs the turn-taking strategies (Mendonça & Wallace 2004; Pressing 1984). Figure 5 shows the

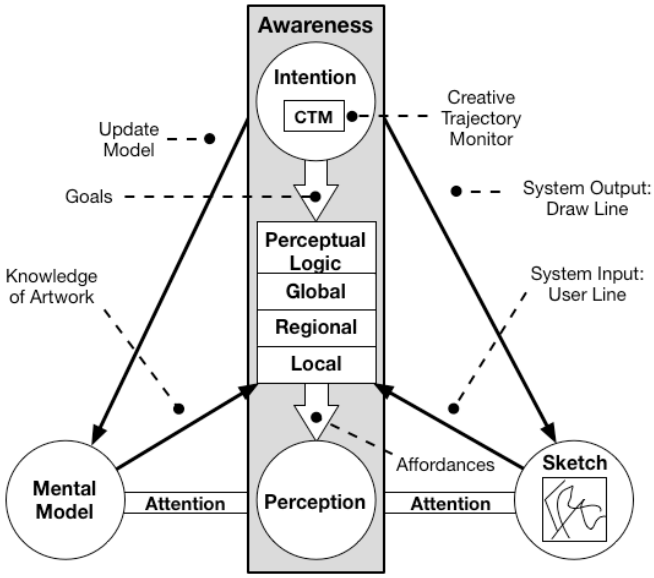


Figure 5: Apprentice Software Architecture

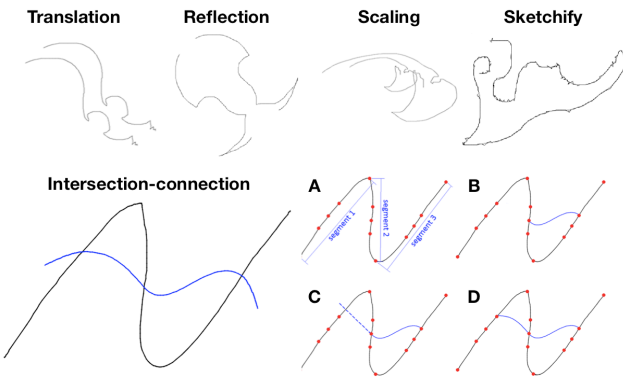


Figure 6: Local (top row) and regional (bottom row) perceptual logic drawing algorithms in system prototype.

system architecture of Apprentice. The creative dialogue begins as the human inputs a line. All current lines from the canvas are sent to the perceptual logic module. The perceptual logic module consults the creative trajectory monitor to determine what perceptual logic to apply to its current data set. The planned creative trajectory monitor has a coarse grained record of the previous drawing behavior based on the time between the user's lines (i.e. longer periods of rest represent reflection, which is categorized as global perceptual logic, and short and rapid detail strokes are categorized as local perceptual logic). The creative trajectory monitor then averages the last 10-15 seconds of user drawing behavior and selects the dominant perceptual logic of the user. The average creative trajectory is adopted by the system to determine what layer of perceptual logic to apply in the current interaction.

Layers of Perceptual Logic

EMC suggests that each layer of perceptual logic should generate unique artistic affordances from the same input, such as shading a circle, intersecting it, and replicating it. Each logic layer sends its algorithms different amount of lines and different features for discriminating lines. There are several critical points that each perceptual logic filter can use in different ways, such as inflection points, start point, end point, segments between inflections, and corners. Moreover, gestalt groupings (e.g. proximity, similarity, closure, etc.) provide additional features to generate unique affordances building relationships between lines, groups of lines, regions, and patterns (Arnheim 2001).

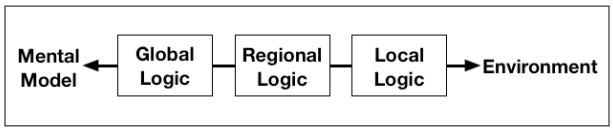


Figure 7: Layers of perceptual logic. Local perceptual logic mimics the *last input line* without any model of the artwork. Regional perceptual logic analyzes *recent input lines into* gestalt groupings to build on regional relationships. Global perceptual logic analyzes *all lines* in the agent's mental model of the artwork to evaluate overall composition and identify opportunities.

There are three layers or types of perceptual logic in EMC (local, regional, and global) determined by the position of awareness on the spectrum of cognition (see Figure 7 for an explanation of the categories of perceptual logic). We are implementing the EMC in steps with one layer of perceptual logic implemented per step. Each successive layer of perceptual logic considers a larger portion of the drawing at a higher level of conceptual abstraction (global being the most complex), which presents additional technical hurdles. Layering our implementation strategy allows a practice-based approach that encourages iterative testing with artists to ensure a meaningful artistic tool.

With only the first two layers (partly) implemented, the system can receive line input from the user, analyze it and generate an improvised response line based on the visual features of the input line and surrounding region. Table 1 and Figure 6 display the first five types of drawing algorithms we implemented in the prototype.

Local perceptual logic considers the visual features of one line. These drawing algorithms perform simple mathematical transformations on the input line and then redraw it, such as translation, reflection, scaling, and sketchify (see Figure 6). Local perceptual logic essentially mimics the creative input of the user by repeating the user's action with a small variation.

Regional perceptual logic, on the other hand, segments recent line inputs into line groups, regions, and containers based on principles of gestalt grouping, such as proximity, similarity, common fate, and continuity (Arnheim 2001). The system then generates a line that builds relationships between objects in the same region or container. Intersection-connection is the first regional algorithm that analyzes an input line into critical regions to respond to the actual shape of the line (shown in Figure 6).

Global perceptual logic (not yet implemented) considers the artwork as a whole. These algorithms are more 'intelligent' than regional and local perceptual logic algorithms because they consider how the different regions of the drawing balance to form an overall composition. When this perceptual logic is applied, the system may decide to completely decouple its contribution from the human's recent input, i.e. it can select non-active regions of the artwork on which to operate if it presents more rewarding artistic opportunities. Global perceptual logic is the highest level of cognitive functioning and will eventually include semantic knowledge such as how to draw a dog, cat, person, etc.

System Evaluation

While creativity support tools typically help users produce a more polished product in less time, computer colleagues aim to support the *creative process* by increasing playful exploration, motivation, and creative engagement. Evaluating computer colleagues therefore involves analyzing and measuring creative engagement in the co-creative process rather than judging the creativity of the final product.

Figure 8 shows an early practice-based art study of an expert artist (the first author) collaborating with the Draw-

ing Apprentice over a period of 2 hours. Drawings 1 & 2 demonstrate short turn taking collaboration between the artist and the Drawing Apprentice (computer lines are blue). Without the regional and global perceptual logic layers, the system only has minimal knowledge of the artwork. It knows what each of the line inputs are, but nothing about their relationship or the overall composition of the artwork. In future work, the regional perceptual logic layer will group line inputs into regions and containers to enable the system to learn and modify entire shapes (rather than individual lines). However, even without regional perceptual logic, the system was able to achieve complex (and artistically valuable) outputs in drawings 3-6 because the human starts defining themes and creating complex artistic patterns by drawing many lines per turn in rapid succession. The basic mimic functions of the Drawing Apprentice leveraged this complexity to achieve equally detailed output. The final product is shown in all black (as the artist saw it) in drawing 9.

To capitalize on the emergent nature of creativity in improvisation, our development efforts focus on building more sophisticated methods of perceiving, analyzing, and understanding drawn human input in such a way that it can be intelligently and creatively re-used by the system. This involves teaching the system how to recognize line groups (regional perceptual logic), how to define relationships between those line groups (global perceptual logic), and when it is appropriate to use them for generating artistic contributions (creative trajectory monitory).

In practice, the current prototype appears like a clumsy novice because it can achieve continuous improvisation, but it cannot detect patterns, make abstractions about the artwork, or understand any user intentionality. This limitation means that many of the system's contributions accidently disrupt things the user intentionally drew, such as a face or a nice curve. This creates tension for the artist and can serve as a creative catalyst or as a source of frustration if the disruptions are too severe or frequent. Skilled artistic collaborators are typically quite flexible and can integrate a wide variety of unexpected line contributions into their drawings with one key exception: completing the drawing.

When the artist was ready to complete the drawing by perfecting and refining each major line (drawings 7 & 8), the system kept blindly mimicking each line input, which effectively produced more work for the artist because each computer contribution was an unpolished line that required refining. This process eventually became frustrating because the artist wanted to stop but was never satisfied with the precision of the lines. Without global perceptual logic, the drawing as a whole cannot be evaluated to determines its level of completion.

With only the local and part of regional perceptual logic implemented, the Drawing Apprentice is able to maintain *continuous* collaboration with an expert artist, which is a milestone for the project. In addition to continuous collaboration, the final prototype will be successful if: (1) It provides similar benefits as a human collaborator (i.e. playful exploration, motivation, and creative engagement) (Carroll 2009); (2) Users find collaboration meaningful and valuable (Candy and Edmonds 2002); and (3) Implementing additional parts of the EMC increases creative engagement (Candy and Edmonds 2002).

Our research agenda includes a user study to evaluate the system. The study is a controlled experiment that compares collaborating with the Drawing Apprentice to human collaboration and a random control. Participants are asked to perform three collaborative drawing sessions on a tablet computer with an unknown 'player' as the computer collaborator (e.g. Apprentice, human, or random lines). After each drawing session, the participant will be interviewed and complete the Creativity Support Index to measure playful exploration, motivation, and creative engagement (Carroll et al. 2009).

Conclusions

This paper described a cognitive model of enactive creativity that is useful for designing continuous improvisational collaboration in creative systems. We built an artistic computer colleague called the Drawing Apprentice to test our enactive model of creativity (EMC). The Drawing Apprentice embodies the principles of EMC using increasingly complex layers of perceptual logic to analyze and react to user input in real time improvisation. We hypothesized collaboration with computer colleagues based on the enactive model of creativity can enrich the creative process like human collaboration (i.e. increase playful exploration, motivation, creative engagement) in open-ended creative domain such as non-representational visual art. We presented the theory, design, prototype details, and early collaborative artwork generated with Drawing Apprentice, the co-creative drawing partner.

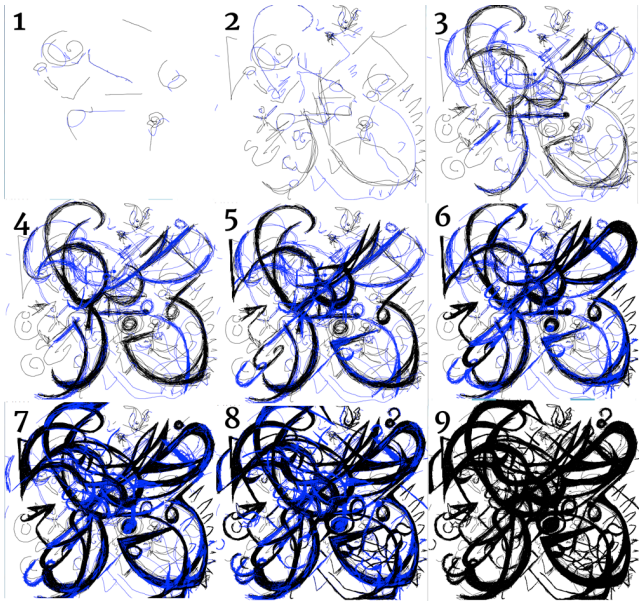


Figure 8: Time-lapse image of expert artist collaborating with the Drawing Apprentice (computer lines are blue).

Acknowledgements

Thanks to Dr. Ellen Yi-Luen Do and researchers Cora Wilson and Monet Tomioka for helping build the Drawing Apprentice.

References

Arnheim, R. (2001). *Art and visual perception*. Stockholms Universitet.

Adamson, R. E. (1952). Functional fixedness as related to problem solving: A repetition of three experiments. *Journal of experimental psychology*, *44*(4), 288.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and brain sciences*, *22*(04), 577-660.

Boden, M. A. (2003). *The creative mind: Myths and mechanisms*. Routledge.

Candy, L., & Edmonds, E. (2002, October). Modeling co-creativity in art and technology. In *Proceedings of the 4th conference on Creativity & cognition*, 134-141. ACM.

Carroll, E. A., Latulipe, C., Fung, R., & Terry, M. (2009, October). Creativity factor evaluation: towards a standardized survey metric for creativity support. In *Proceedings of the seventh ACM conference on Creativity and cognition* (pp. 127-136). ACM.

Clouzot, H., (1956). *Le mystère Picasso*. France: Filsonor.

Colton, S. (2008). Creativity Versus the Perception of Creativity in Computational Systems. In *AAAI Spring Symposium: Creative Intelligent Systems* (pp. 14-20).

Davis, N., Li, B., O'Neill, B., Riedl, M., & Nitsche, M. (2011, November). Distributed creative cognition in digital filmmaking. In *Proceedings of the 8th ACM Conference on Creativity and Cognition*, 207-216. ACM.

De Jaegher, Hanne. 2009. "Social Understanding through Direct Perception? Yes, by Interacting' *Consciousness and Cognition* 18: 535-542.

Dourish, P. (2004). *Where the action is: the foundations of embodied interaction*. MIT press.

Gabora, L. (2010). Revenge of the "neurds": Characterizing creative thought in terms of the structure and dynamics of memory. *Creativity Research Journal*,*22*(1), 1-13.

Gaspar, J. M., & McDonald, J. J. (2014). Suppression of Salient Objects Prevents Distraction in Visual Search. *The Journal of Neuroscience*, *34*(16), 5658-5666.

Gibson, J. J. 1986. *The ecological approach to visual perception*. Routledge.

Glenberg, A. M. (1997). What memory is for: Creating meaning in the service of action. *Behavioral and brain sciences*, *20*(01), 41-50.

Hoffman, G., & Weinberg, G. 2010, May. Gesture-based human-robot jazz improvisation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 582-587. IEEE.

Kosslyn, S. M. 1996. Image and brain: The resolution of the imagery debate. The MIT Press.

Laird, J. E., Rosenbloom, P. S., & Newell, A. (1986). Chunking in Soar: The anatomy of a general learning mechanism. *Machine learning*, *1*(1), 11-46.

Li, B., Zook, A., Davis, N., & Riedl, M. O. (2012). Goal-Driven Conceptual Blending: A Computational Approach for Creativity. In *International Conference on Computational Creativity* (Vol. 10).

Lubart, T. 2005. How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies*, *63*(4), 365-369.

Magerko, B., Fiesler, C., Baumer, A., & Fuller, D. 2010, June. Bottoms up: improvisational micro-agents. In *Proceedings of the Intelligent Narrative Technologies III Workshop*, 8. ACM.

Mace, M. A., & Ward, T. 2002. Modeling the creative process: A grounded theory analysis of creativity in the domain of art making. *Creativity Research Journal*, *14*(2), 179-192.

Mendonça, D., & Wallace, W. A. 2004. Cognition in jazz improvisation: An exploratory study. In *26th Annual Meeting of the Cognitive Science Society, Chicago, IL*.

Nersessian, N. 2008. *Creating scientific concepts*. The MIT Press.

Newell, A. (1959). *The processes of creative thinking*. Rand Corporation.

Noë, A. (2004). *Action in perception*. MIT press.

Norman, D. A. (1999). Affordance, conventions, and design. *interactions*, *6*(3), 38-43.

Pressing, J. (1984). Cognitive processes in improvisation. *Advances in Psychology*, *19*, 345-363.

Stewart, J., O. Gapenne and E. Di Paolo. 2010. *Enaction: Toward a New paradigm for Cognitive Science*. Cambridge, Mass: The MIT Press.

Sawyer, R. K. 2012. Explaining creativity: The science of human innovation. Oxford University Press.

Schön, D. A. 1992. Designing as reflective conversation with the materials of a design situation. *Knowledge-Based Systems*, *5*(1), 3-14.

Shneiderman, B. (2007). Creativity support tools: accelerating discovery and innovation. *Communications of the ACM, 50*(12).

Varela, F.J, Thompson, E, and Rosch, E. 1991. *The Embodied Mind: Cognitive Science and Human Experience.* Cambridge, Mass: The MIT Press.

Vygotsky, L. S. (1980). *Mind in society: The development of higher psychological processes*. Harvard University Press.

Computational Game Creativity

Antonios Liapis[1], **Georgios N. Yannakakis**[1,2] **and Julian Togelius**[1]
1: Center for Computer Games Research, IT University of Copenhagen, Copenhagen, Denmark
2: Institute of Digital Games, University of Malta, Msida, Malta
anli@itu.dk, georgios.yannakakis@um.edu.mt, juto@itu.dk

Abstract

Computational creativity has traditionally relied on well-controlled, single-faceted and established domains such as visual art, narrative and audio. On the other hand, research on autonomous generation methods for game artifacts has not yet considered the creative capacity of those methods. In this paper we position computer games as the ideal application domain for computational creativity for the unique features they offer: being highly interactive, dynamic and content-intensive software applications. Their multifaceted nature is key in our argumentation as the successful orchestration of different art domains (such as visual art, audio and level architecture) with game mechanics design is a grand challenge for the study of computational creativity in this multidisciplinary domain. Computer games not only challenge computational creativity and provide a creative sandbox for advancing the field but they also offer an opportunity for computational creativity methods to be extensively assessed (via a huge population of gamers) through commercial-standard products of high impact and financial value.

Games: the Killer App for Computational Creativity

More than a decade of research in computational creativity (CC) has explored the study of autonomous generative systems in a plethora of domains including non-photorealistic art (Colton 2012), music (Wiggins et al. 1999), jokes (Binsted and Ritchie 1997), and stories (Peinado and Gervás 2006) as well as mathematics (Colton 2002) and engineering (Gemeinboeck and Saunders 2013). While commercial games have used computer generated artifacts such as levels and visuals since the early 1980s, academic research in more ambitious and rigorous autonomous game artifact generation methods, e.g. search-based procedural content generation (Togelius et al. 2011), is only very recent. Despite notable exceptions (Cook, Colton, and Gow 2013; Zook, Riedl, and Magerko 2011; Smith and Mateas 2011), the creation of games and their content has not yet systematically been explored as a computationally creative process. From a CC perspective, procedural content generation (PCG) in games has been viewed — like mathematics and engineering — as a potentially creative activity but only if done exceptionally well. The intersection of CC, game design and advanced game technology (e.g. PCG) opens up an entirely new field for studying CC as well as a new perspective for game research. This paper argues that the creative capacity of automated game designers is expected to advance the field of computational creativity and lead to major breakthroughs as, due to their very nature, computer games challenge computational creativity methods at large.

This position paper contends that games constitute the killer application for the study of CC for a number of reasons. First, computer games are *multifaceted*: the types of creative processes met in computer games include visual art, sound design, graphic design, interaction design, narrative generation, virtual cinematography, aesthetics and environment beautification. The fusion of the numerous and highly diverse creative domains within a single software application makes games the ideal arena for the study of computational (and human) creativity. It is also important to note that each art form (or facet) met in games elicits different experiences to its users, e.g. game rules affect the player's immersion (Calleja 2011); their fusion into the final software targeting the ultimate play experience for a rather large and diverse audience is an additional challenge for CC research. Second, games are *content-intensive* processes with open boundaries for creativity as content for each creative facet comes in different representations, under different sets of constraints and often created in massive amounts. Finally, the creation (game) offers a *rich* interaction with the user (player): a game can be appreciated as an art form or for its creative capacity only when experienced through play. The play experience is highly interactive and engaging, moreso than any other form of art. Thus, autonomous computational game creators should attempt to design new games that can be both useful (playable) and deemed to be creative (or novel) considering that artifacts generated can be experienced and possibly altered. For example, the game narrative, the illumination of a room, or the placement of objects can be altered by a player in a game; this explodes in terms of complexity when the game includes user-generated content or social dynamics in multiplayer games.

Another unique property of games is that autonomous creative systems have a long history in the game industry. PCG is used, in specific roles, by many commercial games in order to create engaging but unpredictable game experiences

and to lessen the burden of manual game content creation by automating parts of it. Unlike other creative domains where computational creativity is shunned by human artists and critics (Colton 2008), the game industry not only "invented" PCG but proudly advertises its presence as a selling point. *Diablo III* (Blizzard 2012), which set a record by selling 3.5 million copies in the first 24 hours of its release, proudly states that "[previous] games established the series' hallmarks: randomized levels, the relentless onslaught of monsters and events in a perpetually fresh world, [...]"[1]. Highly-awarded *Skyrim* (Bethesda 2011) boasts of its Radiant A.I. (which allows for the "dynamic reaction to the player's actions by both NPCs and the game world") and its Radiant Story (which "records your actions and changes things in the world according to what you have done"). The prevalence of e.g. level generators in games makes both developers and end-users acceptant of the power of computational creativity. Unlike traditional art media, where CC is considered more of an academic pursuit, PCG is a commercial necessity for many games: this makes synergies between game industry and CC research desirable as evidenced by Howlett, Colton, and Browne (2010).

This paper introduces *computational game creativity* as the study of computational creativity *within* and *for* computer games. Games can be (1) improved as products via computational creations (*for*) and/or (2) used as the ultimate canvas for the study of computational creativity as a process (*within*). Computational game creativity (CGC) is positioned at the intersection of developing fields within games research and long-studied fields within computational creativity such as visual art and narrative. To position computational creativity within games we identify a number of key creative facets in modern game development and design and discuss their required orchestration for a final successful game product. The paper concludes with a discussion on the future trends of CGC and key open research questions.

Creative Facets of Games

Games are multifaceted as they have several creative domains contributing substantially to the game's look, feel, and experience. This section highlights different creative *facets* of games and points to instances of algorithmically created game content for these facets. While several frameworks and ontologies exist for describing elements of games, e.g. by Hunicke, Leblanc, and Zubek (2004), the chosen facets are a closer match to established creative domains such as music, painting or architecture.

This section primarily argues that each facet fulfills Ritche's definition of a potentially "creative" activity (Ritchie 2007, p.71). Additionally, it uses Ritchie's essential properties for creativity, i.e. *novelty*, *quality* and *typicality* (Ritchie 2007) in terms of the goals of each creation process; whether these goals (or the greater goal of creativity) are met, however, will not be evaluated in this paper.

[1]From the official 'What is Diablo 3?' page at Blizzard's website: http://us.battle.net/d3/en/game/what-is

Visuals

As digital games are uniformly displayed on a screen, any game primarily relies on visual output to convey information to the player. Game visuals can range from photorealistic, to caricaturized, to abstract (Järvinen 2002). While photorealistic visuals as those in the *FIFA* series (EA Sports 1993) are direct representations of objects, in cases where no real-world equivalent exists (such as in fantasy or sci-fi settings) artists must use real-world reference material and extrapolate them to fantastical lengths with "what if" scenarios. Caricaturized visuals often aim at eliciting a specific emotion, such as melancholy in the black and white theme of *Limbo* (Playdead 2010). Abstract visuals include the 8-bit art of early games, where constraints of the medium (low-tech monitors) forced game artists to become particularly creative in their design of memorable characters using as few pixels or colors as possible.

In terms of computer generated visual output for games, the most commercially successful examples thereof are middleware which algorithmically create 3D models of trees with *SpeedTree* (IDV 2002) or faces with *FaceGen* (Singular Inversions 2001). Since such middleware are used by multiple high-end commercial games, their algorithms are carefully finetuned to ensure that the generated artifacts imitate real-world objects, targeting typicality in their creations. Games with fewer tethers in the real world can allow a broader range of generated visual elements. Petalz (Risi et al. 2012), for instance, generates colorful flowers which are the core focus of a flower-collecting game. Galactic Arms Race (Hastings, Guha, and Stanley 2009), on the other hand, generates the colors and trajectories of weapons in a space shooter game. Both examples have a wide expressive range as they primarily target novelty, with uninteresting or unwanted visuals being pruned by the player via interactive evolution. In order to impart a sense of visual appreciation to the generator, Liapis, Yannakakis, and Togelius (2012) assigned several dimensions of visual taste inspired by cognitive research on "universal" properties of beauty (Arnheim 2004); the algorithm was able to evaluate generated spaceships based on size, simplicity, balance and symmetry and adjust the generative rules via artificial evolution. The model of visual taste could further be adapted to a human user, with visual properties prominent in chosen spaceships being targeted in the next evolutionary run. In terms of creativity, this spaceship generator targeted typicality via vertical symmetry and constraints on what constitutes a valid spaceship, as well as quality via the computational model of visual taste. Beyond generating in-game entities, Howlett, Colton, and Browne (2010) generate pixel shaders which substantially change the appearance of a game scene, pointing to a broad expressive range. The shaders' novelty is significant, while their quality is based on a user's a priori specification of target hues; however, the resulting scenes are often too bright and objects are hard to make out, pointing to a low typicality with traditional game shaders.

Audio

While often overlooked when discussing games, a game's audio is an important contributor to the overall experi-

ence; its recognition is demonstrated by two BAFTA Game Awards (music and sound) and, briefly, by a MTV Video Music Award for Best Video Game Soundtrack. Game audio usually includes background music such as the fully orchestrated soundtrack of *Skyrim* (Bethesda 2011), sound effects such as the pellet-eating sound from *Pac-Man* (Namco 1980) or the rewarding sounds of *Bejeweled* (Popcap 2001), and voice-acted dialogue which is deemed essential for large-scale commercial games and often includes Hollywood names such as Liam Neeson in *Fallout 3* (Bethesda 2008).

While the game industry is focusing on larger and more grandiose human productions for game audio, work on generated audio has seen several important developments in the last years, including the creation of the International Workshop on Musical Metacreation which has been, for 2012 and 2013, attached to the game-focused AIIDE conference. Apart from game sound effects such as those procedurally generated by *sfxr* and *bfxr* (both created by indie game developers), the generation of game audio is not much different than music generation outside of games. Collins (2009) goes as far as to consider sound effects caused by player actions or a tempo matching the game's difficulty level as procedural music which transforms the game's soundscape; this paper will not consider such a premise on the grounds that character animations similarly do not constitute a transformation of the game's visual experience. While synergies between facets such as audio and ludus will be discussed later, worth mention is the work of Brown (2012) in composing game soundtracks based on characters on display and the work of Houge (2012) in combining short musical phrases according to in-game events to create responsive background audio for a strategy game. Most, if not all, attempts to generate game audio rely on the synthesis of human-authored pieces, indicating that any creativity involved would be combinatorial. Berndt and Hartmann (2007) argue that such hybrid methods are preferable as they "leave the art creating process at the real artist, i.e., the human composer, and employ the machine beyond the humanly possible — the immediate adaptation in response to interactive events in a virtual environment". However, as research in music metacreation improves the aesthetic quality of generated results, more fundamentally creative methods for generating game audio are expected to become available in the future.

Narrative

Many successful games are applauded for their excellent narratives. Unlike traditional stories (including computationally created ones), however, the highly interactive nature of games necessitates the use of *interactive storytelling* (Crawford 2004). Due to the freedom of players to visit areas and interact with elements of the story in different orders, the creativity required of a game writer differs from that of an author or even a film director. Thus, evaluating the creativity (or simply the quality) of the game narrative depends not only on the beholder but also on the pieces of narrative experienced as well as their order and context.

Like more traditional forms of narrative generation, the design of interactive storytelling relies heavily on a large database of world knowledge — both textual and logical. Games acclaimed for their narrative, such as *Heavy Rain* (Quantic Dream 2010) and *Mass Effect* (Bioware 2007), include thousands of lines of dialogue authored by multiple game writers. While game-like interactive narratives such as Façade (Mateas and Stern 2005) or Prom Week (McCoy et al. 2013) similarly include a large number of prewritten dialogues, the computer is much more proactive and selects a fitting response of a virtual character based on the context of the current discussion, the player's assumed knowledge and the future intended outcome. Since typicality is still a concern in such projects — for instance, Façade wants the game to tell a story of a couple with marriage problems — the novelty of the story's conclusion is often not exceptional, although the events leading to this conclusion may well be.

However, the burden of imparting world knowledge to an interactive narrative system can be somewhat alleviated by directly using real world data to inform the creation process. Human-based computation can use previous user interactions, current world events or online encyclopedias in order to detect items of interest or logical connections between story elements. For instance, Orkin and Roy (2007) use a lexicon of actions and utterances from data of over 5000 players in a simple restaurant game to train virtual agents' verbal responses based on N-grams; of note is the evaluation of this machine learning method which required an audience of human judges to rate the agents' behavior in terms of typicality, i.e. whether they were likely to be heard in a restaurant. Swanson and Gordon (2012) created a co-operative storytelling system where human and computer take turns adding sentences to an emerging story; the computer analyzes the current story, matches it to a database of over a million stories from web blogs and uses the corresponding next sentences from the closest matching story. Cook, Colton, and Pease (2012) used current news items from an online news site as well as wikipedia images of their protagonists (tailored to the story's mood) in order to implicitly tell a story in the background of a platformer game.

Ludus

While games have the previous facets in common with other media, there are also those that are unique to games. The term *Ludus*, established by Caillois (1961) and elaborated by Frasca (1999), refers to an "activity organized under a system of rules that defines a victory or a defeat, a gain or a loss." The uniqueness of the *ludic facet* stems from the fact that rules define the limits of player freedom and pose as player goals; this allows room for creativity in defining the limits and goals of player interaction.

A game's play experience is primarily defined by the game's rules. Rules provide the structures and frames for play (e.g. winning and losing conditions), as well as the game's mechanics, i.e. the actions available to the player. In commercial games, rules are carefully crafted by human game designers. More often than not, such rules follow the standards of the game's genre which constrains the creativity of the designers. While often a sequel to an established series has minor rule changes from its predecessors, there have been cases where a minor tweak in the rules has caused the literal transformation of a genre. An exemplar of this is

a fan-made modification of the strategy game *Warcraft III* (Blizzard 2002) which removed base building and most unit control, allowing the user to control a single 'hero' unit; the resounding success of these tweaks has since given rise to a new, popular game genre: Multiplayer Online Battle Arenas.

Several researchers have attempted to build systems that generate game rules; however, the challenges and affordances of such creativity are naturally different than for visuals or narrative. Early systems used grammar rewriting or similar methods to tweak rules of existing games. As an example, Metagame (Pell 1992) tweaked the rules of Chess in order to create a class of games for evaluating general game-playing AI; since the motivation was to create a class of games, Metagame targets typicality with the base game. Metagame, however, ensured the quality of generated results in part due to the existence of a well-formed, successful inspiring set (Chess) and in part due to human-authored specifications for changing rules in order to maintain fairness between players etc. More recent work targets quality in the form of constraints on playability: Smith and Mateas (2010) generate game rules which satisfy the constraint that the victory condition is attainable, without however evaluating how challenging or intuitive the path to victory is. Evaluating quality in terms of challenge or learnability of the generated rules necessitates that the game is somehow played: the score (or other metrics) of a simulated playthrough can be used as an objective function for evolutionary computation (Togelius et al. 2011). As an example, Togelius and Schmidhuber (2008) evolve rules for simple Pac-Man like games, evaluating the resulting games based on their learnability in simulated playthroughs; by assuming that good games are non-trivial but learnable, the system targets an arguably more elaborate measure of quality than constraints. A successful example of game rule generation is the *Ludi* system (Browne and Maire 2010) which generates complete two-player board games in the style of classic games such as Chess and Go; generated game rules and boards are evaluated via aesthetic measurements made during self-trials.

Level Architecture

Most games are built upon the spatial navigation of levels which determine how the player agent can progress from one point in the game to another. Some examples of levels include the two-dimensional arrangement of platforms and coins in *Super Mario Bros* (Nintendo 1985), the three-dimensional arrangement of houses, trenches and enemies in the World War 2 shooter *Call of Duty* (Infinity Ward 2003), the elaborate structures that the player tears down in *Angry Birds* (Rovio 2009), or the expansive open gameworld in *Minecraft* (Mojang 2011). A game's tone is often set by its levels and the challenges they pose; digital games often have a constant or near-constant set of mechanics throughout, but vary the gameplay and challenge through level design.

Like real-world architecture, level design must take into account both visual impact and functional affordances of the artifacts it creates. Depending on the type of game, functional affordances may include a reachable end-goal for platform games such as *Super Mario Bros*, challenging gameplay for driving games such as *Forza Motorsport* (Turn 10 Studios 2005), or good action pacing with breathing room between difficult sections as in *Resident Evil 4* (Capcom 2005). On the other hand, the level's appearance plays a significant role not only for the visual stimulus it provides but also for the purposes of navigation: a sequence of identical rooms can easily make the player disoriented as was intended in the dream sequences of *Max Payne* (Remedy 2001), while dark sections can add to the challenge level of the ludic elements due to low visibility as well as psychological anxiety as is the case of *Amnesia: The Dark Descent* (Frictional Games 2010). The design of larger, open levels or gameworlds borrows less from architecture and more from city planning (Lynch 1960), with edges to constrain player freedom, districts to break the level's monotony and landmarks to orient the player and motivate exploration.

Procedural generation of levels is one of the oldest and most popular commercial applications of autonomous creative systems. The sheer volume of levels required in modern games, and the unexpectedness of a fresh, unseen level motivate game companies to rely on PCG. Examples include the generated dungeons of *Rogue* (Toy and Wichman 1980), the world in the strategy game *Civilization IV* (Firaxis 2005) or the infinite gameworld in *Minecraft*. Overall, commercial level generators' extensive use of randomness often targets novelty more than quality. Generative algorithms used in commercial games are usually "constructive", i.e. do not evaluate the levels they produce. This is especially true in games where players can interact and change the world to their liking. In *Spelunky* (Yu and Hull 2009), for instance, a player can "repair" a level where the exit can't be reached by blowing up the blocking tiles with in-game bomb items.

Academic interest in procedural level generators is recent yet extensive, focusing more on the quality of the generated levels. Quality can be ensured via a narrow set of constraints on what constitutes a desirable level, with content which satisfies it generated via constraint solvers (Smith, Whitehead, and Mateas 2011), or mathematically defining a measure of level quality/aesthetics and optimizing it via evolutionary search. The level's ludic properties are more accurately estimated via simulated playthroughs of the level using the game's mechanics; Togelius, De Nardi, and Lucas (2007), for instance, used models of driving behavior learned from human playtraces to derive a quality measure for generated racing tracks. Liapis, Yannakakis, and Togelius (2013) target both quality and novelty in generated game levels with quality being ensured via playability constraints and novelty targeted explicitly as novelty search.

Gameplay

While game design (ludus) and level design (architecture) are usually deemed creative activities in the development of the game's play experience, playing the game can also be a creative act. Players often exhibit considerable creativity in developing new strategies for playing the game.Well-constructed strategy games such as *Starcraft* (Blizzard 1998) see the player community develop new and deeper strategies over the course of years or decades. Devising such strategies often involves "thinking outside the box", such as the rush strategies in *Starcraft* which were outrageous to exist-

ing players. Some inventions even seem to go outside the spirit of the game (subversive play): as an example, players in *Quake* (id Software 1996) used rocket jumping (i.e. firing a rocket on the ground below them and thus damaging themselves) in order to propel themselves long distances and reach otherwise unreachable areas. The initial discovery of this technique among players should be considered highly creative as it is fortuitous and involves high risk due to the damage accrued by the blast; by the same account, an AI-controlled agent discovering such behavior should be considered highly creative as it breaks the constraints in terms of accessible locations in the level design and the balance of the game design. Creative gameplay would therefore seem to be an excellent domain for computational creativity.

Except for puzzle/casual games or strictly multiplayer games, most games include artificial agents acting as enemies, e.g. in *F.E.A.R.* (Monolith 2005) or companions e.g. in *Fable II* (Lionhead 2008). Modern agent controllers rarely limit themselves to arguably uncreative processes such as tree search and in several cases learn from player actions as in *Black & White* (Lionhead 2001), adapt to opponents' strategies as in *Endless Space: Disharmony* (Amplitude Studios 2013) and even revise locomotion patterns to match custom creature physiologies in *Spore* (Maxis 2008). Such agent controllers often target typicality (i.e. human-likeness) in cases where believable behavior is the goal (e.g. for the 2K BotPrize competition), while others target quality (i.e. winning) in adversarial games (e.g. for Starcraft competitions). It is not uncommon for agent controllers to be of high quality but atypical: for instance, the A* agent that won the 2009 Mario AI competition performed well while playing the game in a distinctly non-human-like manner (Togelius, Karakovskiy, and Baumgarten 2010). While novelty is not often the explicit goal of such controllers, the particularities of e.g. evolutionary algorithms to find unexpected solutions have been harnessed to test games for "sweet spots" or "exploits", where progress can be made in a game without really playing it well. In the work of Denzinger et al. (2005) on the sports game *FIFA*, evolutionary computation found a number of rather too innovative ways of playing the game. Computational gameplay can also be used to test generated game rules; Cook et al. (2012) highlight a subversive artificial agent using the (generated) teleportation mechanic to teleport directly to the exit without playing the level.

Interactions and Synergies among Facets

The previous section largely covered the different facets of creativity incorporated within games; as is usually the case, however, the whole is more than the sum of its parts. The interplay between the different facets and ultimately their fusion into what becomes the *play experience* is what makes games such a rich and challenging field for computational creativity. As an example of the interaction between facets, player actions (an element of ludus) are usually accompanied by a sound effect, such as the memorable sound of Mario jumping in *Super Mario Bros*. If an algorithm devises a new player action, it automatically constrains the sound effects that may accompany this action based on its duration or purpose. While action/sound (as a case of cause/effect) prioritizes the creation of one before the other, most interactions between facets are less one-sided: a game level is often memorable due to its visuals (such as the presence of a landmark) as much as it is due to the gameplay it affords, e.g. narrow corridors may elicit a claustrophobic feeling but may also facilitate aiming at incoming enemies. Game narratives especially rely on visuals, sound and ideally gameplay in order to be suitably experienced by the player.

Computational game creativity needs to rise to the challenge of tackling the compound generation of multiple facets. So far, many of the game creation projects focus on a single creative facet of a game artifact and do not investigate the interaction between different facets. For instance, Togelius and Schmidhuber (2008) create rules for red, green or blue pawns, but the colors are used for visual identification and are not, for instance, indicators of aggressive (e.g. red) or passive (e.g. blue) behaviors. Although Liapis, Yannakakis, and Togelius (2011a) evolve the speed and combat prowess of spaceships along with their appearance (Liapis, Yannakakis, and Togelius 2011b), the latter does not inform the former (e.g. "spiky" spaceships are not more powerful/aggressive). Holtar, Nelson, and Togelius (2013) use a soundtrack to generate ludic elements (e.g. spawning enemies when a clap sample plays), while the sound effects from player actions influence the enemy behavior in the same way as the soundtrack; however, the soundtrack or sound effects are not tailored (at least not computationally) based on the potential gameplay they can create. Game-o-matic (Treanor et al. 2012) translates user-authored entities and their interactions into game visuals and game mechanics respectively, yet the mechanics do not take into account the visuals or semantics of the game objects they are applied on. Perhaps as the most elaborate example, platformer games generated by the system of Cook, Colton, and Pease (2012) use visuals and sounds that match a news story; however, the actual gameplay (such as the allowed player actions, level geometry or pacing) do not reflect the story's theme. The cited examples are by no means failings of the current early work in this domain; however, the unique blend of narrative, user interaction, visuals and audio within games makes them an ideal, if challenging, domain for creativity to simultaneously explore multiple dimensions.

Potential links which can tie the separate facets together include the game's intended emotion or message. The intended emotional effect of a game element can connect the visuals (Whitfield and Whiltshire 1990) with music (Scherer and Zentner 2001), while the text or dialogue of the story can be adjusted to match the affective goal (Veale 2013). The ludic elements can also be informed by the emotional effect, by e.g. making enemies' abilities more powerful or by adapting their behavior to favor sneaking up behind the player in cases where the intended emotion is fear. The intended message of a game can also connect visuals, music, story and even ludus by measuring the distance of different words in WordNet (Fellbaum 1998) or by discovering associations between the intended message and e.g. color in Google N-grams (Veale and Hao 2007). Cook, Colton, and Pease (2012) have made several breakthroughs in using associations of images and sounds with the message (and

emotion) of news stories in generated games.

Discussion

The survey of the different facets of games and their interaction demonstrates that developing a game (via the different roles of graphic artist, sound designer, game designer or game writer) is perceived as a highly creative activity; by the broad definition of the term, a computer program should also be considered creative if it performed the same tasks. Not only that, but a game should be considered an artifact stemming from a "creative" activity (Ritchie 2007, p.71) as it falls into a large class (possibly including subclasses as game genres such as strategy games or shooters), with somewhat fuzzy boundaries (Karhulahti 2013), and with extensive human-based evaluations of quality[2].

On the other hand, evaluating the type and level of creativity in game content generators is not straightforward and remains a challenging open research question. A number of methods for evaluating computational creativity have been proposed, and could potentially be applied to CGC. The notions of novelty, quality and typicality have already been mentioned as aims of different generators for different facets of games; a more methodological evaluation of whether these goals are met could be performed. Many PCG research papers include user surveys where game artifacts are evaluated by human users, although the dimensions on which they are evaluated are not a one-to-one match with those in CC research. Other theoretical frameworks such as the FACE model (Colton, Charnley, and Pease 2011) could also be used to evaluate the type of content generated. For instance, the commercial game generators which are fine-tuned to create e.g. realistic trees with *SpeedTree* perform creative acts of the form $\langle E^g \rangle$, while evolutionary algorithms with indirect encodings such as genetic programming (Ashlock and McGuinness 2013) perform creative acts of the form $\langle C^g, E^g \rangle$. Special cases where the quality assessment is based on an artificial controller learning to play a generated game (Togelius and Schmidhuber 2008) perform creative acts of the form $\langle A^g, C^g, E^g \rangle$. More traditional categorizations such as those of combinatorial, exploratory and transformational creativity (Boden 1992) can also be applied to game content generators: for instance, the synthesis of game audio from sound samples would qualify as combinatorial creativity, while genetic search for optimal game content would qualify as exploratory creativity. The borders between these types of creativity are unclear, however, while transformational creativity can also be viewed as exploration as suggested by Wiggins (2006); the game asset generator of Liapis et al. (2013), for instance, blurs the edges between transformational and exploratory creativity.

Computational game creativity challenges CC theory's methods for evaluating creativity for two complementary reasons: (1) games as multifaceted entities can not be treated as visual or musical artifacts alone, and (2) games as highly interactive experiences can not be evaluated by a human audience but by active human participants (i.e. players) who introduce their own creativity into that of the system.

[2]e.g. www.metacritic.com compiles hundreds of game reviews.

Evaluating compound game creativity which treats the game as a coherent entity and not the sum of its parts is a key research question which can potentially lead to breakthroughs in creativity research. A possible solution could include the links which tie different facets together: evaluating whether the generated game elicits the intended emotion or communicates the intended message could be a measure of its success, although such an evaluation method would not cater for creativity from ambiguity and serendipity.

The interactive nature of games makes evaluating the creativity of the original designer (or computational creator) harder to disentangle from the player's creativity or even their perceived creativity. An elaborate level design can for instance be ignored because the player is too focused on surviving a difficult combat sequence. A game's narrative may not make sense when the story's elements or locales are visited in a different order than intended. More interestingly, the player's incomplete knowledge of the game — unlike an art critic who can literally see the big picture — may ascribe more causality and creativity to rather uncreative (i.e. random) events. Subversive play can also lead to a perception of creativity even when that was not expected by the (human or computational) creator: for instance, rocket jumping can be attributed to a player's creativity but also to a designer's creativity for adding the affordances for such subversive play in the game's physics. Finally, games where players are afforded significant agency, allowing them to alter the gameworld or make their own stories are even more challenging to evaluate intentional game creativity in vitro. As an example, the gameworld generative algorithms in *Minecraft* are relatively mundane, yet motivate players to fabricate their own goals. In such cases the creativity of a player meshes with that inserted explicitly into the game; it is likely necessary to include the machine/user as a unified entity when evaluating the creativity of such a game.

Apart from evaluating the creativity of existing computational creators, designing new generators of game content geared towards computational creativity is another promising research area. Especially promising for game creativity are compound generators which can iteratively focus on different facets of games. Multi-agent systems could be used to simulate a game development team, with each agent generating different types of game content such as visuals, audio or levels. Each agent's creations could be used as inspiring sets or constraints for the other agents: e.g. generated concept art (visuals) can be used to inspire level design, or a new player action can constrain the sound effects which accompany it. Similar results could potentially be accomplished with co-evolution, where multiple populations evolve genotypes of content of different facets (e.g. level design and game rules). As an early example which does not include all facets, Cook, Colton, and Gow (2012) co-evolve different elements of game levels such as the placement of blocking tiles, powerups and enemies. The aesthetic qualities targeted by each population could be domain-specific (such as harmonic quality or visual impact), could be derived from competition or collaboration with other populations, or could be automatically generated to fit a frame or unifying "theme" for the game, such as an intended message or emotion.

Conclusion

This paper introduced computational game creativity as the study of computational creativity within and for computer games, and provided several arguments as to why games as multifaceted, highly interactive art forms are ideal for computational creativity research. Elaborating on the different creative facets involved in the final play experience, the paper provided a short overview of current work in both game industry and game research on procedural content generation. The orchestration of these facets into a fully automatically generated game entity is a challenging future direction for CC research, and some early suggestions as to how it can all come together were listed. Other open questions for computational game creativity include the evaluation of game content generators using existing CC theory frameworks, the formulation of new frameworks that better account for the interactive and multifaceted nature of games, and the generation of new games encompassing more inclusive standards of appreciation.

Acknowledgements

The research was supported, in part, by the FP7 ICT project C2Learn (project no: 318480) and by the FP7 Marie Curie CIG project AutoGameDesign (project no: 630665).

References

Arnheim, R. 2004. *Art and visual perception: a psychology of the creative eye*. University of California Press, revised and expanded (2004) edition.

Ashlock, D., and McGuinness, C. 2013. Landscape automata for search based procedural content generation. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*.

Berndt, A., and Hartmann, K. 2007. Strategies for narrative and adaptive game scoring. In *Audio Mostly - 2nd Conference on Interaction with Sound*.

Binsted, K., and Ritchie, G. 1997. Computational rules for punning riddles. *Humor - International Journal of Humor Research* 10(1):25–76.

Boden, M. 1992. *The Creative Mind*. London: Abacus.

Brown, D. 2012. Mezzo: An adaptive, real-time composition program for game soundtracks. In *Proceedings of the AIIDE Workshop on Musical Metacreativity*.

Browne, C., and Maire, F. 2010. Evolutionary game design. *IEEE Transactions on Computational Intelligence and AI in Games* 2(1):1–16.

Caillois, R. 1961. *Man, Play and Games*. University of Illinois Press.

Calleja, G. 2011. *In-Game: From immersion to incorporation*. MIT Press.

Collins, K. 2009. An introduction to procedural music in video games. *Contemporary Music Review* 28(1):15–51.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational Creativity Theory: The FACE and IDEA models. In *Proceedings of the 2nd International Conference on Computational Creativity*.

Colton, S. 2002. *Automated Theory Formation in Pure Mathematics*. Springer-Verlag.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI Spring Symposium on Creative Systems*.

Colton, S. 2012. The painting fool: Stories from building an automated painter. In *Computers and Creativity*. Springer Berlin Heidelberg. 3–38.

Cook, M.; Colton, S.; Raad, A.; and Gow, J. 2012. Mechanic miner: Reflection-driven game mechanic discovery and level design. In *Proceedings of Applications of Evolutionary Computation*, volume 7835, LNCS, 284–293.

Cook, M.; Colton, S.; and Gow, J. 2012. Initial results from co-operative co-evolution for automated platformer design. In *Proceedings of Applications of Evolutionary Computation*, volume 7248, LNCS, 194–203. Springer.

Cook, M.; Colton, S.; and Gow, J. 2013. Nobody's a critic: On the evaluation of creative code generators – a case study in video game design. In *Proceedings of the 4th International Conference on Computational Creativity*.

Cook, M.; Colton, S.; and Pease, A. 2012. Aesthetic considerations for automated platformer design. In *Proceedings of the AAAI Artificial Intelligence for Interactive Digital Entertainment Conference*.

Crawford, C. 2004. *Chris Crawford on Interactive Storytelling*. New Riders.

Denzinger, J.; Loose, K.; Gates, D.; and Buchanan, J. 2005. Dealing with parameterized actions in behavior testing of commercial computer games. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG)*, 37–43.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech and Communication. MIT Press.

Frasca, G. 1999. Ludology meets narratology: Similitude and differences between (video)games and narrative. Originally published in Finnish in Parnasso 1999: 3, 36571. Online: ludology.org. Accessed 28 Apr 2014.

Gemeinboeck, P., and Saunders, R. 2013. Creative machine performance: Computational creativity and robotic art. In *Proceedings of the 4th International Conference on Computational Creativity*.

Hastings, E. J.; Guha, R. K.; and Stanley, K. O. 2009. Automatic content generation in the galactic arms race video game. *IEEE Transactions on Computational Intelligence and AI in Games* 1(4):245–263.

Holtar, N. I.; Nelson, M. J.; and Togelius, J. 2013. Audioverdrive: Exploring bidirectional communication between music and gameplay. In *Proceedings of the 2013 International Computer Music Conference*.

Houge, B. 2012. Cell-based music organization in Tom Clancy's EndWar. In *Proceedings of the AIIDE Workshop on Musical Metacreativity*.

Howlett, A.; Colton, S.; and Browne, C. 2010. Evolving pixel shaders for the prototype video game subversion. In *Proceedings of AISB'10*.

Hunicke, R.; Leblanc, M.; and Zubek, R. 2004. MDA: A formal approach to game design and game research.

Järvinen, A. 2002. Gran stylissimo: The audiovisual elements and styles in computer and video games. In *CGDC Conference*.

Karhulahti, V.-M. 2013. Puzzle is not a game! Basic structures of challenge. In *Proceedings of DiGRA 2013*.

Liapis, A.; Martínez, H. P.; Togelius, J.; and Yannakakis, G. N. 2013. Transforming exploratory creativity with DeLeNoX. In *Proceedings of the 4th International Conference on Computational Creativity*.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2011a. Neuroevolutionary constrained optimization for content creation. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*, 71–78.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2011b. Optimizing visual properties of game content through neuroevolution. In *Proceedings of the AAAI Artificial Intelligence for Interactive Digital Entertainment Conference*.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2012. Adapting models of visual aesthetics for personalized content creation. *IEEE Transactions on Computational Intelligence and AI in Games* 4(3):213–228.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Enhancements to constrained novelty search: Two-population novelty search for generating game content. In *Proceedings of Genetic and Evolutionary Computation Conference*.

Lynch, K. 1960. *The Image of the City*. MIT Press.

Mateas, M., and Stern, A. 2005. Procedural authorship: A case-study of the interactive drama façade. In *Digital Arts and Culture*.

McCoy, J.; Treanor, M.; Samuel, B.; Reed, A. A.; Wardrip-Fruin, N.; and Mateas, M. 2013. Prom week. In *Proceedings of the International Conference on the Foundations of Digital Games*.

Orkin, J., and Roy, D. 2007. The restaurant game: Learning social behavior and language from thousands of players online. *Journal of Game Development* 3(1):39–60.

Peinado, F., and Gervás, P. 2006. Evaluation of automatic generation of basic stories. *New Generation Computing* 24(3):289–302.

Pell, B. 1992. Metagame in symmetric, chess-like games. In *Heuristic Programming in Artificial Intelligence 3: The Third Computer Olympiad*. Ellis Horwood.

Risi, S.; Lehman, J.; D'Ambrosio, D. B.; Hall, R.; and Stanley, K. O. 2012. Combining search-based procedural content generation and social gaming in the petalz video game. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Scherer, K. R., and Zentner, K. R. 2001. *Emotional effects of music: Production rules*. Oxford University Press. chapter 16, 361–392.

Smith, A. M., and Mateas, M. 2010. Variations forever: Flexibly generating rulesets from a sculptable design space of mini-games. In *Proceedings of the IEEE Symposium on Computational Intelligence and Games (CIG)*.

Smith, A. M., and Mateas, M. 2011. Knowledge-level creativity in game design. In *Proceedings of the 2nd International Conference on Computational Creativity*.

Smith, G.; Whitehead, J.; and Mateas, M. 2011. Tanagra: Reactive planning and constraint solving for mixed-initiative level design. *IEEE Transactions on Computational Intelligence and AI in Games* (99).

Swanson, R., and Gordon, A. S. 2012. Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling. *ACM Transactions on Intelligent Interactive Systems* 2(3).

Togelius, J., and Schmidhuber, J. 2008. An experiment in automatic game design. In *IEEE Symposium on Computational Intelligence and Games*.

Togelius, J.; Yannakakis, G.; Stanley, K.; and Browne, C. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Transactions on Computational Intelligence and AI in Games* 3(3):172–186.

Togelius, J.; De Nardi, R.; and Lucas, S. 2007. Towards automatic personalised content creation for racing games. In *Proceedings of IEEE Symposium on Computational Intelligence and Games*, 252–259.

Togelius, J.; Karakovskiy, S.; and Baumgarten, R. 2010. The 2009 Mario AI competition. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*.

Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-o-matic: Generating videogames that represent ideas. In *Procedural Content Generation Workshop at the Foundations of Digital Games Conference*.

Veale, T., and Hao, Y. 2007. Comprehending and generating apt metaphors: A web-driven, case-based approach to figurative language. 1471–1476.

Veale, T. 2013. Once more, with feeling! Using creative affective metaphors to express information needs. In *Proceedings of the 4th International Conference on Computational Creativity*.

Whitfield, T. W., and Whiltshire, T. J. 1990. Color psychology: A critical review. *Genetic, Social, and General Psychology Monographs* 116(4):385–411.

Wiggins, G.; Papadopoulos, G.; Phon-Amnuaisuk, S.; and Tuson, A. 1999. Evolutionary methods for musical composition. *International Journal of Computing Anticipatory Systems*.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Zook, A.; Riedl, M. O.; and Magerko, B. S. 2011. Understanding human creativity for computational play. In *Proceedings of the 2nd International Conference on Computational Creativity*.

Ludus Ex Machina: Building A 3D Game Designer That Competes Alongside Humans

Michael Cook and Simon Colton
Computational Creativity Group
Goldsmiths, University of London
http://ccg.gold.ac.uk

Abstract

We describe *ANGELINA-5*, software capable of creating simple three-dimensional games autonomously. To the best of our knowledge, this is the first system which creates complete games in 3D. We summarise the history of the ANGELINA project so far, describe the architecture of the latest version, and give details of its participation in *Ludum Dare*, a game design competition. This is the first time that a piece of software has entered a videogame design contest for human designers, and represents a step forward for automated videogame design and computational creativity.

Introduction

Videogame development is a highly complex creative task incorporating the production of music, art, animation, architecture, narrative, cinematography, rules and system design, amongst others. It is not merely the sum of all these creative acts either, but the result of such acts cooperating together to achieve a creative goal. It is fair to say that videogame development is one of the most creatively diverse mediums that Computational Creativity has available to study.

The games development community has grown rapidly over the last decade. The ubiquity of the Internet and the rise of digital distribution has allowed small developers to bypass traditional publisher routes to selling a game, and the spread of simple development tools and APIs such as Unity, Twine and Flixel has made it easier for people without a background in programming to develop games. This culture of rapid development, of shared learning experiences and the general popularisation of game development has led to game-making *jams* (competitions) playing an increasingly important role in allowing game developers of all levels to interact with and learn from one another. Their simple premise – a time-limited event where entrants develop a game from scratch according to a given theme – makes them ideal for newcomers who wish to work on something small-scale and simple. These features also make them ideal platforms for testing computationally creative software.

We describe here *ANGELINA-5*, henceforth *ANGELINA*, an automated game designer that creates 3D games and interactive experiences using Unity, a modern engine for game development. We give details of the system's implementation and how it differs from earlier versions. We also report on *ANGELINA*'s participation in Ludum Dare, a game design contest which drew 2064 entries in December 2013. We discuss *ANGELINA*'s performance, and the cultural response to its involvement in the contest.

The rest of the paper is organised as follows: in *Background* we give a brief introduction to the *ANGELINA* project and discuss the choice of Unity as a new platform for the system; in *Design Process* we describe the latest version of *ANGELINA*, and the challenges associated with building a game designer that works with modern 3D game design technology; in *Game Jams* we discuss game design contests such as Ludum Dare and their role in the culture of game development; we then discuss *ANGELINA*'s entry to the contest in *ANGELINA and Ludum Dare*. In *Related Work*, we summarise other approaches to building systems capable of designing videogames; in *Future Work* we outline a road map for *ANGELINA*; we then close with *Conclusions*.

Background

ANGELINA is a cooperative coevolutionary system for automating the process of videogame design. There have been several different versions of *ANGELINA* in the past (Cook and Colton 2011) (Cook and Colton 2012) (Cook, Colton, and Pease 2012), each tackling a different kind of game design problem, often on different platforms or game engines. The latest version of the system represents a large step forward and a large shift in the platform that *ANGELINA* is built upon. The research aims of the project are concerned with automated game design and the procedural creation of content, but also target issues in Computational Creativity. Later versions of *ANGELINA* investigated questions of thematic control, context and framing of design decisions, and also whether *ANGELINA* could discover new game mechanics with minimal game knowledge (Cook et al. 2013).

ANGELINA is built as an extension to the Unity game development environment (www.unity3d.com). Unity is an extremely popular, versatile and powerful game engine that ships with a comprehensive development environment that is also highly extensible. Unity games can be deployed to web browsers, all major desktop operating systems as native applications, every modern games console and handheld device, and most smartphone operating systems including iOS, Android and Blackberry. This versatility means that distribution of *ANGELINA*'s games is extremely simple, and are

also distributable to a wide variety of people, hopefully increasing the success of future studies, as well as improving the dissemination of our results. Unity also supports both 3- and 2-dimensional game development, meaning that we can begin to investigate the automation of fully-3D game design.

Moving into the development of 3D games allows *ANGELINA* to explore a wider variety of game types, and also strengthens the image of *ANGELINA* as a game designer in terms of using contemporary technology, which is an important aspect of the project from a computational creativity perspective. It also allows us to improve on the design and structure of *ANGELINA* as a research tool: Unity's extensibility means that we can build *ANGELINA* as a series of modifications to the Unity tool itself. This means the system can have a full user interface, better visualisation and statistical analysis of the development process, and an easier platform on which to run experiments or integrate with other software. In terms of our project's focus, we also hope to use Unity's breadth as a platform to apply *ANGELINA* to design tasks on the spectrum between games and interactive artworks. Unity is used for a wide variety of projects besides traditional games, including interactive art installations such as *Canis Lupus*[1] and *Mothhead*[2]. We hope to make contributions to this spectrum also.

Game Jams

Structure A *game jam* is a co-ordinated event in which groups of people develop games in a fixed timeframe (commonly 48 hours), either alone or in groups. Some game jams are structured as contests, with judging, while others are organised for the self-improvement, to build communities of developers. Almost all game jams feature a theme which must be incorporated into the games designed for the event. These themes are used as creative aids, to focus people on a task or to make them explore unusual ideas.

Interpretation of the theme is often a crucial creative step in producing an interesting game, particularly when trying to distinguish an entry from potentially thousands of others. As an example, a game jam held in 2013 was run with the theme *Ten Seconds*. Entries to the jam included many games incorporating time limits of some kind, ten seconds in length. Here is a selection of alternative interpretations of the theme, used in games for the competition: the player controls an orphan asking for *seconds* of food; the player controls a *second*, someone who replaces someone else in a duel; the game records ten seconds of microphone input from the player, and procedurally converts it into a three-dimensional world to explore.

Role in Game Culture Game jams play a major role in the culture and community of game developers, particularly at independent and amateur level. In 2012, CompoHub[3] recorded a total of 134 game jams taking place, including *Ludum Dare*[4]. Ludum Dare is a thrice-annual event that takes place in April, August and December and has been running since 2002. Ludum Dare is split into two events which run in parallel – the *Competition Track* which is a 48-hour event in which solo developers make a game from scratch themselves, including any art and sound assets; and the *Jam Track* which is a 72-hour event in which the rules for the main competition are relaxed, allowing groups of developers to work together, and existing assets to be used. In December 2013, 2064 games were submitted.

After the submission period is over for Ludum Dare, a review period commences which lasts 22 days. During this period, anyone who submitted a game to the event in either track can enter ratings and leave comments on other submissions. On the main rating page, games are ordered based on a ratio of the number of ratings they have received versus the number of ratings they have given out, weighted so that this ratio is amplified at low numbers of ratings. This means that people who have submitted a game are encouraged to rate other games, since this is the fastest way of obtaining ratings for their own submission.

Reviews are broken down into eight categories: Fun, Overall, Audio, Mood, Innovation, Theme, Graphics, Humour. Note that Overall is a separate category, not an average of the other seven. Each category can be left unrated, or given a score between 1 and 5. Reviewers are encouraged to leave non-anonymous comments along with their reviews, but are not obliged to. At the end of the review period, the rankings are announced, including breakdowns per category, separated into the competition track and jam track.

Design Process

Predesign Phase

ANGELINA is given a word or phrase which acts as a theme for the game it is about to design. This method of starting a game design is derived from *game jams*, as described in the section *Background*. Examples of themes might be fairly straightforward, such as 'fishing', or more abstract, such as 'alone'. In some cases, the themes are intentionally unusual or restricting in order to stimulate creativity. For instance, the theme for the 2013 *Global Game Jam* was the sound of a heart beating. Developers are encouraged to incorporate the theme into their game in whichever way they can, such as through the ruleset, the narrative or the visuals.

When an input theme is given, if it is longer than a single word, *ANGELINA* will first attempt to isolate a single word most likely to be a suitable theme. Single words work better than phrases for our current methods of media acquisition and framing, because many of these processes are based on querying web services that expect singular queries. However, it should be noted that this single word approach is not a long term solution, and better theme parsing is a point of future work. In order to choose a single word from a phrase, *ANGELINA* uses a frequency analysis against a large corpus of English text[5], in order to find the least common noun. This approach was developed by analysing 150 game jam themes by hand and running similar filters on them. We

[1] http://tinyurl.com/canlupus

[2] http://tinyurl.com/mothhead

[3] http://www.compohub.net

[4] http://www.ludumdare.com/compo

[5] http://www.kilgarriff.co.uk/bnc-readme.html

found that the most prominent theming information tended to be in more specific words, particularly, nouns. 'You are the villain' simplifies to *villain*, for instance, while 'End of the Universe' simplifies to *universe*. The exception to this rule is where the theme includes meta-references to the game itself, such as 'build the level you play' – here, the important information is contained within the phrase as a whole and can't easily be condensed into a single word.

Once *ANGELINA* has a theme word, it attempts to expand the theme using word association databases[6]. We plan to replace this technique with a more relevant topic association approach in future, but for most applications word association provides a reasonable set of words relating to the source theme word. These word associations are combined with the theme word to provide a list of possible words relating to the game's overall theme. For example, the theme word *secret* would lead to a list of words including *secret*, *spy* and *mystery*. A typical list of associations runs to about thirty words. These associations are then used to perform a series of multimedia searches, one for each association, in order to build a database of assets for use in theming the final game. *ANGELINA* downloads public domain fonts from DaFont[7], 3D models from TF3DM[8] and sound effects from FreeSound[9]. These media are archived as they are downloaded, so that they can be retrieved quickly if needed in the future.

ANGELINA generates a *zone plan* which defines a number of themed zones for use within the game design. A zone is a collection of a floor texture, a wall texture, a 3D model for use as scenery, and a sound effect. The sound effect and scenery model are both randomly selected from the media downloaded from the associations list. In order to select the texture, *ANGELINA* searches through a list of 622 tagged texture files for ones which are related to one or more of the association words. A relationship can be established in one of two ways: first, it can compare the associations with the filename or folder name of the textures, which are categorised roughly according to their type (such as 'clouds' or 'paper'). Secondly, it can call on a database of word associations mined using crowdsourcing via Twitter. *ANGELINA* regularly posts random untagged texture files to its Twitter account[10] and asks its followers to provide single words which they associate with the image. These are retrieved and recorded in a database file, and used as a secondary means to relate associations to textures in the case that the filename match fails. Reply counts for a single tweet range from single replies to a dozen or more, and so far 901 responses have been recorded for 84 textures. If no matches are found through either method, *ANGELINA* selects textures randomly for the zones.

Once *ANGELINA* has selected two textures and randomly chosen a 3D model to act as scenery (we describe scenery later) and a sound effect for each zone, the zone map is complete. Before it proceeds to the main design phase, *ANGELINA* will generate a title for the game, and select a piece of music. The game's title is generated using a rhyming dictionary[11] and a corpus of popular culture references, including famous examples of media such as music and books collated from Top 1000 lists such as IMDB's Top 250 Movies[12], as well as idioms and common sayings. *ANGELINA* attempts to create puns using these resources and the list of source word associations, using a similar approach to the one described in (Cook, Colton, and Pease 2012).

To select a piece of music, *ANGELINA* attempts to choose a suitable mood for the game. It first takes the main theme word, and passes it to Metaphor Magnet[13] (Veale 2012) to obtain feelings people express in relation to the theme word. Metaphor Magnet is a tool for exploring a space of metaphors, mined from Google N-Grams. It has an array of features that are built on top of this concept, including the ability to show feelings people commonly express about a topic, such as poetic or metaphorical qualities of something, with the knowledge that these feelings are backed up by concrete examples in the N-Gram corpus.

As an illustration, if we submit the word *winter* to Metaphor Magnet, we are presented with a number of possible metaphors for winter, such as a 'frightening night' or a 'refreshing spring'. By selecting one of these, *ANGELINA* can use words which express feelings that Metaphor Magnet has corpus evidence for - e.g., winter in the context of a frightening night is commonly described as 'frightening'. This word is chosen as the base mood for the music for the game. It now has to relate this emotion to a piece of music. The music database *ANGELINA* currently uses is Incompetech[14], which categorises pieces according to twenty different moods. In order to relate the mood discovered through Metaphor Magnet with an appropriate tagged mood in Incompetech, we use DisCo[15] to rate the semantic similarity between each of the twenty known emotions and the one discovered emotion. The most similar emotion is used as the search mood for music, and a piece of music is randomly selected from the resulting pieces.

In total, *ANGELINA* uses fifteen web services or APIs during the predesign phase, from linguistic tools to databases of tagged content. In (Pease et al. 2013) the authors discuss the concept of serendipity in the context of creative software, and they note in relation to web services that "we believe this [accessing web services] will increase the likelihood of chance encounters occurring, [and] expect serendipity to follow". Note that the web services *ANGELINA* interacts with include unconstrained data sources such as Twitter as well as unedited automatically scraped databases such as Metaphor Magnet. This means that the results of the combinations of services are hard to predict, which offers a strong force of *chance*, one of the three dimensions of serendipity highlighted in (Pease et al. 2013).

[6] http://wordassociations.net
[7] http://www.dafont.com/
[8] http://tf3dm.com/
[9] http://freesound.org/
[10] twitter.com/angelinasgames
[11] http://www.wikirhymer.com
[12] http://www.imdb.com/chart/top
[13] http://ngrams.ucd.ie/metaphor-magnet-acl/
[14] http://www.incompetech.org
[15] http://www.linguatools.de/disco/disco_en.html

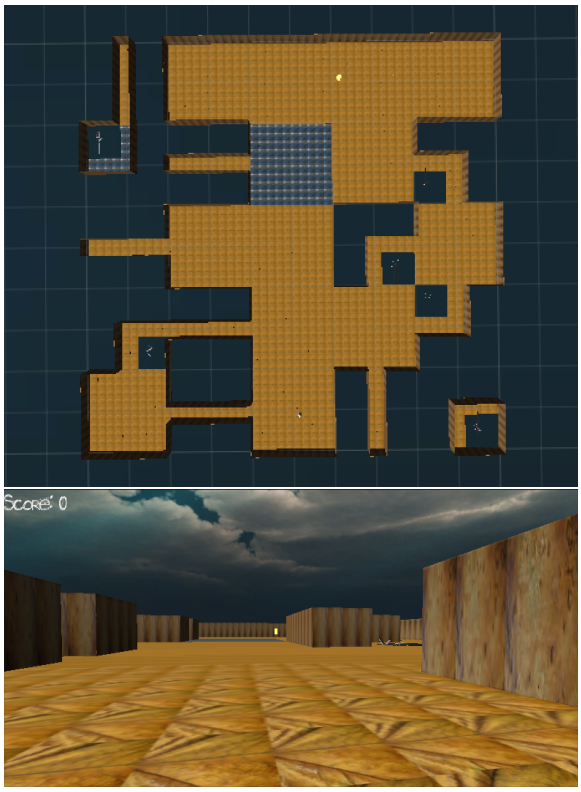


Figure 1: Screenshots of *Hit The Bulls-Spy*, a game designed by *ANGELINA*-. *Top:* The game world as viewed from above in the Unity editor. *Bottom:* A screenshot from the running game.

Design Phase

As with *ANGELINA-3* described in (Cook, Colton, and Pease 2012), *ANGELINA* is composed of several evolutionary systems that work in tandem to cooperatively evolve a game design. Each evolutionary system has two aspects to its fitness function: internal, objective rules that are considered to be unchanging regardless of the overall game design, and external, subjective rules that take into account what properties the current most fit game design has to adjust its fitness evaluation accordingly. In order to evaluate these subjective rules for a given member of a population, *ANGELINA* takes the most fit example from every other evolutionary process, combines them together to form a game, and then simulates playing that game in real-time. Currently, this simulation is very basic – *ANGELINA* will attempt to guide the player object from the starting point to the level exit, if such a path exists, and records any rules which activate (as well as how often they activate) during the course of the pathfinding. This data is used in the evaluation of the game designs, as detailed below. For more details on cooperative coevolution, see (Potter and De Jong 2000). For more details on our specific use of cooperative coevolution in *ANGELINA*, including details on the applicability of cooperative coevolution to multifaceted design problems, see (Cook and Colton 2011) and (Cook and Colton 2012).

There are currently four separate evolutionary processes:

- Level Design – which forms a basic layout of solid space in the game world. The top image in Figure 1 shows a birds-eye view of a level designed by *ANGELINA*. Level designs are currently built out of smaller tiles which are selected from a library of hand-designed tiles and arranged into a variable-size array. For instance, in Figure 1, the size of the map is five tiles wide by five tiles high. A tile is a ten by ten array of integers denoting solid ground, empty space or scenery. Scenery regions are impassable to the player, and when the game is exported, they are replaced with large, static 3D models for theming purposes.
- Zoning – which describes the visual and aural qualities of different regions of the game world. Zones are defined in the predesign phase, and during evolution a *zone map* is evolved, which is an array of integers relating each tile in the Level Design to one of the premade zones.
- Placement – which describes the start position of the player, and the position of the level exit. The primary objective in all of *ANGELINA*'s games is to reach the exit. In addition, a Placement defines the number and starting position of the game's *entities*. Entities are objects which are placed in the game world and given code to execute to play a role in the game's rules. A Placement contains a list of starting positions for each type of entity – currently all games by *ANGELINA* include exactly two entity types, the purpose of which is defined by the Ruleset.
- Ruleset – which describes the set of *behaviours* possessed by each entity. In Unity, 'behaviour' is an overloaded term used to describe any piece of code which implements a particular interface. In the current version of *ANGELINA*, we have supplied a stock of behaviours which can be attached to the entities in *ANGELINA*'s games to form a basic ruleset. These behaviours include providing motion for the entity (such as random walks, or wall following) and adding mechanical rules (such as killing a player, or providing score when collected). Expanding this set with automatically generated code is a point of future work, see (Cook et al. 2013) for details.

Each of these four processes evolve their populations in isolation, according to various fitness criteria, normally expressed as parameters which can be easily varied, so as to give *ANGELINA* the ability to alter its own fitness functions in the future. Currently, all parameters have been set through experimentation to find values which produce an interesting variety of outputs in such terms as maze style variation (a mix of open spaces as well as some labyrinthine designs too) or level layouts (dense and sparse entity placement, varying approaches to extending the distance between start and exit). The fitness criteria are as follows:

- Level Designs are selected to maximise the size of the largest contiguous island, whilst simultaneously avoiding overfitting by limiting fitness to a maximum island size. This encourages level designs in which the tiles join up to form a single level space, but avoids the situation where the entire level is one open expanse by penalising levels

which are too full of solid tiles. A level design is penalised if the player or exit start position is in empty space.

- Zone Maps are selected to maximise connectedness in zones of the same type. This means that a zone map which has two Zone 1 zones separated by a Zone 2 zone scores lower than a zone map which has a single contiguous Zone 1 zone and another single Zone 2 zone. This is done to provide consistency in when and how often a zone is encountered by the player. We anticipate this will become more important as *ANGELINA* develops, as zones will define clearly themed areas such as a forest, and having these frequently broken up by other zones would be disorienting and may reduce immersion for the player.
- Placements are selected to maximise spread of entity placements across the map, but are penalised for any placements, including player or exit placements, which are not on solid ground. Placements are also selected to maximise the distance of the path from the start position to the exit position, with a penalty if no such path exists.
- Rulesets are selected to maximise the number of rules fired in a simulation of a game. *ANGELINA* records which rules fire during an execution of the game, using a simple player controller which attempts to follow a direct path to the exit. Rulesets are penalised if there is no way for the player to gain score or die, but does not guarantee both score gain and death are in the game.

It should be noted that many of these fitness criteria are in place only to complete *ANGELINA* as a game design system, particularly Rulesets and Zone Maps. We intend to replace these by giving the system the ability to create its own fitness criteria. These might therefore be considered baseline criteria for producing a complete game design.

A typical setup for *ANGELINA* consists of a population size of 30 for each of the four evolutionary species, and a run of 40 generations for the system as a whole, meaning that each species undergoes 40 generations of evolution itself. We utilise one-point crossover and single-element mutation for all four species, since representation is almost entirely array-based. Selection is elitist, and we carry forward the parents of the previous generation, something which we found useful in previous versions of *ANGELINA*, due to the volatile nature of cooperative coevolutionary systems.

Postdesign Phase

When *ANGELINA* has completed the set number of generations and completed a game design, the game export process begins. Unity games are meant to be developed inside a single project which contains all the art and audio assets for the game, the data, the levels, the code and logic. Unity has export features that compile these various components together into a single package for a chosen platform (such as iOS). However, in our case it is *ANGELINA* that is the Unity project, not any single game that it develops. This means that the asset folders contain databases of models used in the past, music that has been downloaded, metadata and information about *ANGELINA* as a system, and so on. Exporting the games as-is is therefore not possible, as Unity cannot be told to avoid exporting certain resources, and would attempt to export gigabytes of data for each small game developed.

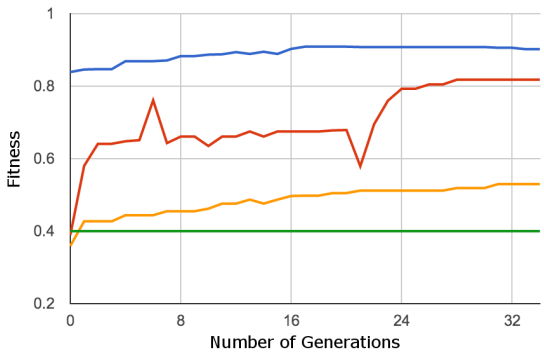


Figure 2: A graph showing the highest fitness as generations pass, for a single run of ANGELINA. The blue is Zone Map fitness; the red is Placement fitness; the yellow is Level fitness; and the green is Ruleset fitness.

For this reason, and because of a desire to archive games designed by the system, we have *ANGELINA* export all the relevant information about a game design into a separate folder. This includes a text file describing the level design and the locations of resources, as well as the asset files such as models and textures. This folder can then be read as a standalone Unity project that only imports the necessary resources, and can then export executable game binaries.

In addition to the game export, *ANGELINA* also produces a commentary describing some of the decisions it made in the production of the game, using template paragraphs which are filled in using resources it finds on the Internet, and data from the game's production. Previous versions of *ANGELINA* also used commentaries, as per (Cook, Colton, and Pease 2012). Figure 3 shows a sample commentary.

Evolutionary Performance

Figure 2 shows a sample fitness graph for each of the four evolutionary species that make up *ANGELINA*. The coloured lines are described in the caption to the figure. Note that there is little evolutionary improvement in the Zone Map or Ruleset species – these species are underdeveloped in the current version of *ANGELINA*. The system will eventually be able to track information about player routes through levels and use this to guide the placement of zones so that they affect the player's experience in a particular way, such as matching it against the emotional valence of a narrative, or to reflect changes in location. Similarly, the Ruleset species is awaiting an extension of work done on generating game mechanics through code (Cook et al. 2013) so that *ANGELINA* can propose rules itself which it can then use in a game design. Until then these evolutionary species remain incomplete. However, in the Level and Placement design species, we can see more clearly that evolution is working as intended. We anticipate that the other species will behave in this way, as they are integrated more fully into the cooperative coevolution.

This is a game about a disgruntled child. A founder. The game only has one level, and the objective is to reach the exit. Along the way, you must avoid the Tomb as they kill you, and collect the Ship.
I use some sound effects from FreeSound, like the sound of Ship. Using Google and a tool called Metaphor Magnet, I discovered that people feel charmed by Founder sometimes. So I chose a unnerving piece of music to complement the game's mood.

Figure 3: Title screen and excerpted commentary.

ANGELINA and Ludum Dare 28

The Ludum Dare 28 game jam took place on the weekend of December 13th 2013, following a week of voting which narrowed down a list of 100 themes to a shortlist of 20, and a final announcement of the winning theme at the moment the game jam started. The chosen theme was the phrase *You Only Get One*. It generated 1284 entries to the competition track, and 780 entries to the jam track.

ANGELINA entered Ludum Dare with two entries. In both cases, the system was given the theme in plain text, and configured to run for 60 generations, with a level population size of 35, a placement population size of 35, a ruleset population size of 20, and a zone population size of 15. Both games took approximately three hours to generate in their entirety, including the retrieval of game assets from the web.

The motivation behind producing two games for the jam was to investigate the presence of bias in the assessment of creative software in the medium of videogames. Our hypothesis was that, contrary to anecdotal reports and studies from Computational Creativity researchers e.g. (Pease and Colton 2011) and (Moffat and Kelly 2006), people tended to be *positively* biased towards creative software working in videogames. We submitted the first game ANGELINA produced with a commentary explaining the background of the system, and an unabridged commentary from ANGELINA about the game[16]. To anonymise the entry, the second game was submitted under a pseudonym to the game jam, without any reference to ANGELINA or the research project, and with ANGELINA's commentary edited to avoid references to software or other phrasing that might give away the game's background.[17]

[16]This game can be viewed at http://tinyurl.com/tothatsect
[17]This game can be viewed at http://tinyurl.com/stretchpoint

Entries

To That Sect *ANGELINA*'s first game, and the one which was submitted with full disclosure, was titled *To That Sect*. Figure 3 shows a screenshot from the game. The player must avoid strange demonic statues while collecting ships, on their way to reaching the exit. An unsettling piece of music plays, and a ship's bell tolls in the background. The scenery chosen for the game is a model of a player character from the game Lineage 2, dressed in armour. In both this game and *Stretch Bouquet Point* below, *ANGELINA* extracted the word 'one' from the input theme as the most likely theme word, but then found it to be too general to use as a specific theme, and so chose to use the narrowing technique we described earlier to select a word associated with 'one' as the target theme. In the case of To That Sect, it chose the word *founder*. Words associated with 'founder' included religion and sect, which accounts for the references in the game's title as well as the musical choice. Metaphor Magnet suggested that people feel charmed by founders – presumably relating to the context of a cult or a religious sect – and *ANGELINA* narrowed this emotion down to 'unnerving' using DISCO. The references to ship are due to an ambiguation of the theme word – since a ship can *founder* on rocks, as a verb.

Stretch Bouquet Point This game was submitted anonymously under a different username, without any references to software or ANGELINA in the description, and an edited commentary to hide similar references in ANGELINA's output text. The player must avoid girls referred to as 'daughters' while trying to reach the exit. An untextured model of a woman is used as scenery, and very loud chanting plays over the top of the game's music, drowning it out.

As with the previous game, 'one' is further narrowed due to it being deemed an insufficient theme. This time, 'bridesmaid' is chosen as the target word, as it was found to be associated with the word 'one'. This leads to words such as bouquet, found in the title, as well as woman and daughter. The chanting that plays over the top of the game is from the keyword 'marriage' – a recording of an African griot singing during a marriage ceremony. The connection of 'bridesmaid' to 'one' is not obvious. Many of the results from basic word association rely on words appearing in proximity to one another, and 'one' is a very generic word which may lead to erroneous or weak connections being made. Improving the association step is a point of future work.

Results

The scores for both games for each of the eight categories are listed in Table 1. Votes are not made public in Ludum Dare, and we were unable to obtain specific data from the organisers. Despite this, we can see that for many of the rating categories, the game which was publicly labelled as being created by a piece of software was ranked higher in all categories except humour – hundreds of places in some cases. For humour, we believe the sole reason the anonymised game was ranked higher was because the (unintentional) surreality of the games was perceived as funny when it was believed to be coming from a person rather than software.

	To That Sect	Stretch Bouquet Point
Overall	500	551
Fun	515	543
Audio	211	444
Graphics	441	520
Mood	180	479
Innovation	282	525
Theme	533	545
Humour	403	318

Table 1: Rankings for *ANGELINA*'s two games entered into Ludum Dare 28. There were 780 total submissions to this track. Lower rankings are better.

In order to try and maintain equal prominence for the two submissions, we rated an equal amount of Ludum Dare submissions whilst logged in as each account. To avoid both games rising to the top of the rating system at the same time and risking identification, we performed rating sessions at least 24 hours apart and at different times of the day, to minimise the risk that the same reviewer would encounter both submissions. In order to minimise the impact of our experiment on the event as a whole, we ensured that no game was rated twice, and we did not leave any written comments when rating other entries.

While the results indicate some potential positive bias towards the non-anonymised entry to Ludum Dare, we were unable to obtain specific voting data from the event organisers, leaving us unable to calculate specific confidence values for the reviews. Nevertheless, it does act as a good foundation for further investigation to be done in this area. These results are further reinforced by the written comments left underneath each submission by reviewers.

Reviews for *To That Sect* largely balanced positive with negative remarks. No comments were universally negative, tempering any criticism with positivity: "Angelina seems really good at creating an atmosphere with both sound and visuals. But the game part of it seems a bit lacking still." "The game itself is too simple. It seem the AI got the mood, but not the [game]play." By contrast, comments on *Stretch Bouquet Point* were passive-aggressive or outright critical: "this was a rather annoying experience." "You made me feel something there. Don't make me put it into words though."

The response to *To That Sect* was not without bias. One comment on the game notes that "If it [had] added shooting at the statues that you must avoid and a [target] of ships you to collect, it would have been better. It felt like playing [an] 'art-message' type of game". We can contrast this with *LITH*,[18] a game entered into the competition by a human designer, where the player navigates a maze and collect bags of gold coins, while avoiding patrolling robots. While not exactly the same, the rules of *LITH* are very close to those of *To That Sect*: search for as many objects of a certain type as possible, while avoiding another object, then exit. *LITH* was entered in the same track as *ANGELINA*'s games, and ranked 95th Overall, 125th for Fun, and 274th for Theme.

None of the comments on *LITH* reference the game's rulesets in a critical way. Contrary to the comments that *To That Sect* felt like an 'art' game, one comment actually praises LITH for feeling 'old-school', a quite opposite compliment. The games are by no means identical: *LITH*'s level is more closed in to accentuate a feeling of claustrophobia, but the similarities are many. This analysis suggests a fundamental difference in how people evaluate a game when they have knowledge and when they have no knowledge of its designer and design process. We plan further experimentation to investigate this notion.

Although the results for Ludum Dare have an extremely long tail, it is still notable that *ANGELINA*'s entry outperforms many hundreds of other entries to the contest. Low ranking entries included games which had very passive gameplay mechanics (such as a game in which single bets are placed on extremely long non-interactive races) or games which were lacking in appropriate art and audio content (many games were lacking audio entirely, or used music or sound effects which clashed with the game's theme). While these are small differences, and this was not a large, conclusive study, it is nevertheless significant that *ANGELINA* was ranked, by a community of game developers, to have outperformed many other entrants.

Related Work

Procedurally generating specific types of content for videogames is a well-explored area of research (Togelius et al. 2011). Many different types of content have been generated automatically, from rulesets (Togelius and Schmidhuber 2008) to levels (Williams-King et al. 2012) to art assets (Liapis et al. 2013) and even procedural generators themselves (Kerssemakers et al. 2012).

More specifically, the creation of software to automate the process of game design has been looked at by others in the past. In (Treanor et al. 2012) the authors describe the Game-o-Matic, a design assistant for journalists that could be given a graph representing relationships between concepts (such as *police arrests protester*) and then construct a game that reflected the network of relationships. The Game-o-Matic only understood a limited set of verb relations, and sourced its initial rulesets from a library of human-authored rules. However, it was able to source artwork for its games automatically, and could tweak rules to refine a game design, which gave it a good expressive range.

In (Nelson and Mateas 2007), the authors present a simple mini-game generation system that takes *verb-noun* constructions and presents games based on the given relationship. The input *shoot pheasant*, for example, presents games where the player controls a crosshair trying to shoot birds, or controls a bird trying to avoid being shot. Connections are made between human-tagged game mechanics and known words using a combination of ConceptNet and WordNet.

ANGELINA is not the first piece of creative software to engage with people in a social or cultural context. The Painting Fool, a piece of software its designer hopes will one day be taken seriously as an artist, has exhibited its work in public fora multiple times, e.g. (Colton and Pérez-Ferrer 2012), and has sold its artworks to collectors. Elsewhere, Ventura's PIERRE system (Morris et al. 2012) evolved soup recipes using a database of existing recipes and an understanding

[18] *LITH* game: www.tinyurl.com/lith-ludum

of food groups. PIERRE's recipes were evaluated anonymously in online cookery forums, as well as having its creations cooked by a person and evaluated via tasting on multiple occasions, with the knowledge of the recipe's origin in these latter cases. Anecdotal evidence suggested positive bias where the consumers had knowledge of PIERRE's existence, however we do not present this as serious evidence for positive bias, as the author notes that the presentation of the recipes may have contributed to the negative response to the anonymised recipe submissions.

Future Work

The work described here represents a new foundation for our research into automated game design. The flexibility of Unity as a platform, and the more general architecture of *ANGELINA*, means that we hopefully will be able to work on a single piece of software for some time, and go deeper into some of the issues we have brushed up against over the past few versions of the software. In particular, the following areas present themselves to us for further study.

- **Improved Communication** Entering *ANGELINA* in a game jam underlined the importance of the use of commentaries and context in conveying the intelligence and creativity of a system to an observer. For further exploration of the role of the observer in the context of *ANGELINA*'s entry to Ludum Dare, see (Cook and Colton 2013). In the future, ANGELINA will provide interactive commentary material that can be interrogated in-game to provide more detailed information about the design process. We believe this will ultimately increase the perception that the software is creative.
- **Innovation in Design** Because of the preliminary nature of some elements of *ANGELINA*, the game's main gameplay and objectives varied very little between different runs of the system. In order to improve this, we aim to bring in previous work on generating code for the invention of game mechanics as described in (Cook et al. 2013), and expand this to allow *ANGELINA* to generate code that produces new types of gameplay, and new styles of game. This will help strengthen the argument that ANGELINA is designing new games, and will also increase the independence of the system.
- **Better Theme Interpretation** A key aspect of entering a game jam is interpreting the given theme and working it into the final game design. We aim to integrate the theme into more aspects of the game's design than just the visual and aural theming. Good games incorporate the theme into their mechanics and design. We have discussed methods for doing this previously in (Cook and Colton 2013), and we will look to build some of them into *ANGELINA*.

Conclusions

We have described *ANGELINA*, the latest iteration of our automated game design system. *ANGELINA* is a redevelopment of the system in the Unity game engine, the first automated game designer that we know of to produce output in 3D. *ANGELINA* was developed to take a different approach to previous versions of the software, in that it would work from arbitrary phrases acting as themes. This allowed the software to take part in a game jam – the first time an automated game designer has done so, gaining a higher ranking than hundreds of other human-authored games.

We described the process of entering a game jam, as well as describing the system's two entries into the jam – one of which was publicly annotated as being developed by *ANGELINA*, while the other was anonymously submitted. We looked at the different reactions, both in terms of the scores the games received and the surrounding commentary on the games, and discussed the potential implications for creative software acting in the videogames medium in the future.

For all the mixed reactions and ratings, the response to *ANGELINA* entering a game jam was overwhelmingly positive, and the interaction with the development community will benefit us as researchers as well as the project in the long run. Hopefully we will see this trend continue, and we aim for more interaction between *ANGELINA* and the community in the future.

Acknowledgements

The authors wish to thank the reviewers for their comments which helped improve the paper, as well as Mike Kasprzak, Phil Hassey, Seth Robinson and Mike Hommel. This project has been supported by EPSRC grant EP/L00206X/1.

References

Colton, S., and Pérez-Ferrer, B. 2012. No photos harmed/growing paths from seed - an exhibition. In *Proceedings of the Non-Photorealistic Animation and Rendering Symposium*.

Colton, S.; Cook, M.; Hepworth, R.; and Pease, A. 2014. On acid drops and teardrops: Observer issues in computational creativity. In *Proceedings of the 7th AISB Symposium on Computing and Philosophy (forthcoming)*.

Cook, M., and Colton, S. 2011. Multi-faceted evolution of simple arcade games. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*.

Cook, M., and Colton, S. 2012. Initial results from co-operative co-evolution for automated platformer design. In *Proceedings of the Applications of Evolutionary Computation*.

Cook, M., and Colton, S. 2013. From mechanics to meaning and back again: Exploring techniques for the contextualisation of code. In *Proceedings of the AI & Game Aesthetics Workshop at AIIDE*.

Cook, M.; Colton, S.; Raad, A.; and Gow, J. 2013. Mechanic miner: Reflection-driven game mechanic discovery and level design. In *Proceedings of 16th European Conference on the Applications of Evolutionary Computation*.

Cook, M.; Colton, S.; and Pease, A. 2012. Aesthetic considerations for automated platformer design. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*.

Kerssemakers, M.; Tuxen, J.; Togelius, J.; and Yannakakis, G. N. 2012. A procedural procedural level generator generator. In *IEEE Conference on Computational Intelligence and Games*.

Liapis, A.; Martínez, H. P.; Togelius, J.; and Yannakakis, G. N. 2013. Transforming exploratory creativity with DeLeNoX. In *Proceedings of the Fourth International Conference on Computational Creativity*.

Moffat, D., and Kelly, M. 2006. An investigation into peoples bias against computational creativity in music composition. In *Proceedings of Third Joint Workshop on Computational Creativity*.

Morris, R. G.; Burton, S. H.; Bodily, P. M.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the Third International Conference on Computational Creativity*.

Nelson, M. J., and Mateas, M. 2007. Towards automated game design. In *Proceedings of the 10th Congress of the Italian Association for Artificial Intelligence*.

Pease, A., and Colton, S. 2011. On impact and evaluation in Computational Creativity: A discussion of the Turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and Philosophy*.

Pease, A.; Colton, S.; Ramezani, R.; Charnley, J.; and Reed, K. 2013. A discussion on serendipity in creative systems. In *Proceedings of the Fourth International Conference on Computational Creativity*.

Potter, M. A., and De Jong, K. A. 2000. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computing* 8(1).

Togelius, J., and Schmidhuber, J. 2008. An experiment in automatic game design. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*.

Togelius, J.; Yannakakis, G. N.; Stanley, K. O.; and Browne, C. 2011. Search-based procedural content generation: A taxonomy and survey. *IEEE Trans. Comput. Intellig. and AI in Games*.

Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the Third Workshop on Procedural Content Generation in Games*.

Veale, T. 2012. From conceptual "mash-ups" to "bad-ass" blends: A robust computational model of conceptual blending. In *Proceedings of the Third International Conference on Computational Creativity*.

Williams-King, D.; Denzinger, J.; Aycock, J.; and Stephenson, B. 2012. The gold standard: Automatically generating puzzle game levels. In *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Adapting a Generic Platform for Poetry Generation to Produce Spanish Poems

Hugo Gonçalo Oliveira **Raquel Hervás** and **Alberto Díaz** **Pablo Gervás**
CISUC, Dep. Engenharia Informática D. de Ing. de Software e Int. Artificial Inst. de Tecnología del Conocimiento
Universidade de Coimbra Universidad Complutense de Madrid Universidad Complutense de Madrid
Portugal Spain Spain
hroliv@dei.uc.pt {raquelhb,albertodiaz}@fdi.ucm.es pgervas@sip.ucm.es

Abstract

PoeTryMe was created as a generic system for the generation of poetry that takes into account both semantics, in the form of triplets of relations between concepts, and textual structure, in the form of a grammar of templates extracted from existing poems. It was originally instantiated to generate poetry in Portuguese. The present paper describes an effort to create a different instantiation of PoeTryMe, this time focused on the production of poems in Spanish. The instantiation effort involved the creation of a set of triplets of relations to represent the semantics of Spanish terms, the extraction of a grammar of templates for Spanish from a corpus of Spanish poetry, the application of a different tool for Spanish syllabic division, the integration of the various modules, and several experiments with the resulting system.

Introduction

Existing efforts at the automatic generation of poetry in recent years have uncovered a number of methods for implementing computationally this task, both from a semantic seed (Manurung 2003; Manurung, Ritchie, and Thompson 2012) and from a set of templates (Oulipo 1981; Colton, Goodwin, and Veale 2012), or by combining both (Gonçalo Oliveira 2012; Veale 2013). Yet most existing efforts consist of custom-tailored solutions for specific languages, and it is difficult to envisage what amount of effort might be required to port one of them from the language for which it was originally designed to a different language. The present paper addresses the question of exploring the effort required by the task of adapting an existing generic platform for poetry generation, PoeTryMe (Gonçalo Oliveira 2012), to a language (Spanish) different from the one over which its original instantiation was designed (Portuguese). To produce its poems, the PoeTryMe platform combines both semantic information, in the form of relation triplets between concepts which are used during the selection of content for the poems, and textual information, in the shape of template-like grammar rules used to render as text the selected content. In this sense, it presents a special challenge because resources specific to the new target language need to be produced at both levels, semantics and textual.

The present paper reports on the engineering and development effort for these required resources and presents an exploration of the effect of their characteristics on the performance of the poem generation process. Throughout this effort, the overall goal has been to reuse or adapt existing resources and/or to extract automatically any novel ones, in order to avoid as much as possible the risks of *fine tuning* (Colton, Pease, and Ritchie 2001) inherent in handcrafting them.

Previous Work

Over recent years, many efforts that address the study of creativity from a computational point of view acknowledge the work of Margaret Boden (Boden 1990) as a predecessor. One of Boden's fundamental contributions was to formulate the process of creativity in terms of search over a conceptual space defined by a set of constructive rules. Poetry generation systems explore a conceptual space characterised by form and content. The concept of *articulation* (Gervás 2013) describes the initial analysis of a target artifact with a view to select a particular frame for understanding and decomposing it into parts that can later be used to assemble equivalent instantiations of the same type. This captures both the concept of different parts being joined together in a whole and the concept of allowing the parts to move with respect to one another. Different decisions on articulation can lead to processes that select a particular textual template with which the poems are produced (Oulipo 1981; Colton, Goodwin, and Veale 2012), or reuse a predetermined set of verses (Queneau 1961), or draw upon given sets of lexical items to employ (Gervás 2001), or even rely on a language model to follow, obtained from a reference corpus (Barbieri et al. 2012). The degree of articulation determines a particular conceptual space of possible poems, with poems outside that space being unreachable unless the articulation is changed.

In terms of computational techniques used to explore these conceptual spaces, several solutions have been applied. The generate & test paradigm of problem solving has also been widely applied in poetry generators such as the early version of the WASP system (Gervás 2000b) and the initial work by Manurung (Manurung 1999). Evolutionary solutions have as well been applied (Manurung, Ritchie, and Thompson 2012). An evolution of the WASP system (Gervás 2001) used case-based reasoning (CBR) to build verses for an input sentence by relying on a case

base of matched pairs of prose and verse versions of the same sentence. Alternative approaches to poetry generation include the application of constraint programming techniques (Toivanen, Järvisalo, and Toivonen 2013), which has a great potential for adequately modelling the large amount of constraints that poetry generation deals with.

Although a pleasant sound and a regular rhythm can sometimes make up for poor or inexistent semantics (Gonçalo Oliveira, Cardoso, and Pereira 2007), meaning is also seen as an important feature in computer-generated poetry, whether more precise, vague or figurative. (Veale 2013) describes a system heavily influenced by semantic information, used to drive the poetry generation process, with special focus on figurative language and rhetorical tropes. But different systems handle semantics differently. In evolutionary approaches, among the other constraints, the goal state should consider meaning (Manurung 2003; Manurung, Ritchie, and Thompson 2012), whereas in CBR approaches, words are selected according to a given prose message. In fact, in several systems generation starts with a theme or a set of seed words, which constrain the poem search space and may be seen as setting the semantics of the poem (Wong and Chun 2008; Netzer et al. 2009; Yan et al. 2013). The choice of relevant words may be achieved either with the help of semantic knowledge bases (Netzer et al. 2009; Agirrezabal et al. 2013), by exploring models of semantic similarity, extracted from corpora (Wong and Chun 2008; Toivanen, Järvisalo, and Toivonen 2013; Yan et al. 2013), or both (Colton, Goodwin, and Veale 2012).

PoeTryMe

PoeTryMe, originally presented in (Gonçalo Oliveira 2012), is a poetry generation platform, on the top of which different systems for poetry generation can be implemented. It relies on a modular architecture (see Figure 1), which enables the independent development of each module and provides a high level of customisation, depending on the needs of the system and ideas of the user. It is possible to define the semantic relation instances to be used, the sentence templates of the generation grammar, the generation strategy and the configuration of the poem. In this section, the modules, their inputs and interactions are presented.

Generation Strategies

A Generation Strategy organises sentences according to some heuristics, such that they suit, as much as possible, a target template of a poetic form and exhibit certain features. A poem template contains the poem's structure, including the number of stanzas, lines per stanza and of syllables in each line. Templates may also use a symbol for denoting the target rhyme for the lines. Figure 2 shows poem structure templates for a haiku (5-7-5) and a sonnet (14*10-syllable verses). There is no rhyme pattern specified for the haiku, but each line of the sonnet has a symbol that results in the following rhyme pattern: *ABBA ABBA CDC DCD*.

Each strategy uses the Sentence Generator module to retrieve natural language sentences, which might be selected as poem lines. For the generation of a poem, a set of seed words is provided and used to narrow the set of possible generations, this way defining the generation domain.

```
#haiku
stanza{line(5);line(7);line(5)}

#sonnet
stanza{line(10:A);line(10:B);line(10:B);line(10:A)}
stanza{line(10:A);line(10:B);line(10:B);line(10:A)}
stanza{line(10:C);line(10:D);line(10:C)}
stanza{line(10:D);line(10:C);line(10:D)}
```

Figure 2: Templates with the structure of a haiku and a sonnet with a rhyme pattern.

An instantiation of the Generation Strategy does not generate sentences, but follows a plan to select the most suitable sentences for each line. Selection heuristics might consider features like metre, rhyme, coherence between lines or other, depending on the desired purposes. Some of these features are evaluated with the help of the Syllable Utils.

Syllable Utils As its name suggests, this module consists of a set of operations on syllables. Given a word, Syllable Utils may be used to divide it into syllables, to find the stress, or to extract its termination, useful to identify rhymes.

Sentence Generator

This is the core module of PoeTryMe. It is used to generate meaningful natural language sentences, with the help of:

- A semantic graph, managed by the Relations Manager, that connects words according to relation predicates (see Figure 3 for a very simple semantic graph, centered in the word *poetry*, in Portuguese/English).
- Generation grammars, processed by the Grammar Processor, with textual renderings for the generation of grammatical sentences that express semantic relations.

The generation of a sentence starts by selecting a random relation instance, in the form of a $triplet = \{word_1, predicate, word_2\}$, from the semantic graph. Then, a random rendering for the predicate of the $triplet$ is retrieved from the grammar. After inserting the arguments of the $triplet$ in the rule body, the resulting sentence is returned. A third module, the Contextualizer, keeps track of the instances that were used to generate the lines and may be used to explain the choices made.

Relations Manager The Relations Manager is an interface to the semantic graph. It may be used to retrieve all words related to another, or to check if two words are related by indicating their relation.

To narrow the space of possible generations, a set of seed words is provided to the Relations Manager. This set defines the generation domain represented by a subgraph of the main semantic graph, where the relation triplets should either contain one of the seed words or somehow related words. More precisely, the subgraph will only contain triplets with words that are at most δ nodes far from a seed word, where δ is a

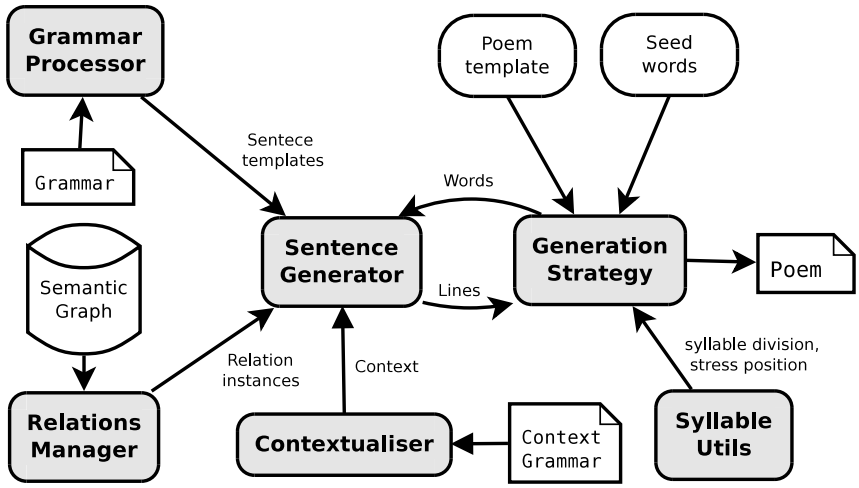


Figure 1: PoeTryMe's architecture

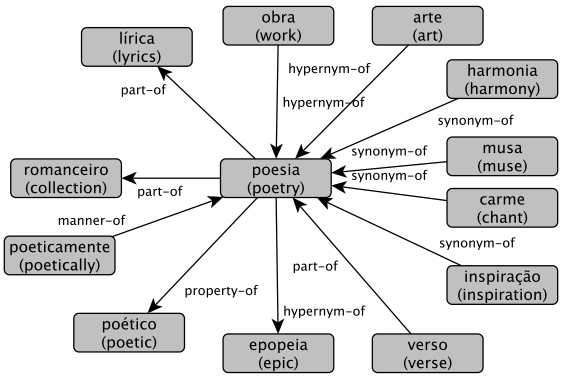


Figure 3: Semantic Graph example

neighbourhood depth threshold. It is also possible to define a surprise factor, ν, interpreted as the probability of selecting triplets one level further than δ.

The number of seed words is open, and it can be enlarged with the top n relevant words for those seeds. For this purpose, the PageRank (Brin and Page 1998) algorithm is run in the full semantic graph. Initial node weights are randomly distributed across the seeds, while the rest of the nodes have an initial weight of 0. After 30 iterations, nodes will be ranked according to their structural relevance to the seeds. The n higher ranked nodes are selected.

Grammar Processor The Grammar Processor is an interface for the generation grammar. Similarly to Manurung (Manurung 1999), it performs chart generation with a chart-parser in the opposite direction. A grammar is a editable text file with a list of rules, whose body should consist of natural language renderings of semantic relations and there must be a direct mapping between the relation names, in the graph, and the rules' name, in the grammar. Besides simple terminal tokens, that will be present in the poem without change, this module supports terminal tokens that indicate the position of the relation arguments (<*arg1*> and <*arg2*>), to be filled by the Sentence Generator. This way, given a relation predicate, the Grammar Processor can retrieve one (or several) renderings for any triplet of that kind.

A very simple example of a valid rule set, with three hypernymy patterns, is shown in Figure 4. These rules could be used to generate sentences as: *a tool like a hammer*, *mango is a delicious fruit*, *man before animal*.

HYPERNYM-OF → a <arg1> like a <arg2>
HYPERNYM-OF → <arg2> is a delicious <arg1>
HYPERNYM-OF → <arg2> before <arg1>

Figure 4: Grammar example rule set.

Contextualizer The ability to explain how its artefacts are created is an important feature of a creative system. PoeTryMe provides this feature by keeping track of all the relation instances that originated each line. Towards the notion of framing (Charnley, Pease, and Colton 2012), these can later be used to contextualize the poem by indicating the relation instances used to form the lines and how they are connected to a word in the generation domain. The context can be a mere list of relation instances or, if a contextualisation grammar is provided, it may consist of a natural language piece of text.

Generating Poetry in Spanish

The process of instantiating the PoeTryMe platform to generate Spanish poems required three separate processes of relevant system resources: (i) construction/adaptation of Spanish lexical resources (morphological lexicon, lexical-semantic knowledge base, syllable division tool); (ii) construction/extraction of a set of template-like renderings; and

(iii) configuration of an appropriate generation strategy. Before describing those processes, some remarks on the requirements and on the flexibility of PoeTryMe are provided.

Remarks on Requirements and Flexibility

As presented in the previous section, PoeTryMe's architecture is very flexible and may be used to generate poetry in different languages and/or on different domains. This applies as long as there are three main tools available, namely a lexical-semantic network, a generation grammar and syllable utilities, all targeting the same language.

The lexical-semantic network, handled as a semantic graph, can be broad-coverage or on any specific domain, as long as it contains relation instances represented as triplets ($word_1$ related_to $word_2$). The generation grammar must contain textual renderings for the relation types covered by the lexical-semantic network. And the syllable tool should at least provide a method for each of the following operations: splitting a word into syllables, stress identification and termination extraction.

As a lexical-semantic network typically contains only lemmatised words, if we want to use also inflected words, a morphological lexicon might also be needed in a preprocessing step. This lexicon should be as broad as possible and provide the part-of-speech (POS) of the words of the target language, as well as other morphological information, such as the gender and number of nouns and adjectives. It can be used for adding inflected words to the lexical-semantic network and contribute to more variation, Moreover, if the generation grammar is learned automatically, with the help of the network, it will enable to learn more complete grammars.

For Portuguese, there have been different instances of PoeTryMe where, apart from different generation strategies, the main differences in external resources were the different sizes of the lexical-semantic network and of the generation grammar. In fact, in the first instantiations of PoeTryMe, the generation grammars were handcrafted. Regarding the adaptation to Spanish, we used a morphological lexicon with the same information as the Portuguese, a syllable tool that performed the same operations, and a lexical-semantic network with the same format. The main difference probably relies on the latter. While, for Portuguese, the lexical-semantic network was extracted automatically from dictionaries (CARTÃO (Gonçalo Oliveira et al. 2011)), for Spanish, it was obtained from a handcrafted resource. This resulted in a larger semantic graph for Portuguese (about 286,000 triplets between lemmas) covering more relation types, and more figurative language, but also more imprecisions. Another obvious difference on the instantiations for different languages results from the different generation grammars, which are learned from different collections of text, each written in its own language.

Lexical Resources Used

In order to handle the inflection of nouns and adjectives (number and gender), the dictionary from FreeLing (Padró and Stanilovsky 2012) has been used as lexicon of Spanish. It contains over 650,000 inflected word forms including nouns, verbs, adjectives and adverbs. For each form, there is information on the lemma, the POS, and inflection details that include the tense of the verbs and the number and gender of the nouns and adjectives.

As the source of relation instances that would build our semantic graph, we have used the Spanish WordNet from the Multilingual Central Repository (MCR) version 3.0 (Gonzalez-Agirre, Laparra, and Rigau 2012). MCR follows the classic wordnet structure, and thus contains synsets and relations between them. The following example shows how a synset relation is converted to relation triplets between words:

Synset relation	{*automóvil, carro, coche*} hypernym-of {*coche_deportivo, deportivo*}
Word triplets	*automóvil* hypernym-of *coche_deportivo* *automóvil* hypernym-of *deportivo* *carro* hypernym-of *coche_deportivo* *carro* hypernym-of *deportivo* *coche* hypernym-of *coche_deportivo* *coche* hypernym-of *deportivo*

A total of 366,125 relation triplets were obtained from the MCR relation tables. Additionally, 58,052 synonymy instances were obtained from the synsets. But we did not use relations of some types, namely those indicating that some word is in a synset gloss (*rgloss*), nor those that reference a previous version of WordNet (*see_also_wn15*). After filtering, we had about 103,000 triplets, held between lemmas, to which we add all possible inflections of nouns and adjectives. In the end, this resulted in 231,296 relation triplets.

To compute the metric scansion of the poems in Spanish in terms of syllables, the corresponding module of the WASP generator of Spanish poetry (Gervás 2000b) was employed. This module is a Java reimplementation of an original set of rules designed as a logic program (Gervás 2000a). For integrating this module in PoeTryMe, an interface with the operations needed by the Syllable Utils module, and shared by the Portuguese tool, was implemented.

Learning Renderings for Semantic Relations

While we could have handcrafted generation grammars with semantic relation renderings, we decided to learn those automatically. This way, a larger and broader set of renderings was obtained, with much less manual labour.

For this purpose, we exploited a collection of human-written Spanish poetry, with poems from an existing anthology of Spanish poetry on the web[1] and also from the WASP knowledge base. Those amounted to 395 poems. The poems of this collection were processed while renderings, represented as grammars rules, were extracted from each line in the human-written poems where two words in a semantic triplet co-occurred. We used the aforementioned 231,296 triplets, collected from MCR.

Generation Strategy

In all our experiments, we have used a generate & test strategy (`GT`), already implemented in previous versions of

[1] `http://www.poemas-del-alma.com/`

PoeTryMe. From the currently available strategies, this achieves rhymes more consistently. For each target line, GT consists of the successive generation of sentences, while keeping only the best scoring ones. Line generation stops either after a predefined number of generated sentences (n), or when a sentence is generated precisely with the target number of syllables and target rhyme, if there is one.

Sentences are first scored according to the absolute difference between their number of syllables and the target number of syllables, for the line. The higher the score, the less suited the sentence's metre is. On the top of this score, there are bonuses for rhymes (-2 points) and penalties for sentences that end with the same word as another in the same stanza. Moreover, we may set a progressive multiplier (π) to increase the number of generations for lines of higher order in the stanzas, this way increasing the probability of rhymes.

Experimentation

Different configurations have been used to test the performance and behaviour of the system. In order to study the relation of input knowledge (both lexical and semantic) and the performance of the system, we have worked with different sets of data in the experiments.

Regarding the discovery of lexical renderings to create the final text, we have trained the system using two different sets of Spanish poems:

- The whole collection of 395 Spanish poems (GR+), which produced a total of 1,285 grammar rules.
- A subset of the previous collection with only 64 poems (GR−), which produced a total of 245 grammar rules. Note that all the grammar rules in GR− are also in GR+.

In addition, different sets of semantic relations were used:

- The whole set of semantic relations from MCR (SR+), which contains 231,296 triplets.
- A subset of SR+ with only synonymy relations (SR−Syn), which contained 55,300 triplets.
- A subset of SR+ with only hypernymy relations (SR−Hyp), which contained 130,669 triplets.

In order to produce comparable results, all the experiments were performed using the same configuration. The goal was to generate a sonnet without a predefined rhyme pattern, using the generate & test strategy (GT), with a maximum of 1000 generated sentences per line. For setting the semantic domain, two values for the neighbourhood depth threshold were tested, $\delta = 1$ and $\delta = 2$, each used to generated a set of 100 poems, always with the surprise factor $\nu = 0.1$. The seed words used were always *amor* (love), *muerte* (death), *suerte* (luck), *vivir* (to live), *sentir* (to feel), and *morir* (to die). These were chosen especially because they were the main topics in the original set of poems. PageRank was not used, so the system only worked with this exact set of seeds.

Experiments on Semantic Relations and Evaluation

Table 1 presents the results obtained regarding the semantic relations used and the evaluation scores of the resulting poems. The former is presented as the size of the explored subgraph, given in terms of the percentage of distinct triplets used from the full semantic graph, in each case – all (SR+), only synonymy (SR−Syn), only hypernymy (SR−Hyp). About evaluation, the presented scores gave -2 bonuses to each line ending with a termination previously used in the same stanza. As the lower the score, the better, this results in a possible best score of -20. We recall that this is not exactly the same evaluation function used in GT. In this strategy, the best possible score for a sonnet would be -12, because every time a rhyme occurs, the target termination is discarded. This however does not prevent the generation of poems as the one in Figure 5, where all lines share the same termination.

δ	GR	SR	% of SR	Evaluation		
				Avg.	Worst	Best
1	GR−	SR+	0.67%	-8.76	-2	-14
1	GR+	SR+	0.77%	-5.19	0	-10
2	GR−	SR+	13.80%	-8.19	-3	-13
2	GR+	SR+	17.78%	-5.93	-1	-12
1	GR−	SR−Hyp	0.56%	-10.86	-6	-19
1	GR+	SR−Hyp	0.61%	-4.68	-1	-9
2	GR−	SR−Hyp	13.04%	-12.03	-7	-19
2	GR+	SR−Hyp	15.30%	-5.53	-1	-10
1	GR−	SR−Syn	0.56%	-6.77	-3	-11
1	GR+	SR−Syn	0.55%	-4.62	0	-9
2	GR−	SR−Syn	2.49%	-8.91	-5	-14
2	GR+	SR−Syn	2.49%	-4.69	0	-10

Table 1: Use of semantic relations (SR) and evaluation results for the different configurations of the experiments

On the semantic relations used, values are consistent among different configurations. When $\delta = 2$ instead of 1, more triplets are used by definition. The increase in the percentages between $\delta = 1$ and $\delta = 2$ is proportional in all the experiments, including those using SR−Syn, where it is smaller because the full semantic graph contains about 23.9% synonymy triplets but 49.0% hypernymy.

Regarding the scores automatically assigned by the system, the average poem score is higher (and therefore less desirable) when more grammar rules are used (GR+). A possible explanation for this counterintuitive behaviour is the increased number of grammar rules without extending the cut-off values for the resulting search. It is therefore possible that the search over the larger conceptual space is cut off prematurely, thereby having less options to find exactly the combination of relations, words and renderings most appropriate from the point of view of rhyme and length. There is not a clear relation between system assigned scores and the number or type of semantic triplets used.

The best scoring poems were obtained with the smaller set of grammar rules (GR−) and only hypernymy relations (SR−Hyp). Figure 5 shows the best poem of experimental runs, along with its rough translation and the experimental configuration that lead to its production. This sonnet uses the same lexical template for all lines and adjusts it by using different pairs of verbs, where one is a hypernym of the other. The rhyme is perfect, but not especially interesting,

mi hospedar no quiere albergar	**Strategy** GT	*my hosting wants no holding*
mi pensar no quiere relacionar	**Renderings, relations**	*my thinking wants no relating*
mi olvidar no quiere arrojar	GR-, SR-Hyp	*my forgetting wants no throwing*
mi morir no quiere soportar	**Generations/line** 1000	*my dying wants no tolerating*
mi ocupar no quiere trabajar	$\delta + \nu$	*my busying wants no working*
mi indicar no quiere informar	1.01	*my indicating wants no informing*
mi recibir no quiere saludar	**PageRank**	*my receiving wants no greeting*
mi tragarse no quiere soportar	no	*my swallowing wants no tolerating*
	Domain	
mi albergar no quiere albergar	*amor* (love)	*my holding wants no holding*
mi resolver no quiere terminar	*muerte* (death)	*my resolving wants no ending*
mi ocupar no quiere trabajar	*suerte* (luck)	*my busying wants no working*
	vivir (to live)	
mi residir no quiere habitar	*sentir* (to feel)	*my residing wants no living*
mi percibir no quiere observar	*morir* (to die)	*my perceiving wants no observing*
mi olvidar no quiere descartar	**Score** -19	*my forgetting wants no discarging*

Figure 5: System configuration in the experiment that obtained the best-scoring sonnet

as all the lines end with *'ar'*.

On the contrary, the worst scoring poems are always obtained with the complete set of grammar rules (GR+) regardless of the semantic relations used. Figure 6 presents one of these poems where the choice of lexical templates is not as repetitive as in the best poem, but there are just no rhymes.

Besides the best and worst-scoring, from all the generated poems, we manually selected a more balanced one, which is shown in Figure 7. This choice was based on the variety of lexical templates used, metre matching, presence of rhymes, and evocative semantics.

Experiments on Grammar Rules

Table 2 has some figures on the experiments regarding the lexical renderings used from the grammar rules and the diversity on their selection. Although more configurations were tested, only those with the complete set of semantic relations (SR+) and $\delta = 1$ are shown. Results with other configurations were similar.

	Distinct renderings	Repetitions		Renderings from GR
		average	maximum	
GR-SR+	57	15.72	259	14.29%
GR+SR+	257	6.83	114	16.31%

Table 2: Use of lexical renderings for different experimental configurations

These results show that the repetition of the same rendering is very common. In both configurations, the average number of repetitions per rendering used is relatively high. The number of repetitions is even higher in the configuration with GR-. This is expected because the number of available lexical renderings is smaller and the ones suitable for the poem must be used more times.

The number of lexical renderings used from the whole set of grammar rules (GR) is quite small in both experiments. In fact, only 15% of the lexical renderings derived from the grammar rules are used in the generated poems. This is due to the nature of the grammar rules derived from the original poems. For example, many lexical renderings correspond to lines in the original poems with significantly more or significantly less than 10 syllables. Therefore, their suitability for generating 10 syllable lines required by our sonnets is low.

In order to test the coverage of the lexical renderings in the generated poems, we carried out a process of obtaining the grammar rules implicit in the generated poems. This was done in an equivalent manner to that used for obtaining renderings from the original set of poems – the poems generated automatically were processed, and grammar rules were extracted for each line where two words in a triplet co-occurred. This led to an interesting finding: new lexical renderings, not in the original generation grammar rules, were discovered in the generated poems. From the total of lexical renderings obtained from the generated poems in both experiments (57 and 257 respectively), about 53% and 39%, respectively, were different from those in the original set of grammar rules. Considering repetitions, respectively 77% and 85% of the lexical renderings used in the poems were in the original set of grammar rules.

New renderings obtained from the generated poems are discovered because of new relations between words in the triplets and words in the final realization of grammar rules. On the one hand, the new renderings could be incorporated as new rules of the generation grammar. This would result in a broader set of more varied and possibly more interesting renderings, worth being explored in the future. On the other hand, we should take some precautions because, while the new renderings would still be grammatically correct, they might be less semantically coherent.

About the most frequent lexical rendering in all the experiments, it is "*mi <arg2> no quiere <arg1>*" (my <arg2> does not want to <arg1>) where both arguments are expected to be verbs, and <arg2> a hypernym of <arg1>. When hypernyms are not used (SR-Syn), the most frequent rendering depends on the configuration. It can be: "*quiero <arg>*

de vivir y poblar la fe de cristo	**Strategy**	*from living and populating the faith of Christ*
quiero quedarse entregar el alma	GT	*I want to stay give up my soul*
murió como un cabo el final	**Renderings, relations**	*he died like a corporal at the end*
quiero identificar distinguir	GR+, SR−Syn	*I want to identify distinguish*
	Generations/ line	
murió como un gusto el afecto	1000	*he died like a pleasure the tenderness*
de poblar y vivir la fe de cristo	$\delta + \nu$	*from populating and living the faith of Christ*
gran muerte de matanza concurriendo	1.01	*great death of slaughter concurring*
quiero perder la vida sucumbir	**PageRank**	*I want to loose my life succumb*
	no	
de vivir y durar la fe de cristo	**Domain**	*from living and lasting the faith of Christ*
trayendo el final a fin dudoso	*amor* (love)	*bringing the ending to dubious end*
y la desaparición y la muerte	*muerte* (death)	*and the dissapearance and the death*
	suerte (luck)	
murió como un afecto el gusto	*vivir* (to live)	*he died like a tenderness the pleasure*
de encontrar y dar la fe de cristo	*sentir* (to feel)	*of finding and giving the faith of Christ*
quiero percibir poner atención	*morir* (to die)	*I want to perceive to pay atention*
	Score	
	0	

Figure 6: System configuration in the experiment that obtained the worst-scoring sonnet

sordos a las estimas y afectas	**Strategy**	*deaf to appreciations and affections*
en el dulce amor ejercitados	GT	*in sweet love exercised*
en los presentes trabajos y cuidados	**Renderings, relations**	*in present works and cares*
hinchen de tristes desgracias el viento	GR+, SR+	*swell the wind with disgrace*
	Generations/ line	
llamar oler sentir les aprovecha	1000	*calling, smelling, feeling profits them*
y cálidos indómitos cordiales	$\delta + \nu$	*and warm cordial untamed*
por los odiosos los amables males	1.01	*by the hated the kind evils*
hinchen de tristes desgracias el viento	**PageRank**	*swell the wind with disgrace*
	no	
ocupará los actos y la pérdida	**Domain**	*it will fill actions and loss*
hinchen de tristes desgracias el viento	*amor* (love)	*swell the wind with disgrace*
que ni la matanza ni el violento	*muerte* (death)	*that neither killing nor violent*
	suerte (luck)	
duras puentes romper cual tiernas cañas	*vivir* (to live)	*hard bridges to break like tender reeds*
mi lamentar no quiere lamentarse	*sentir* (to feel)	*my regret does not to want to regret*
mi ocupar no quiere esforzarse	*morir* (to die)	*my labor does not want to exert*
	Score	
	-7	

Figure 7: System configuration in the experiment that obtained a more balanced sonnet

<*arg*>" (I want <arg> <arg>), where the arguments are synonym verbs; "*murió como un* <*arg*> *el* <*arg*>" (he died as a <arg> the <arg>), where the arguments are synonym nouns; or "*de* <*arg*> *y* <*arg*> *la fe de cristo*" (of <arg> and <arg> the faith of Christ), where the arguments are synonym verbs.

Experiments on the Choice of Seed Words

Another set of experiments has been performed to compare the effect of using different seeds for generation.

First, the seed words used in the previous experiments were changed to study the effect of choosing seeds according to the term-frequency in the original poems. So, the six most and the six least frequent terms occurring in the original collection of poems were used. They were *yo* (I), *gente* (people), *tierra* (dust), *amor* (love), *vida* (life), and *ser* (to be). The least used terms were *abismo* (abyss), *austro* (south wind), *tempestades* (storms), *detenerse* (to stop), *creer* (to believe), and *combatir* (to fight). These experiments were only run with the GR+SR+ configuration with $\delta = 2$. The obtained results shown a big difference regarding the number of semantic triplets used (75,622 vs 9,014), but not very different evaluation scores (on average, -5.78 vs -7).

In another experiment, instead of using a predefined set of seed words, *amor* has been chosen as initial seed and PageRank was used to obtain the top-5 most relevant words. As expected, this set contained the word *amor* itself, and four other words, including some inflections: *amores*, *cariño*, *afectas*, *afecta*. The tested configurations were GR+SR+, GR−SR+ and GR+SR−Hyp, always with $\delta = 2$. With the five previous seeds and these configurations, the best scoring poem was obtained with the complete set of grammar rules (GR+) and with the whole set of semantic relations (SR+).

Discussion

The approach followed by PoeTryMe constitutes an effort to integrate the two classic approaches to poetry generation: it combines a degree of processing to obtain the structure of the poem from a given semantic input (semantic-based generation), and resorts to a grammar of possible renderings of the semantics so obtained to provide the final syntactic form of the resulting poem (syntax-aware generation). This procedure involves a double articulation into a set of semantic elements, each coupled with one or more syntactic elements from a parallel set. This structuring of the process has a certain similarity to the work of (Manurung 1999; Manurung, Ritchie, and Thompson 2012), where logical forms taken as input semantics were paired with TAG constructions that rendered them into text.

The fact that the set of renderings is obtained from a corpus of existing poems has parallelism to a case-based reasoning approach such as the one advocated in (Gervás 2001), but the renderings themselves are closer to the templates used in the Rimbaudellaires (Oulipo 1981).

Nevertheless, the procedure has its limitations. The fact that patterns for rendering are extracted only from contexts where two terms connected by a semantic relation occur within a small distance of one another in the original set of poems is a very strong constraint. As a result only a small percentage of the total set of lines of the original poems is selected into the final set of grammar rules used for rendering. Where an articulation solution based on lines, such as the one applied by (Queneau 1961) would make every line in the original set of poems available to be included in the resulting poems, the articulation solution used for PoeTryMe restricts the conceptual space to be explored to only those lines that contain pairs of terms connected by semantic relations. This has a secondary effect in that available patterns for rendering are very unlikely to originate from lines that are contiguous in the original poems. As a result, the chances are very low for fluent connection to arise during construction between lines that follow one another in the resulting poems.

An additional restriction arises from the fact that each of the grammar rules used for rendering, by virtue of being a template with part of its contituent words already fixed during extraction, imposes a particular number of syllables that acts as starting point for the resulting line. Although different choices of words that will be employed to fill it may produce a slight variation (the final line will be longer if longer words are used, shorter otherwise), particular templates will be better suited for producing lines of length similar to that of the poem from which they originate. This could explain why such a small percentage of the extracted set of possible renderings are employed in the final set of poems, obtained with system configurations for a particular set of restrictions in terms the length of lines. Only grammar rules for renderings obtained from poems with lines of length similar to the target size are likely to be useful in producing new poems.

From the point of view of the perception of creativity that the resulting poems inspire in their readers, the first impression is surprisingly positive. Generated poems have a high degree of variation in spite of being produced by means of templates. This is due to the relative richness of different lexical terms, achieved by the use of the Spanish wordnet. The use of semantic triplets as a constraint when filling in the templates enforces a logical connection between the various ingredients that ensures an impression of cogency. This is the result of constraints at two different levels: the existing link between each template and a particular semantic relation, and the imposition that the two terms used to fill the template be connected to one another by the corresponding relation. The metric pattern imposed by the Generation Strategy ensures that the form of the poem fulfills very closely the breakdown of lines into stanzas, the required number of syllables for each line, and, if availability of resources permits it, even appropriate rhymes.

Concluding Remarks

The present paper reports on the effort to adapt the PoeTryMe generic platform for producing poems in Spanish. This involved mostly the construction, reuse and extraction of the required resources to inform system operation. These resources were integrated with existing operational modules of the platform. The development of resources has been engineered with care to reduce the risk of fine-tuning the system towards a particular set of results. Nevertheless, the resulting set of poems shows heavy evidence of a particular style apparent in the lack of fluent grammar across sentences, a tendency to repeat successful patterns of speech (corresponding to optimal templates for lines), and a preference for infinitives as rhyming solutions.

In more general terms, the set of operational modules, strategies and configurations of input parameters available in the PoeTryMe is much larger than the limited subset that has been explored to obtain the results presented in this paper. Further work can be considered to explore the possible conceptual spaces that may be reached by applying the combinations left untried at the closure of this paper. Among other parameters, it would definitely be interesting to: explore the PageRank way of augmenting the seed words more deeply; generate other poems with a different structure than sonnets, possibly with a predefined rhyme pattern; and to explore the Contextualizer to provide some insights on the contents of the poem, useful to frame it and possibly to evaluate it.

The reported effort constitutes evidence that PoeTryMe can indeed be extended to operate in languages other than Portuguese. The evidence provided by a Spanish instantiation is limited, given the close similarity between the two languages. However, the adaptations required were in no way made easier by those similarities. The possibility of extending the platforms is only restricted by the availability of the lexical, semantic and grammatical resources described, by the existence of a certain affinity between the definition of poetry in the target language (such as being based on length in syllables and rhyme), and by the availability of software solutions for scansion of the desired metrics.

Acknowledgements

This work was supported by projects PROSECCO and ConCreTe. Part of this work was developed during a short term visit funded by the PROSECCO CSA project, Euro-

pean Commission under FP7 FET grant number 600653. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

References

Agirrezabal, M.; Arrieta, B.; Astigarraga, A.; and Hulden, M. 2013. Pos-tag based poetry generation with wordnet. In *Proceedings of the 14th European Workshop on Natural Language Generation*, 162–166. Sofia, Bulgaria: ACL Press.

Barbieri, G.; Pachet, F.; Roy, P.; and Esposti, M. D. 2012. Markov constraints for generating lyrics with style. In *Proceedings of 20th European Conference on Artificial Intelligence (ECAI)*, Frontiers in Artificial Intelligence and Applications, 242, 115–120. IOS Press.

Boden, M. 1990. *Creative Mind: Myths and Mechanisms*. London: Weidenfeld & Nicholson.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks* 30(1-7):107–117.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In *Proceedings of the 3rd International Conference on Computational Creativity*, ICCC 2012, 77–81.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full FACE poetry generation. In *Proceedings of 3rd International Conference on Computational Creativity*, ICCC 2012, 95–102.

Colton, S.; Pease, A.; and Ritchie, G. 2001. The effect of input knowledge on creativity. In *Proceedings of the ICCBR'01 Workshop on Creative Systems*.

Gervás, P. 2000a. A logic programming application for the analysis of Spanish verse. In *1st International Conference on Computational Logic*, 1330–1344.

Gervás, P. 2000b. WASP: Evaluation of different strategies for the automatic generation of spanish verse. In *Proceedings of AISB'00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science*, 93–100.

Gervás, P. 2001. An expert system for the composition of formal spanish poetry. *Journal of Knowledge-Based Systems* 14:200–1.

Gervás, P. 2013. Computational modelling of poetry generation. In *Proceedings of the AISB'13 Symposium on AI and Poetry*, 11–16.

Gonçalo Oliveira, H.; Antón Pérez, L.; Costa, H.; and Gomes, P. 2011. Uma rede léxico-semântica de grandes dimensões para o português, extraída a partir de dicionários electrónicos. *Linguamática* 3(2):23–38.

Gonçalo Oliveira, H.; Cardoso, F. A.; and Pereira, F. C. 2007. Exploring different strategies for the automatic generation of song lyrics with Tra-la-Lyrics. In *Proceedings of 13th Portuguese Conference on Artificial Intelligence*, EPIA 2007, 57–68. Guimarães, Portugal: APPIA.

Gonçalo Oliveira, H. 2012. PoeTryMe: a versatile platform for poetry generation. In *Proceedings of the ECAI 2012 Workshop on Computational Creativity, Concept Invention, and General Intelligence*, C3GI 2012.

Gonzalez-Agirre, A.; Laparra, E.; and Rigau, G. 2012. Multilingual central repository version 3.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, 2525–2529. ELRA.

Manurung, R.; Ritchie, G.; and Thompson, H. 2012. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence* 24(1):43–64.

Manurung, H. 1999. A chart generator for rhythm patterned text. In *Proceedings of 1st International Workshop on Literature in Cognition and Computer*.

Manurung, H. 2003. *An evolutionary algorithm approach to poetry generation*. Ph.D. Dissertation, University of Edinburgh.

Netzer, Y.; Gabay, D.; Goldberg, Y.; and Elhadad, M. 2009. Gaiku: generating haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, CALC'09, 32–39. ACL Press.

Oulipo, A. 1981. *Atlas de littérature potentielle*. Number vol. 1 in Collection Idées. Gallimard.

Padró, L., and Stanilovsky, E. 2012. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference*, LREC'12. Istanbul, Turkey: ELRA.

Queneau, R. 1961. *100.000.000.000.000 de poèmes*. Gallimard Series. Schoenhof's Foreign Books, Incorporated.

Toivanen, J. M.; Järvisalo, M.; and Toivonen, H. 2013. Harnessing constraint programming for poetry composition. In *Proceedings of the 4th International Conference on Computational Creativity*, ICCC 2013, 160–167. The University of Sydney.

Veale, T. 2013. Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit. In *Proceedings of the International Conference on Computational Creativity 2013*, 152–159.

Wong, M. T., and Chun, A. H. W. 2008. Automatic haiku generation using VSM. In *Proceeding of 7th WSEAS International Conference on Applied Computer & Applied Computational Science*, ACACOS '08.

Yan, R.; Jiang, H.; Lapata, M.; Lin, S.-D.; Lv, X.; and Li, X. 2013. I, poet: Automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *Proceedings of 23rd International Joint Conference on Artificial Intelligence*, IJCAI'13, 2197–2203. AAAI Press.

Poetry generation system with an emotional personality

Joanna Misztal[1] **and Bipin Indurkhya**[2]
[1] Faculty of Mathematics and Computer Science, Jagiellonian University, Krakow, Poland
[2] Computer Science Department, AGH University of Science and Technology, Krakow, Poland

Abstract

We introduce a multiagent blackboard system for poetry generation with a special focus on emotional modelling. The emotional content is extracted from text, particularly blog posts, and is used as inspiration for generating poems. Our main objective is to create a system with an empathic emotional personality that would change its mood according to the affective content of the text, and express its feelings in the form of a poem. We describe here the system structure including experts with distinct roles in the process, and explain how they cooperate within the blackboard model by presenting an illustrative example of generation process. The system is evaluated considering the final outputs and the generation process. This computational creativity tool can be extended by incorporating new experts into the blackboard model, and used as an artistic enrichment of blogs.

Introduction

> Not until a machine can write a sonnet or compose a concerto because of thoughts and emotions felt, and not by the chance fall of symbols, could we agree that machine equals brain. (Lister 1949)

This expresses one of the strongest requirements for AI quoted by (Turing 1950). It takes the view that the process of expressing feelings by means of artistic artifacts is a hallmark of human capability. Such requirements have created a challenging task for AI: how to design a computer program that could write a sonnet inspired by its thoughts and emotions. In recent years, various poetry-generating systems have been developed, discussed in more details below, some of which focus only on producing entertaining artifacts, while others simulate the creativity process and incorporate affective computing techniques. However, most of them do not model a sense of *self* capable of expressing its own feelings. The main goal of this project is to take up this challenge and to create a system with an emotional personality. Specifically, we plan to create an empathic system that changes its mood according to the emotions evoked by reading the given text, and expresses them in the form of a poem.

The *affective empathy* has been defined in the psychological literature as the observer's emotional response to the affective state of others (Davis 1983). Similarly, we propose a term *computational empathy* to mean recognition and interpretation of emotions of another person by the computer system. Our work introduces a system with a complex emotional model that attempts to understand affects in human artifacts, and expresses those feelings in the form of a poem. The design considers an optimism rate which is an individual feature of the system influencing its perception of the environment (the text).

This paper is organized as follows. The *Background* section presents existing approaches to sentiment analysis and emotional modeling. It also presents the blackboard idea and other poetry-generation systems. The *Overview* section explains the general idea of the system. The poetry-generation process in our approach is implemented on a blackboard model, which is described in the *System Architecture* subsection. In this approach, the poetry is composed by a group of experts - each of whom has some specific knowledge about the poetry-generation process, and all of them share a global work-space called the blackboard.

The details of the poetry-generation algorithm are presented in the *Poetry Generation Algorithm* section and explained with an illustrative example. The system takes the inspiration for its creativity from the text provided by the user. Key phrases are extracted from the text to determine the theme of the poem, and also to set its sentiment. The key phrase that is found to be the most *inspiring* by the experts is used as the title and main theme of the poem.

The experts start to perform their tasks - *words-generating experts* produce words related to the topic based on their knowledge. Some of them use lexical resources such as synonyms dictionary or word collocations. There is also one expert incorporating a model of emotional intelligence that defines the mood evoked by the given text, and generates words describing this sentiment.

The *poem-making experts* choose words from the pool and try to arrange them into phrases. Each of them uses its own Context-Free Grammar to construct phrases. Some *poem-making experts* use poetic tropes like metaphors or epithets to enrich the style.

The *evaluating experts* select the best phrases according to some constraints, considering the stylistic form.

The *control component* tries to regulate the poem composition by maximizing its diversity and choosing the experts

that were the least frequent before.

Some illustrative results are presented in the *Examples* section. *Evaluation* contains a summary of system's performance in the context of the proposed algorithm and the evaluation of the final outputs.

Current version of the system includes some basic types of experts. However, the blackboard architecture allows the system to be extended by adding new experts. Possible improvements and proposition of new experts as well as possible application of the program are mentioned in the *Conclusions* section.

Background

Sentiment analysis and affective lexical resources

The goal of text sentiment analysis is to extract the affective information or writer's attitude from the source text. Basically the sentiments may be considered within the polarity classification (*positive, negative or neutral*). However, this method does not provide us with a detailed understanding of the author's emotional state, and another approach is needed.

The computational methods for sentiment analysis are usually based either on machine learning techniques such as naive Bayes classifiers trained on labeled dataset, or use lists of words associated with the emotional value (positive-negative evaluation or sentiment score values). In our research we use ANEW database consisting of nearly 2500 words rated in terms of pleasure, arousal, and dominance (Bradley and Lang 2010) for text arousal calculation.

To extract the sentiment evaluation, we use the *Sentistrength* (Thelwall et al. 2010) sentiment analysis tool. It estimates the negative and positive sentiment values in short informal texts (rating both positive and negative scores with 1-5 scale), considering common and slang words, emoticons and idioms. The base of the algorithm is the *sentiment word-strength list* containing terms with 2-5 scale of positive or negative evaluation. The initial, manually-prepared words-sentiments list has been optimized by a training algorithm to minimize the classification error for some training texts. The system also considers a spelling correction algorithm and *booster words list* with terms that can increase or decrease other words' scores (such as *very, extremely*) as well as *negating word list* with terms which may invert emotion value (*not, never*). Additionally, the algorithm uses a list of emoticons commonly used in social web texts, and considers some other stylistic parameters such as questioning and repeated letters.

In our approach, we also use the WordNet-Affect lexical resource (Strapparava and Valitutti 2004) to build a hierarchy of words describing emotional states that are used later to generate the affective content of poems. The lexicon contains WordNet hyponyms of the emotion word, which are a subset of synsets suitable to represent affective concepts correlated with affective words. For example, for the emotional word *compassion,* we can derive a correlated set of words describing this state: *forgive, merciful, excusable, affectionate, commiserate, tender*.

Emotional modeling

As mentioned in (Cambria, Livingstone, and Hussain 2012), the research on human emotions dates back to ancient times. One of the first categorization of emotional states was made by Cicero who separated them in four categories of fear, pain, lust and pleasure. Later studies on this topic were developed by Darwin (19th century), Ekman (who defined six basic emotions as happiness, sadness, fear, anger, disgust and surprise in 1970s) and many others.

One approach towards emotional modeling that has been commonly used by scientists since 20th century is the dimensional model, where particular emotions are represented as coordinates in a multi-dimensional space. One of the first examples is the circumplex model (Figure 1) presented in (Russel 1980). In this model, the horizontal (...) dimension is the pleasure-displeasure and the vertical is arousal-sleep(Russel 1980). In the Whissel's model (Whissel 1989), the 2D spatial coordinates are evaluation (*positive-negative*) and activation (*passive-active*). The author places words from her Dictionary of Affects in Language in this space. Another example of such model is Plutchik's wheel of emotions (Plutchik 2001) consisting of 8 basic and 8 composed emotions placed in the circle, where the similarity of emotions is represented by radial dimension.

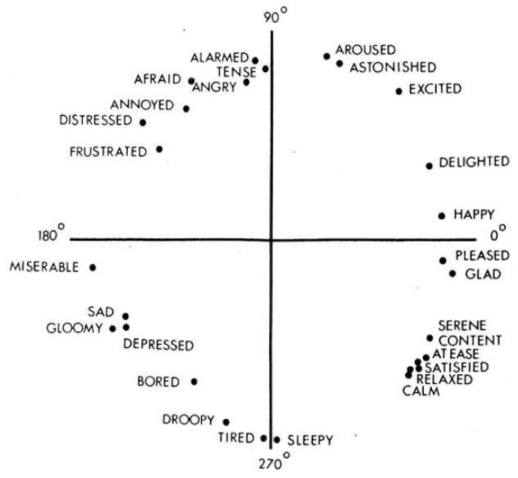


Figure 1: 2D circumplex model of emotions adatpted from (Russel 1980).

The dimensional models are a promising tool for computational modeling of emotions as they provide simple way to measure, define and compare the affective states. They are used in AI systems to simulate the emotional personality as presented in (van der Heide and Trivino 2010; Kirke and Miranda 2013). However, they have some significant limitations as they are based mostly on the verbal representation of affects. As mentioned in (Cambria, Livingstone, and Hussain 2012), they do not allow defining more complex emotions and they do not consider the situation of several emotions being experienced at the same moment.

Blackboard architecture

According to the Global Workspace Theory (Baars 1997; 2003) the brain functioning may be illustrated by a theater metaphor where:

> Consciousness (...) resembles a bright spot on the stage of immediate memory, directed there by a spotlight of attention, under executive guidance. The rest of the theater is dark and unconscious. (Baars 2003)

Thus, in the conscious part the actions are performed by a large number of autonomous specialized modules (the actors).

The blackboard architecture is a model that fulfills the assumptions of GW Theory of mind and therefore has a potential to be used in simulating cognitive processes such as creativity. The model may be visualized by another metaphor (Corkill 1991) of a group of independent experts with diverse knowledge who are sharing a common workspace (the blackboard). They work on the solution together and each of them tries to add some contribution on the blackboard until the problem is solved. The blackboard model is an appropriate solution for problems that require use of many diverse sources of knowledge, or for ill-defined, complex problems. It allows a range of different "experts" – they may be represented as diverse computational models as their internal representation is invisible at the top level.

The idea of using experts representing knowledge has been previously used to simulate cognitive tasks. For example in Word Expert Parser (Small 1979), experts cooperate to provide better understanding of text during the process of conceptual analysis of natural language.

Poetry-generation systems

Since making a system that would produce aesthetically pleasing poems based on predefined templates is not such a difficult task, there exist various poetry-generation programs working in this way. An elaborate example is Kurzweil's *Cybernetic Poet* (Kurzweil 1992), which generates a language model from a set of poems input by the user, and composes new ones in the same style. However, a really challenging task is to make a program that produces the poems in an intentional way. (Gervas 2010) notes that the simulation of human creativity may be significantly different from the original process of creativity itself. Accordingly, there exist various approaches towards computer poetry generation. The McGONAGALL system (Manurung, Ritchie, and Thompson 2012) uses evolutionary algorithms to make a poem that fulfills the constraints on *grammaticality, meaningfulness and poeticness*. ASPERA (Gervas 2001) generates poems with a forward reasoning system. (Toivanen et al. 2012) present a system that creates novelty by substituting words in existing Finnish poetry. In subsequent work, (Toivanen, Jarvisalo, and Toivonen 2013) introduce a constraint programming technique for poetry generation. There are also several projects that incorporate emotional affects in the creation process. (Colton, Goodwin, and Veale 2012) present a corpus-based poetry generator that creates poems according to *days mood* estimated from the news of the day. However, the mood is only defined as *good* or *bad*, without any further refinement of the emotional state. The *Stereotrope* system (Veale 2013) generates the emotional and witty metaphors for given topic based on the corpus analysis. Another interesting approach is MASTER (Kirke and Miranda 2013), which is a tool for computer-aided poetry generation. In this system, a *society* of agents in various emotional states influences each other's moods with their pieces of poetry. The final poem is a result of *social learning*. The poems produced by the system are not meaningful in the usual sense, but they consist of repeated words and sounds that create poeticity.

Among the above-mentioned systems, we can distinguish two different approaches towards modeling the system's *personality*. In the first approach, the system's behavior is determined by some predefined parameters (e.g. in MASTER - agents have initial moods and words). Another alternative is to adapt the emotional state to some environmental factors. This approach is taken by (Colton, Goodwin, and Veale 2012), where the mood of the day is calculated from the sentiment value in daily news. The *Cybernetic Poet* also builds a data-driven model, but it does not exhibit any creative nor emotional behaviors – the system can only replicate the style of the existing poetry.

In our system, we combine both approaches - the emotional state is acquired based on the affective information extracted from the blog text, but it is also dependent on the individual features of the system – the model of emotions and its optimism rate that give the system an individual personality. Hence the external factors are used only as an inspiration for the theme and stimulus for the affective state.

Our approach may be also compared to MASTER, which is also a multi-agent model for poetry generation with emotions. In MASTER (Kirke and Miranda 2013), the agents interact by reciting their own pieces of poetry to each other. Thus, in contrast to our model, they do not share any global knowledge. The mood-defining factor for MASTERs agents is the poetry produced by the societal agents themselves. Hence the method for calculating the emotional state differs from ours, where we extract sentiments from web text. Moreover, all of the agents in (Kirke and Miranda 2013) have the same structure, while in the blackboard model they represent diverse computational units with distinct knowledge sources and roles.

Our approach may be considered as similar to the idea of using specialized families of experts that cooperate during the poetry-generation process incorporated in the later version of WASP (Gervas 2010). Groups of experts work there as a *cooperative society of readers/critics/editors/writers*. However, WASP does not incorporate the blackboard model directly.

Evaluation approaches

The evaluation of any creative system is a nontrivial problem. As the task is not only to generate a satisfying output but also to imitate the creation process, the evaluation needs to consider both the aspects. The most obvious way to evaluate the output is to make a kind of Turing test (Turing 1950) for poetry as in (Kurzweil 1992). In such a test, some computer-generated poems mixed with the human-authored

poetry are presented to the human judges. The score is based on how many poems composed by the system were classified by judges as human-authored. However, the domain-specific Turing test does not consider the evaluation of the creation process. Another approach, taken in FACE descriptive model (Colton, Charnley, and Pease 2011), is based on evaluating the generative act performed by the system and its impact. FACE introduces a set of parameters evaluating the creativity of the program, and considers not only the artifacts produced by the system but also the process of generation, which is essential for creativity evaluation. A *creative act* that satisfies all FACE criteria is denoted by a tuple $< F^g, A^g, C^g, E^g >$, where the *C – concept* means the system taking input and producing outputs denoted by *E – expressions*, the *A – aesthetic measure* is the fitness function evaluating the (concept, expression) pairs with real-number values and the *F – framing information* is the linguistic comment explaining the context or motivation of the outputs.

Overview

The system structure is based on the blackboard model. It consists of a group of experts that represent diverse sources of knowledge, the common blackboard workspace and the control component that regulates the process by choosing one of competing experts that will contribute to the final solution. The modules are described in the *System architecture* subsection.

At the beginning of the poetry-generation process, the input text is set on the blackboard and the agents start to work on it. Each agent has a special role and knowledge and it waits until it finds something on the common workspace that it can use for performing its task. When something interesting appears, the agent processes the information using its individual knowledge and adds new partial solution to the blackboard. The control module decides which agent's contribution should be used for the final poem. The algorithm is explained in more details in *Poetry generation algorithm* subsection along with an illustrative example of the generation process.

System architecture

The system architecture is presented in Figure 2. The main modules of the system are described below.

Blackboard is a common workspace with partial solutions and other information about the problem, shared by the experts. In our system, it consists of:

Text – The input text which is used as an inspiration for the poem. The experts analyze it to define the main theme and sentimental content for the poem.

Constraints – The initial constraints and information about the poem. In the example, we use constraints on the number of lines, the number of syllables in each line and the grammar constraints on tense and person to ensure grammatical consistence of the poem. These constraints are set manually at the beginning of the process or chosen randomly by the system.

Key phrases – Most frequent noun phrases retrieved from the text by one of the experts. Each phrase has its *inspiration* value defined by $W * Cat$, where *W* is number of words that the experts can generate from this phrase and *Cat* is number of non-empty categories of words (categories are explained in *Pool of ideas*).

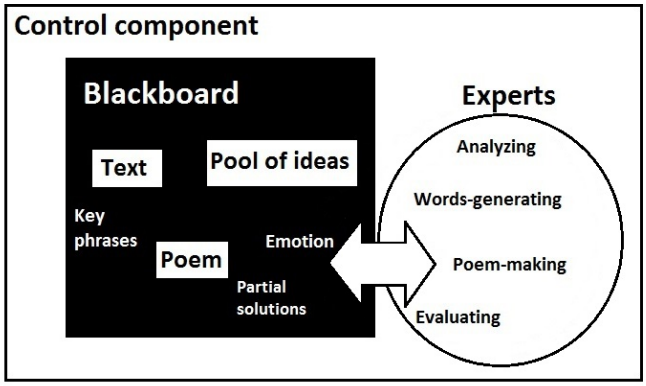


Figure 2: Blackboard architecture used in the system. A group of experts that represent diverse sources of knowledge works on the common blackboard workspace. The control component regulates the process by choosing one of competing experts that will contribute to the final solution.

Topic – The main theme for the poem selected from the key phrases as the phrase with highest *inspiration* score. If there are more phrases with the same value, one is selected at random. Once the topic is set, the experts start to produce their artifacts associated with it.

Emotion – The emotional state for the poem defined by one of the experts by analyzing sentiments in sentences from the text containing the *topic* phrase.

Pool of ideas – A part of blackboard that is used as a workspace for experts. It contains all words and partial solutions produced by the experts. It is also a source of inspiration, as some of them use artifacts generated by others to produce new ones. The expressions in the pool are divided into categories based on their grammatical form and meaning. The main categories are:

Nouns – list of nouns from the topic phrase and their synonyms.
Adjectives – list of adjectives from the topic phrase and their synonyms.
Epithets – lists of adjectives that are most frequently preceding the noun for each noun from the topic phrase.
Verbs – lists of verbs that are most frequently following the noun for each noun from the topic phrase.
Comparisons – lists of nouns that are most frequently following the adjective for each adjective from the topic phrase.
Hypernyms – lists of hypernyms of the noun for each noun from the topic phrase.
Antonyms – lists of antonyms of the words for each noun and adjective from the topic phrase.
Emotional words – words describing the emotional state defined for the poem.
Phrases – list of expressions generated by experts, candidates for the new line in poem.

Poem draft – Current version of the poem consisting of lines. Each line is selected from *phrases* candidates by the *evaluation experts*.

Model of emotions – A 2-dimensional model, where each emotional state is represented by coordinates in *(valence, arousal)* space. The emotions used in the model are WordNet hyponyms of the word *emotion* used in WordNet-Affect lexicon in the hierarchy of emotional categories. The *(valence, arousal)* coordinates for emotional labels in the model have been retrieved from the ANEW database. The choice of emotional categories is based on the lexical resources that we use. It is possible to improve the model by rearranging the categories or their spacial coordinates or to use other more complex models of emotions as mentioned in the *Background* section.

Experts – Independent modules that have access to the common blackboard. They are triggered by events on the blackboard – when they find something that they can use, they try to add new information to the blackboard. Each of them has an individual knowledge and they have diverse roles in the system.

Analyzing experts – Experts that retrieve information from the initial text and add their data to the blackboard.

Keywords expert – Extracts the most frequent noun phrases from the text and adds them to the *key phrases* section on the blackboard.

Emotion expert – Defines the emotional state for the poem and sets the *emotion* on the blackboard. As the whole text may be long, and the emotional attitude may vary within it, the sentiments are considered only for the sentences containing the topic of the poem. Sentiments are calculated in terms of *valence* (positive/negative evaluation of pleasure scaled to -5 to 5) and *arousal* (passive/active scaled to -5 to 5) levels. The valence of the text is calculated by using SentiStrength tool, which estimates the negative and positive sentiment strength in sentences based on the *Emotion Lookup Table*. However, as we want our system to represent an independent emotional intelligence, it should perceive the affects of the text in a more subjective way. Therefore, we introduced the *optimism rate* which is a parameter set at the beginning of the algorithm (or chosen randomly) that biases the valence result so that the perception of the text may be more optimistic or pessimistic. Thus, the final valence estimated by the program is given by:

$$V = \alpha_{opt} \cdot \sum_{s \in Text} Sent_{pos} + (2 - \alpha_{opt}) \cdot \sum_{s \in Text} Sent_{neg} \quad (1)$$

where α_{opt} is the optimism rate of the system (between 0,7 and 1,3), $\sum_{s \in Text} Sent_{pos}$ and $\sum_{s \in Text} Sent_{neg}$ is the sum of positive and negative sentiments respectively for all sentences in the text.

The arousal value has been calculated with use of ANEW. The algorithm combines the average ANEW arousal value for the words in text. The basic formula for arousal calculation:

$$A = (\sum_{w \in Text} A_{ANEW}(w))/length(Text) \quad (2)$$

where $A_{ANEW}(w)$ is the arousal value of word w retrieved from ANEW database. However, the sentiment in the text may be expressed not only within words but by other features of the text, similarly to expressing emotions with voice intonation in a spoken message. For example, the text *"That's great..."* can be perceived as less arousing than the same words written in a different way: *"That's GREAT!!!"*. Hence, the arousal calculation uses a punctuation-sensitive algorithm, i.e. some punctuation marks in the text increase the arousal value, while others decrease it. The calculated arousal score may be modified according to the rules:

$$f(A) = \begin{cases} A - 1 & \text{if "..." in text} \\ A + 1 & \text{if "!" in text or word in capitals in text} \\ A + 2 & \text{if "!!!" in text} \end{cases} \quad (3)$$

where A is the text arousal.

Once the valence and arousal of the text are calculated, the emotional state is defined as follows:

$$emotion = \arg\min_{x \in S} d((v_t, a_t), (v_x, a_x)), \quad (4)$$

where *emotion* is the current emotional state, *S* is the set of all emotional states from the *model of emotions*, v_t and a_t are the valence and arousal of the text, v_x and a_x are valence and arousal of the emotional state and $d(x_1, x_2)$ is Euclidean distance.

Words-generating experts – Experts that have some lexical knowledge. They generate words associated with the topic and add them to the *pool of ideas* sections.

WordNet expert – generates synonyms, hypernyms and antonyms for nouns and adjectives based on the WordNet lexical resource (Miller 1995). Adds to *nouns*, *adjectives*, *hypernyms*, *antonyms* sections of the pool.

Collocation expert – generates words that are frequently used together with given nouns and adjectives. Retrieves information from 2gram model of texts from Brown Corpus. Adds adjectives that describe nouns to the *epithets* section, verbs that follow nouns to the *verbs* section and nouns that follow adjectives to the *comparisons* section of the pool.

Emotional-Words expert – generates words that describe the emotional state defined for the poem. The affective words are derived from WordNet Affect as the hyponyms of given category name. For instance, if the emotional state was defined as *calmness*, the generated set of words would contain *peace, calm, tranquilly, easiness, cool, still*.

Poem-making experts – Experts that compete to produce new lines for the poem. They use partial solutions generated by other experts in the *pool of ideas* to produce new phrases. Their outputs are added to the *phrases* section of the *pool* and are evaluated by the *selection experts*. These phrases may be also extended by others. These experts are triggered when they find something on the blackboard that they could use for their phrases. They can generate a number of phrases proportional to their *importance factors* that are set manually at the beginning of the algorithm. Some of these experts compose stylistic forms typical for poetry.

Grammar experts – Experts that use Context-Free Grammar rules to produce phrases.

Apostrophe expert – Generates apostrophes with the noun,

its description and hypernym. For example: *O life the heavenly being*
Comparison expert – Generates comparisons for adjectives using nouns that are most frequently described by them. For example: *As deep as a transformation*
Epithet expert – Generates expressions with a noun and its epithets or emotional adjectives. For example: *marvelous sophisticated fashion*
Metaphor expert – Generates metaphors by comparing the person to an object. For example: *You were like the downtalking style*
Oxymoron expert – Composes phrases with antonym words. For example: *good and bad*
Rhetorical expert – Composes rhetorical questions about noun, or noun and its epithets. For example: *why was the style so peculiar ?*
Sentence expert – Generates sentences according to its grammar rules. Uses all the words categories, and also the emotions describing words. For example: *She loved the peaceable new york*

Recycling experts – Experts that generate new phrases by transforming phrases generated by other experts.
Exclamation expert – Generates a new phrase by adding "!" exclamation mark to the phrase from the pool.
Overflow expert – Generates a new phrase by breaking phrases from the pool into two lines.
Repetition expert – Generates a new phrase by repeating a phrase from the pool .

Selection experts – Experts that select the best solutions according to given constraints and heuristics.
Inspiration expert – Selects the topic for the poem from the set of key phrases according to formula:

$$Topic = \underset{x \in Keyphrases}{\arg\max} \; W_x \cdot Cat_x, \tag{5}$$

where W_x is the number of words that the experts can generate from this phrase, and Cat_x is the number of non-empty categories to which these words belong.
Syllables expert – Selects phrases that have the number of syllables closest to the target number of syllables for the current line in poem.

$$Lines = \underset{x \in phrases}{\arg\min} \; |S_x - S_t[i]|, \tag{6}$$

where i is current line number, S_x is number of syllables in phrase x, $S_t[i]$ is number of target syllables for line i The syllables are counted using the CMU Pronouncing Dictionary combined with the syllables-estimating algorithm used for words that are not included in the dictionary.

Control component – the unit responsible for setting initial constraints for the poem, setting experts' probabilities and evaluation expert whose contribution should be used for the current line of poem. In the current version of the system, the constraints are set for the number of lines and the numbers of syllables in each line, grammar form and tense. The stylistic constraints are selected at random from a set of templates. The experts' importance factors are chosen manually, and are used during the generation process when an expert produces a number of phrases proportional to its importance factor. The control module also tries to maximize the diversity of the poem by giving preference to the artifacts generated by those experts that contributed less frequently before. For instance, if the poem consists of two lines generated by the *grammar expert* and one by *apostrophe expert*, and for the fourth line the *grammar expert* is competing with the *oxymoron expert*, the control component will give preference to the *oxymoron expert.*

Poetry generation algorithm

We present below the generation process along with an illustrative example. The algorithm can be divided into following phases:

Modules initialization Blackboard is initialized with the text input by the user. The form of the poem is selected from a set of templates, and grammar constraints are defined for stylistic consistency.

Text:

When someone leaves you, apart from missing them, apart from the fact that the whole little world you've created together collapses, and that everything you see or do reminds you of them, the worst is the thought that they tried you out and, in the end, the whole sum of parts adds up to you got stamped REJECT by the one you love. How can you not be left with the personal confidence of a passed over British Rail sandwich? [1]

Constraints:

Number of syllables in lines: (line 1: 8; line 2: 8; line 3: 8; line 4: 8)
Grammar form:
Person: *she*; Tense: *present*;

Poem-making experts are initialized with individual *importance factors* varying from 1 to 5, determining how many phrases they can generate in each turn. The default values presented below may be modified manually.

Poem making experts importance factors: *Apostrophe expert: 2, Comparison expert: 3, Epithet expert: 5, Metaphor expert: 2, Oxymoron expert: 2, Rhetorical expert: 3, Sentence expert: 5, Exclamation expert: 1, Overflow expert: 1, Repetition expert: 1.*

Emotional expert is initialized with a random *optimism factor* between 0,7 and 1,3. A higher value means a more optimistic attitude.

Optimism factor: 0,84

Topic selection The topic is chosen as the most *inspiring* key phrase from the text. To define it, first all key phrases are retrieved and evaluated with the *inspiration* score.
Keywords expert extracts key phrases as the most frequent phrases consisting of a noun and descriptive adjectives.
Key phrases:

[someone, end, whole little world, whole sum, british rail sandwich, parts, personal confidence, fact]

[1] `http://www.jaceandjenelle.com/my-personal-blog.php`

Words-generating experts estimate how many words they can produce from each key phrase. The *inspiration* for each phrase is calculated according to formula (5). The *inspiration expert* selects the *most inspiring* phrase for the topic.

Inspirations: *whole little world: 6920, personal confidence: 3920, whole sum: 3880, someone: 2324, parts: 1918, fact: 1512, end: 910.*

Poem topic: *Whole little world*
Emotional expert defines the emotional state for the poem. The sentiments are retrieved from sentences containing the topic phrase. The expert calculates *valence* and *arousal* according to (1), (2) and (3). Then the emotional state is defined as in (4).

Sentences containing topic phrase :

When someone leaves you, apart from missing them, apart from the fact that the whole little world *you've created together collapses(...).*

Valence: -0.94;*Arousal:* 2.0; *Emotional state: despair*.

Words generation Once the topic and emotional state for the poem are defined, the *words-generating experts* start to produce their ideas. They store their artifacts under appropriate categories in the *pool of ideas* section of the blackboard.

Pool of ideas:

Nouns – *[macrocosm, existence, universe, cosmos, world, creation]*
Adjectives – *[whole, little, small]*
Verbs – *existence: [loses, reflects, becomes, fails, is, belongs], world: [centered, admired], universe: [is, had, are, was], creation: [is, does, prevents]*
Epithets – *world: [little, contemporary, real, previous] , existence: [happy, celestial, historical], universe: [interdependent, entire], creation: [own, inventive, artistic*
Comparisons – *whole: [lines, block, incident, country]*
Hypernyms – *existence: [state], world: [natural object], creation: [activity]*
Antonyms – *whole: [fractional], little:[big]*
Emotional words – *[pessimistic, cynical, resignation, discourage, hopeless]*

Phrases generation As the words start appearing in the *pool of ideas*, the *poem-making experts* start to produce phrases for new lines according to grammar constraints. They add their artifacts to the *phrases* section.

Phrases:

Epithet Expert: *corporate existence, great world*
Apostrophe Expert: *oh world the little natural object*
Sentence Expert: *the creation prevents abjectly, she likes the hopeless, she loves the pessimistic cosmos*
Comparison Expert: *as whole as a story, whole like a convocation*
Metaphor Expert: *she is like the human existence*
Exclamation Expert: *as whole as a story!*
Rhetorical Expert: *why is the existence so nonfunctional?*
Oxymoron Expert: *whole but fractional*

Line phrase selection When all experts finish their generation, the phrases that fulfill the line constraints best are selected by *selection experts*. Then the *control module* makes the final selection judging by the experts' frequencies in former lines. The same algorithm is repeated for each line of the poem.

Generating line 4. Target syllables number: 8 *Poem:*

line 1: *what is the jewish cosmos?* (Rhetorical Expert)
line 2: *o existence the daily state* (Apostrophe Expert)
line 3: *perceptual physical world* (Epithet Expert)

Syllables expert – best phrases candidates:
happy corporate existence (Epithet Expert) : 8,
she sees the pessimistic world (Sentence Expert): 8
Control module – selecting less active experts in former lines generation:
Epithet Expert: 1 line,
Sentence Expert: 0 lines
Line phrase selection: *she sees the pessimistic world* (Sentence Expert)

Examples

Below we present some example outputs of the system inspired by three input texts. We include some remarks on the interpretation of the produced poems, which are further analyzed in *Evaluation* section.

Compassionate poem about the life

Inspired by the text:

With the holiday craziness yesterday, and having to work, i didn't get to finish posting all of my thankfulness pictures. So you might see them pop up over the next few days.this morning i am thankful for the adult men in my life. My dad and mr P. i am fortunate to have both of them in my life to encourage me, support me, take care of me, and love the kids with all of their hearts.[2]

Topic: *Life*, Emotion: *compassion*
Poem:

O life the personal beingness
You are like the simple life!
Musical sacrificial life
You are like the general life
You see the excusable life
Emotional musical life
O life the heavenly being

Remarks:
The topic *Life* provided a wide range of epithets associated with the main phrase. Produced output presents a big lexical diversity of adjectives describing *life* what creates the poetical stylistics. The apostrophes are used in the first and

[2] http://storyofmylifetheblog.blogspot.com.es/

last lines of the poem, giving it a closed form. This effect was accidental, however it could be an interesting improvement to order experts in this way. The emotional state is expressed only by the adjective *excusable* as numerous adjectives dominated the emotional words.

Angry poem about the end

Inspired by the text:

> *I remember being endlessly entertained by the adventures of my toys! Some days they died repeated, violent deaths, other days they traveled to space or discussed my swim lessons and how I absolutely should be allowed in the deep end of the pool, especially since I was such a talented doggy-paddler.* [3]

Topic: *Deep end*, Emotion: *anger*
Poem:

I knew the undisrupted end
I was like the various end
As deep as a transformation
O end the left extremity
Objective undisrupted end
I hated the choleric end
O end the dead extremity

Remarks: The emotional state for the poem is *anger*, which may correspond to some negative expressions in the text (*died, violent deaths, deep end*). The mood is expressed in the poem by words *choleric* and *hated*.

Fearful poem about the way

Inspired by the text:

> *Lately everyone has been wondering "Is Jenelle and Gary going to get back together?!". NO! He is living his life and I'm living mine. We are both happy with our lives the way they are at the moment, I know for me at least I'm EXTREMELY happy. Gary might of been tweeting things because he might of been jealous in a way that I was dating Courtland but he agrees to stop today.* [1]

Topic: *Way*, Emotion: *fear*
Poem:

O mode the symbolic property
Quickest moderate way
She was like the mode
She seemed hysterical because the way left

Remarks: We can observe here that the system does not do well with ambiguous words. The *way* is once interpreted as *property* or *mode* but the algorithm does not consider what was the phrase context in the text. However, the poetry may allow some less strict interpretations of meaning as the ambiguity can be used as an intentional poetical operation.

[3] `http://hyperboleandahalf.blogspot.com`

Evaluation

The evaluation of a creative system is a difficult and ill-defined problem. As the goal is not only to generate a satisfying output but also to imitate the creation process, the evaluation needs to consider both the aspects.

Output evaluation

As the human interpretation of poetical artifacts is a subjective process, we claim that the *Turing tests* are not reliable ways to evaluate poetry. However, the system requires some kind of evaluation for its outputs. Hence, according to (Manurung, Ritchie, and Thompson 2012) we assume that generated texts need to meet the constraints of *grammaticality, meaningfulness and poeticness* to be considered as valuable poetic artifacts. Below we evaluate our outputs along these dimensions.

Grammar The consistency of grammatical form is controlled by the constraints on person and tense. Use of *Context-Free Grammars* as the knowledge for poem-making experts provides the poem with a proper grammatical structure. As we can observe in *Examples* section, the outputs generally represent proper grammar. Some minor mistakes are caused by mis-classification of ambiguous words. This problem could be solved by improving the text-analyzing phase so that the key phrases are analyzed considering the context in which they are used.

Meaning The meaning of the poem is derived from the lexical (WordNet) and statistical (Brown Corpus analysis) associations of words in the topic phrase. Poems contain synonyms, hypernyms and antonyms as well as words that are most commonly used together with the main phrase. This combination results in a higher diversification of produced poems. The choice of the topic as the most *inspiring* phrase causes more possibilities to produce varied and meaningful poems. Also, the use of phrases describing the emotional state gives the impression of intentionality in produced compositions.

However, as observed in the last example in *Examples* section, the algorithm lacks handling of ambiguous phrases. Thus. the interpretation may differ from the meaning of the phrase in the initial text, and may not be consistent throughout the poem. This problem could be resolved by analyzing the context of words in the text but, as mentioned above, for poetry the ambiguity may sometimes be perceived as an intentional operation.

Poeticness The poetic form of generated poems is created by two main factors – the experts using poetical forms for their phrases and the stylistic constraints for lines. As can be observed in presented outputs, the poetical forms used by experts, such as epithets and apostrophes, make an important contribution to the overall perception of poetical composition.

The stylistic constraints in the current version consider only the number of syllables for each line, and are used for selecting best candidates for lines. This approach does not allow more elaborated poetical operations, such as the use of rhymes or rhythm. However, this could be easily improved

by adding new *selection experts* to the blackboard architecture. Each expert should use some heuristics to evaluate the competing phrases and the final selection should respect all criteria.

Another important aspect influencing the poetical character of outputs is the use of emotionally rich words that evoke imagery and are typical for poetical expressions.

Output evaluation summary

As presented above, the products of the system meet the triple constraints on *grammar, meaning and poeticness* to some extent. Further improvements of these factors in the system should include context-based analysis of words and introducing more stylistic constraints for the poetical form.

Model evaluation

As the main focus of computational creativity systems is to produce their outputs in an intentional way, the generation process should consider this as an important concern for evaluation. We propose evaluation of our system using the FACE model (Colton, Charnley, and Pease 2011) which is aimed at evaluating *creative acts* performed by a computer. The details of the model are presented in the *Background* section. We present below how our system architecture corresponds to these criteria.

Concept and concept expression In our case, the *concept* is the blackboard architecture with the set of experts cooperating to compose the poem. The motivation to use the blackboard architecture as presented in *Background* section is the Global Workspace theory which compares the brain functioning to a group of independent modules sharing a public workspace. The program takes a text as an input and produces the *concept expressions* in the form of poems. The outputs are evaluated in the *Output evaluation* subsection. In this approach we could also consider each expert as an independent *concept* producing its own *expressions* as partial solutions for the problem.

Aesthetic measure The *aesthetic measure* in the system may be considered as the heuristic functions evaluating candidates for new lines in poem. Each pair expert (*concept*) – phrase (*expression*) is evaluated respecting the stylistic constraints (6) and the expert's frequency before. The result is a real number. Another measure is used for topic selection — each key phrase is evaluated according to its *inspiration* value as in (5).

Framing information The *framing information* in system might be found only in the name of the emotional state defined according to the model of emotions (4). This output provides some information about the context of the poem.

FACE evaluation summary

As presented above, the generation process performs the *generative acts* of the form $< A^g, C^g, E^g >$. The F^g is provided by description of the emotional state only, but it may not be sufficient to satisfy the *framing information* criterion.

Conclusions

We proposed a system that is capable of expressing its own feelings in the form of a poem. The emotional state is generated by empathic perception of the text, and the mood is modulated by the optimism rate factor given to the character.

The blackboard architecture used in the system provides an effective way to model creativity: it is easily extensible with new linguistic resources and stylistic constraint. It could even incorporate experts representing other existing poetry generation systems such as *Stereotrope* for generating metaphors. Moreover, the blackboard model is a computational representation of Global Workspace theory of mind, which makes it a promising tool for simulating cognitive processes.

The poems produced by the system generally satisfy the triple constraints of *grammar, meaningfulness and poeticness*. However, in the future work, more attention should be paid to the context of analyzed words. According to the FACE evaluation, our system performs the creative acts of the form $< A^g, C^g, E^g >$. The *aesthetics measure* could be improved by defining more stylistic constraints for the poem.

The approach presented here can also be applied for generating poetry based on blogs.

References

Baars, B. J. 1997. In the theater of consciousness. *Journal of Consciousness Studies* 4.

Baars, B. J. 2003. The global brainweb: An update on global workspace theory. *Science and Consciousness Review*.

Bradley, M. M., and Lang, P. J. 2010. Affective norms for english words(anew): Affective ratings of words and instruction manual. Technical report, NIMH Center for the Study of Emotion and Attention University of Florida.

Cambria, E.; Livingstone, A.; and Hussain, A. 2012. *The Hourglass of Emotions*. Springer.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The face and idea descriptive models. In *Proceedings of the International Conference on Computational Creativity*.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-face poetry generation. In *Proceedings of the International Conference on Computational Creativity*.

Corkill, D. D. 1991. Blackboard systems. *AI Expert* 6.

Davis, M. 1983. Measuring individual differences in empathy: Evidence for a multidimensional approach. *Journal of Personality and Social Psychology* 44(1):113–126.

Gervas, P. 2001. An expert system for the composition of formal spanish poetry. *Knowledge-Based Systems*.

Gervas, P. 2010. Engineering linguistic creativity: Bird flight and jet planes. In *Proceedings of Workshop on Computational Approachesto Linguistic Creativity*.

Kirke, A., and Miranda, E. 2013. Emotional and multi-agent systems in computer-aided writing and poetry. In *Proceedings of the Artificial Intelligence and Poetry Symposium*.

Kurzweil, R. 1992. *A (Kind of) Turing Test*. The MIT Press.

Manurung, R.; Ritchie, G.; and Thompson, H. 2012. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental and Theoretical Artificial Intelligence*.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Communications of the ACM*.

Plutchik, R. 2001. The nature of emotions. *American Scientist* 89(4):344–350.

Russel, J. A. 1980. A circumplex model of affect. *Journal of personality and social psychology*.

Small, S. 1979. Word expert parsing. In *Proceedings of the 17th annual meeting on Association for Computational Linguistics*.

Strapparava, C., and Valitutti, A. 2004. Wordnet-affect: an affective extension of wordnet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation.*

Thelwall, M.; Buckley, K.; Paltoglou, G.; and Cai, D. 2010. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the International Conference on Computational Creativity*.

Toivanen, J. M.; Jarvisalo, M.; and Toivonen, H. 2013. Harnessing constraint programming for poetry composition. In *Proceedings of the International Conference on Computational Creativity*.

Turing, A. 1950. Computing machinery and intelligence. *Mind*.

van der Heide, A., and Trivino, G. 2010. Simulating emotional personality in human computer interfaces. In *Proceedings of IEEE International Conference on the Fuzzy Systems (FUZZ).*

Veale, T. 2013. Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit. In *Proceedings of the International Conference on Computational Creativity*.

Whissel, C. 1989. The dictionary of affect in language. *Emotion: Theory, Research, and Experience*.

Pemuisi: a constraint satisfaction-based generator of topical Indonesian poetry

Fam Rashel[1] and **Ruli Manurung**[2]
Faculty of Computer Science
Universitas Indonesia
Depok 16424, Indonesia
[1]fam.rashel@ui.ac.id, [2]maruli@cs.ui.ac.id

Abstract

Pemuisi is a poetry generation system that generates topical poems in Indonesian using a constraint satisfaction approach. It scans popular news websites for articles and extracts relevant keywords that are combined with various language resources such as templates and other slot fillers into lines of poetry. It then composes poems from these lines by satisfying a set of given constraints. A Turing Test-style evaluation and a detailed evaluation of three different configurations of the system was conducted through an online questionnaire with 180 respondents. The results showed that under the best scenario, 57% of the respondents thought that the generated poems were authored by humans, and that poems generated using the full set of constraints consistently measured better on all aspects than those generated using the other two configurations. The system is now available online as a web application.

Introduction

Poetry is a form of literature with an emphasis on aesthetic aspects such as alliteration, repetition, rhyme and rhythm, which distinguishes it from other literary forms. In poetry, the specifically chosen wording is infused with much more meaning and expressiveness, hence the difficulty in translating poetry compared to translating prose.

Poetry generators are systems capable of automatically generating poetry given certain restrictions and contexts. Gervás (2002) presents an overall evaluation of various poetry generators. Other notable works include Manurung (2003), Colton et al. (2012), and Toivanen et al. (2013).

Colton et al. (2012) proposes an architecture for poetry generation that is able to generate poetry along with a commentary on the various decisions it chose in constructing the poem. Toivanen et al. (2013) show how constraint logic programming can be used to generate poems that satisfy various poetic and linguistic constraints.

Our system, *Pemuisi* (a rather archaic Indonesian word meaning *poet*), combines the architecture and approach proposed by Colton, particularly the fact that generated poems are based on current news articles, with the constraint satisfaction-based approach of Toivanen, and generates poems using a combination of handcrafted and automatically extracted Indonesian language resources.

The main contribution of this work, aside from the combination of these approaches, and the adaptation to the Indonesian language, is the user evaluation that was conducted, as both Colton et al. (2012) and Toivanen et al. (2013) present no user evaluation.

In the *Background* section below, relevant previous work will be presented, especially the generator described in Colton et al. (2012). The *Language Resources* section introduces the various language resources required by our system. *Pemuisi* utilizes two kinds of language resources, templates and slot fillers. Slot fillers are divided into poetic words and keywords. Each of these language resources play their own role in satisfying poetic properties. In the *Constraint Satisfaction Poetry Generation* section, we present our constraint satisfaction approach to poetry generation. Poetic features such as number of lines, syllable counts, and rhymes are defined as a set of constraints. Hence, the system will try to satisfy the constraints while composing the poem. The *Experiments and Evaluation* section details the various experiments we conducted. We took the output for evaluation through online questionnaire with 180 respondents. The results were analyzed based on several criteria, such as structure, topic, and message of the poem. Finally we briefly discuss our implementation of a live web application that continuously monitors popular news websites for articles and produces corresponding poems.

Background

Manurung (2003) claims that poetry must satisfy the three properties of meaningfulness, grammaticality, and poeticness. The property **meaningfulness** states that a text should aim to convey a message or concept that has meaning when readers try to interpret the text. This property could be a common element for any text, not just poetry. The property **grammaticality** states that a poem must comply with linguistic rules defined by a given grammar and lexicon. This property is also one of the most common needs that must be met by any natural language generation (NLG) system. The last one is **poeticness**. This property states that poetry must contain strong characteristics of poetry elements, e.g. pho-

netic features such as metre and rhyme. This is the key property to distinguish poetry from other texts. Such requirements imply that it is insufficient for poetry generation systems to simply produce random words.

Colton et al. (2012) states that the first poetry generator to be developed is most probably the *Stochastische Texte* system developed by Lutz that utilizes a small lexicon consisting of sixteen subjects and predicates from Kafka's *Das Schloβ*. The system randomly chooses words from Kafka's works and fits them into a grammatical template that previously has been defined.

Other poetry generators can be grouped into several categories. Referring to Gervás (2002) who provides a taxonomy of poetry generation systems based on the approach and techniques used, there are at least four different categories, namely (i) template-based systems, (ii) generate and test systems, (iii) evolutionary-based systems, and (iv) case based resoning systems.

Another perspective from Colton et al. (2012) is that most existing poetry generation systems behave more as assistants, with varying degrees of automation, for the human user who has provided the majority of the resulting context of the poem. Departing from this view, they propose a fully autonomous computer system poet, which we refer to as Full-FACE. Full-FACE is a corpus-based poetry generator that utilizes various resources such as lexical databases, simile corpus, news articles, pronouncing dictionary, and sentiment dictionary. Given these resources, the system is able to generate poetry independently, to the extent of deciding its own form of poetry such as the number of lines, rhyme structure, message, and the theme of the poetry. Overall, this system consists of several stages. The first is **retrieval,** where the various resources needed to produce poetry are gathered, i.e. the Jigsaw Bard simile corpus, a set of constraints, and a collection of keyphrases from Guardian news articles that will be the topic of poetry. Then we go to **multiplication** stage, where the aforementioned resources are permutated to obtain variations in order for the resulting poetry to be more expressive. For example, the existing simile corpus yields similes in the form of a triple <*object, aspect, description*>, which contains information about the simile, e.g. the tuple <*child, life, happy*> represents the simile "*as happy as a child's life"*. Multiplication is done by applying three kinds of substitution methods: using the DISCO corpus, the simile corpus, or WordNet to find words that are similar. During the **combination** stage, Full-FACE produces lines of poetry through combining simile corpus, the simile multiplication result, and article keyphrases. This combination is done by following a certain template. For example, there is a keyphrase "*excess baggage*" that match the simile "*the emotional baggage of a divorce*" can be applied to the process of combination into line poem "*Oh divorce! So much emotional excess baggage*" in accordance with the specifications of the template. Finally, the results of the previous process are collated in accordance with the user-given constraints or existing template in the last stage called **instantiation**.

A fully autonomous computer system poet was established by handing over high-level control to the system itself. This was done by the system with context generation alongside with the commentary. Context generation is a process of how context, topics, templates to structure the poetry, such as lines and rhyme patterns, determined by the system to form poetry. In order to deliver the context, commentary generation is a process to produce a commentary on the poetry made. In general, the comments contain the condition of the heart/emotions at the time of making the poetry, a summary of the article reference, and how the process of writing poetry.

Language Resources

Our system requires at least two types of resources, templates and slot fillers. These resources are necessary pieces for the system to make poetry. To prepare these resources we need to go through several processes. Hereby is the explanation of each process.

Templates

A template is a ready-made sentence (canned text) that has one or more slots to be filled by certain words. Each slot is associated with a part-of-speech tag, such as noun, verb, adjective, or pronoun. Templates are used to fulfill the grammaticality property of a poem.

Firstly, we applied an Indonesian part-of-speech tagger on a corpus consisting of 213 poems written by famous Indonesian poets. Template extraction is then performed by removing words that have specific part-of-speech tags, i.e. nouns, verbs, adjectives, and pronouns. The positions of these removed words become slots to be filled later. A slot is also associated with a part-of-speech tag indicating what words may fill the slot. For example, consider the following sentence:

Aku mencintai kamu dengan sepenuh hati
I love you with full heart
I love you with all of my heart.

Each word is initially tagged with its part-of-speech. Subsequently, we remove all words tagged as <PR> (pronoun) and <NN> (noun) to obtain the following template:

<PR> mencintai <PR> dengan sepenuh <NN>
<PR> love <PR> with full <NN>
? love ? with all of (my/your/their) ?.

After extracting such templates, the feasibility and appropriateness of a template is evaluated by considering the semantic specificity embedded in the template. This consideration is important to prevent providing too much context to the system, and to avoid the risk of plagiarism against an existing line of poetry. Furthermore, with this evaluation we can determine the limits of human intervention concerning the poetic knowledge provided to the system. To illustrate, consider the following two templates (note, VBI indicates an intransitive verb):

ada yang <VBI>, ada yang <VBI>
some that <VBI>, some that <VBI>
some are ?, some are ?

ya, <PR> tahu mereka masih menggunakan <NN>
yes, <PR> know they still use <NN>
yes, ? know that they still use ?

From these two examples we can see that the latter template already carries with it a fairly specific semantic message. We believe such templates should be avoided. Furthermore, the former template is much more general and does not overconstrain the semantics. Such are the desired templates for our knowledge base. Using this consideration, we manually identified 22 templates to be used in our experiments. Theoretically it is possible to automate this process by computing the ratio of open class words remaining in the template, as opposed to function words, or closed class words.

The selected templates, along with illustrative English translations, are presented in Table 1. Note that due to grammatical differences, the translations may not be well-formed, but they are intended to illustrate the level of generality of the templates. In particular, note that almost all of the canned text contained within the templates consist of function words.

Additionally, other information that must be provided along with the template is the number of lexical slots available and the number of syllables that currently exist in the canned text of the template. This information is required for the selection process, such as to count the number of syllables and keywords. Figure *1* provides an example of how

```
TEMPLATE: SYLLABLE COUNT, SLOT COUNT
[<nn>,dan,<nn>,bisa,dibawa,<vbi>]: 6, 3
[<pr>,<vbi>,<pr>,<vbi>]: 0, 4
```

Figure 1. Two examples of templates

templates are represented in our system. It shows two templates (#11 and #5 from Table 1). The first template contains 6 syllables within its canned text ("**dan**", "**bi**", "**sa**", "**di**", "**ba**", "**wa**"), and has 3 lexical slots (2 nouns and an intransitive verb). The second template has 0 syllables within its canned text and has 4 lexical slots (2 pronouns and 2 intransitive verbs).

Slot fillers

Slot fillers are simply words used to fill the slots contained in the template. They must also be associated with a part-of-speech tag and other information that is needed in the selection process. Slot fillers can be divided into two types, keywords and poetic words.

Keywords are slot fillers that will determine the theme of the constructed poem. These words are expected to fulfill a sense of meaningfulness in the poem so that readers of the poem will capture some message that is being conveyed.

At the beginning of the poetry generation process, we crawl popular Indonesian news websites such as kompas.com and detik.com. This is motivated by FullFACE, which crawls the Guardian news website to determine the theme of the poem. An article is selected based on a given criteria, such as most recent, most commented on, or most read. After selecting an article, keyword extraction is done to obtain the keywords. Keyword extraction is done using simple unigram statistics, with stopword removal.

Templates manually selected to be used	Illustrative translations of the templates
1. <PR>	1. <PR>
2. <PR> <VBI>	2. <PR> <VBI>
3. <PR> <VBI> <RB>	3. <PR> <VBI> <RB>
4. <PR> <VBT> <PR>	4. <PR> <VBT> <PR>
5. <PR> <VBI> <PR> <VBI>	5. <PR> <VBI> <PR> <VBI>
6. dari <NN> ke <NN>	6. from <NN> to <NN>
7. adalah <ADJ> <NN>	7. there is <ADJ> <NN>
8. tapi <PR> <VBI>	8. but <PR> <VBI>
9. <PR> dan <PR> <VBI>	9. <PR> and <PR> <VBI>
10. <PR> ini hanyalah <NN>	10. <PR> is just <NN>
11. <NN> dan <NN> bisa dibawa <VBI>	11. <NN> and <NN> can be brought <VBI>
12. <PR> <VBT> <NN> bersama <PR>	12. <PR> <VBT> <NN> with <PR>
13. <VBT> <PR> adalah <ADJ> untuk <PR>	13. <VBT> <PR> is <ADJ> for <PR>
14. dengan penuh <ADJ> dalam <NN>	14. with full <ADJ> in <NN>
15. tak ada lagi <ADJ> dan <NN>	15. no more <ADJ> and <NN>
16. adakah <NN> padaku atau <NN>	16. is there <NN> with me or <NN>
17. ada yang <VBI> ada yang <VBI>	17. some are <VBI> some are <VBI>
18. mengapa <NN> <VBI>	18. why <NN> <VBI>
19. oh <PR> begitu <ADJ>	19. oh <PR> is so <ADJ>
20. terlalu <ADJ> bagi <PR>	20. too <ADJ> for <PR>
21. <NN> menjadi <NN>	21. <NN> becomes <NN>
22. apa itu <NN	22. what is <NN>

Table 1. List of templates along with illustrative translations

```
WORD: POS, PRONOUNCE, SYLL.COUNT, FLAG
senja: nn, [s, eu, n, j, aa], 2, keyword
```

Figure 2. Example of keyword representation

```
WORD: POS, PRONOUNCE, SYLL.COUNT, FLAG
kalbu: nn, [k, aa, l, b, oo], 2, filler
```

Figure 3. Example of poetic word representation

Words that have the most frequency of occurrence will be the keywords candidate. An expanded collection of keywords is then constructed by identifying words that frequently occur together with the extracted words using the Wortschatz-Leipzig Corpora Collection (Quasthoff et al., 2006).

Other information that should be associated with each keyword is its pronunciation and syllable count. This information is used for the selection process, such as for the computation of rhyme and the number of syllables in a line. Figure 2 shows an example of how keywords are represented in our system. In this example, the keyword, senja, has a part-of-speech value of NN (noun), pronunciation (s, eu, n, j, aa), 2 syllables ("sen" and "ja"), and a "keyword" flag that indicates that senja is one of the keywords of the article.

For the experiments that we conducted, we selected 3 news articles and extracted a total of 247 keywords: 88 from the 1st article, 72 from the 2nd article, and 87 from the 3rd article.

Poetic words are obtained from the same corpus of poetry used for template extraction. They are designed to help the generated poem satisfy the property of poeticness. Unlike other constraints that are more focused on the structure, this property is more focused on the selection of words to add to the aesthetics of the poem.

The frequency of appearance of every word in the existing corpus is computed and stopwords are removed. The fifty words that most frequently appear in the corpus are selected. Finally, we apply an Indonesian POS Tagger to obtain their part-of-speech tags. Poetic words tend to convey a more general concept as opposed to the specific keywords based on news article. Furthermore, they tend to be more archaic in nature.. The technical representation of poetic words is similar to how keywords are represented, as they must also be associated with pronunciation, and number of syllables. Figure 3 shows an example of how poetic words are represented in our system. The poetic word kalbu has a part-of-speech value of NN (noun), pronunciation (k, aa, l, b, oo), 2 syllables ("kal" and "bu") and a "filler" flag that indicates that the word kalbu is a poetic word.

Constraint Satisfaction Poetry Generation

Our system adapts the approach proposed by Colton et al. (2012). The system creates poetry from the collection of templates combined with a particular set of words. The result of combining templates with keywords and poetic words will be the lines that will be collated to construct the poem. Overall, the system is implemented as three stages: retrieval, combination, and selection.

It differs from Full-FACE in the following ways. Firstly, *Pemuisi* is a much more knowledge-poor system, as there are far fewer lexical resources available for Indonesian as there are for English, in particular the Jigsaw Bard resource that appears to provide a major contribution to the poeticness and coherence to the poems generated by Full-FACE. Secondly, following Toivanen et al. (2013) (and to a lesser degree, Manurung (2003)), it explicitly treats the generation process as a constraint satisfaction problem, which affords a declarative formulation of the generation process, and the use of efficient off the shelf constraint solvers. Currently, *Pemuisi* is implemented as a logic program in Prolog. All lexical resources are encoded as factual assertions in the Prolog database, and the poetic constraints are implemented as clauses with subgoals that must be satisfied. Lastly, *Pemuisi* does not attempt the handing over of high level control that is implemented in Full-FACE, which is equipped with various definitions of aesthetics.

Retrieval

During this stage, a simple retrieval is performed by taking the relevant resources previously described from the knowledge base. Given an input news article, the system will populate the Prolog database with all relevant keywords, poetic words, and appropriate templates. The retrieval process can be set to randomly reorder the sequence of factual assertions, so that the systematic Prolog depth first search can yield novel results on repeated runs. Figure 4 shows an example of the output of this stage.

Combination

After collecting all the necessary resources, the system can start building the poem from the simplest unit, namely the poetry line. The combination process produces a poetry line through merging of a template with slot filler(s) by obeying certain rules. Each slot in the template must be filled with precisely one slot filler. A slot can only be filled with a slot filler with a corresponding part-of-speech tag. For example, a slot with a POS tag of NN (noun) can only be filled by a keyword or poetic word with a POS tag of NN. The system

```
TEMPLATE:
[<nn>,dan,<nn>,bisa,dibawa,<vbi>]: 6, 3
[<pr>,<vbi>,<pr>,<vbi>]: 0, 4

SLOT FILLER
aku:pr,[aa,k,oo],2,filler
kau:pr,[k,aa,oo],2,filler
senja:nn,[s,eu,n,j,aa],2,keyword
kalbu:nn,[k,aa,l,b,oo],2,filler
bayang:nn,[b,aa,y,aa,ng],2,keyword
pergi:vbi,[p,eu,r,g,ee],2,filler
kembali:vbi,[k,eu,m,b,aa,l,ee],3,filler
menunggu:vbi,[m,eu,n,oo,ng,g,oo],3,keyword
```

Figure 4. Example output of retrieval stage

will exhaustively consider all possible valid combinations of templates and slot fillers.

Consider the following example. Suppose that the resources obtained from the retrieval stage are as in Figure 4, which means the system must now combine 2 templates with 8 slot fillers consisting of: 2 <PR> slot fillers; 3 <NN> slot fillers, 2 of which are keywords; and 3 <VBI> slot fillers, 1 of which is a keyword.

Going by the previous explanation of how the process is done then all slot combinations are instantiated with the corresponding slot fillers to form the poem lines. Based on simple observation, it can be calculated that the number of combinations of lines of poetry can be generated from the collection of the above resources. 63 valid combinations of poetry lines can be obtained from the combination of templates and corresponding slot fillers.

Selection

After the combination stage, the system now has a large collection of poem lines that are ready to be built into larger units, i.e. the poem itself. This stage combines the lines that have been previously obtained as results of the combination. The resulting poem must satisfy the elements of poetry, such as the number of syllables, rhyme, rhythm, and number of lines. Such poetic elements are defined as constraints. Constraints that will be used include:

1) **Number of lines**: a constraint that states the number of poetry lines. As explained in the combination stage, the definition used for a single line is a result of a combination of a template with one or more slot filler.
2) **Rhyme**: a constraint that states the rules of rhyme in between lines of the poetry.
3) **Number of words**: a constraint that states the number of words contained in a single line of poetry. The number of words can be specified differently for each row.
4) **Number of syllables**: a constraint that states the number of syllables contained in a single line of poetry. Number of syllables can be specified differently for each row.
5) **The number of keywords relative to the number of slots**: a constraint that states the number of keywords relative to the number of slots contained in the whole poetry. In order to be more intuitive and easier, this constraint is expressed as a percentage. It can be used to control how the content of the poem focuses on a topic.

The above set of constraints must be met when choosing combinations of line results from the previous stage. This is an important point of the concept of constraint satisfaction approach as also seen in Toivanen et al. (2013).

From the previous example results obtained 63 lines of poetry that can be built into a combination of poetry. For instance, assume the following constraints:

1) The poem consists of 2 lines.
2) Line 1 and 2 share the same end-of-line rhyme.
3) Line 1 consists of 6 words with a total of 12 syllables.
4) Line 2 consists of 4 words with a total of 10 syllables.
5) 40% of all slots must be filled with content keywords.

If we only look at the first constraint, it can be calculated that there are 63^2 poems that could be generated. But the more we continue to meet the subsequent constraints, the less the combinations of lines of poetry that are able to meet all the constraints.

There are at least three cases that may occur after the selection process is done: (i) the system does not produce a single poem at all, (ii) it produces exactly a single poem, and (iii) it generates more than one poem.

If no poem is produced, it means there is no combination that successfully meets the constraints that have been defined. In this case, the constraints will be gradually relaxed and the selection stage repeated until eventually a poem can be produced. In loosening constraints, the constraint that has the lowest precedence is first chosen to be relaxed. This process is repeated until the system is capable of producing a poem that satisfies the remaining constraints.

If the system is able to produce one or more poems, it will randomly select one as its eventual output. Another alternative is to provide all the poetry as the output.

Pemuisi is currently equipped with six poem structures, i.e. sets of constraints, to be used during the experiments. The purpose of the provision of six alternative structures is for the poetry generated by the system to be more diverse.

An Illustrative Example

In this section we provide an example of the output of *Pemuisi.* It was run to construct a poem based on an article from an Indonesian news portal, kompas.com, about Sir Alex Ferguson's retirement in 2013 as Manchester United head coach. We situated *Pemuisi* to compose a poem with full constraint parameter and then randomly took 3 stanzas. Figure 5 shows the poem made by *Pemuisi*.

The corresponding constraints which became the reference for *Pemuisi* while generating this poem can be seen in Figure 6. While comparing Figure 5 and Figure 6, we can see that the set of constraints were all satisfied by the resulting poem.

Experiments and Evaluation

We conducted experiments using several constraint configurations through an online web-based questionnaire to see the respondents' opinions about the poetry generated by the system. Information about the experiment was distributed through various mailing lists and social media channels (e.g. Facebook, Twitter), targeting native Indonesian speakers including public groups, academic communities, and poetry appreciation communities in order to provide a more balanced and valid distribution of respondents, ranging from a layman's appreciation of poetry to communities that specifically discuss poetry appreciation. At the end of the data collection, we managed to obtain 180 respondents.

fergie pergi	*fergie is gone*
ferguson pensiun, ferguson berhenti	*ferguson retired, ferguson stopped*
adakah masa padaku atau juri	*is there time with me or jury*
fergie berhenti	*fergie stopped*
fergie pensiun sendirian	*fergie retired alone*
dengan penuh merah dalam perjuangan	*with full red in struggle*
tak ada lagi akrab dan perjalanan	*no more friendship and trips*
fergie pensiun sendirian	*fergie retired alone*
dengan penuh biru dalam kesedihan	*with full blue in sadness*
tak ada lagi akrab dan pertandingan	*no more friendship and matches*
ferguson, ini hanyalah kompetisi	*ferguson, this is just a competition*
usia dan keputusan bisa dibawa pensiun	*age and decisions can be brought in retirement*
fergie, ini hanyalah tradisi	*fergie, this is just a tradition*
pemain dan manajemen bisa dibawa pensiun	*players and management can be brought in retirement*

Figure 5. Illustrative output of *Pemuisi*

Constraint configurations

There are three constraint configurations that were applied. In the first configuration, the full set of poetic constraints are applied, and a ratio of 50% of the open slots must be filled by content keywords. The second configuration is similar to the first, but in this case all the open slots must be filled by content keywords. Finally, the loose constraint configuration is one where the system is more or less left unguided to generate poems, with the only constraints being the use of templates, part of speech tags, and the number of lines to be generated, i.e. poetic features such as syllable counts, rhymes, and content keyword ratios are ignored. Obviously, respondents were not made aware of the distinction of these three configurations, and were simpy asked to rate the perceived quality of the generated poems regardless of the configuration of the generator.

Stanza 1
Number of lines: 4
Line 1 – number of words: 2; number of syllables: 4
Line 2 – number of words: 4; number of syllables: 12
Line 3 – number of words: 5; number of syllables: 12
Line 4 – number of words: 2; number of syllables: 5
Line 1, 2, 3, and 4 rhyme with each other
Keywords composition: 100%

Stanza 2
Number of lines: 6
Line 1 – number of words: 3; number of syllables: 10
Line 2 – number of words: 5; number of syllables: 12
Line 3 – number of words: 6; number of syllables: 12
Line 4 – number of words: 3; number of syllables: 12
Line 5 – number of words: 5; number of syllables: 12
Line 6 – number of words: 6; number of syllables: 12
Line 1, 2, 3, 4, 5, and 6 rhyme with each other
Keywords composition: 100%

Stanza 3
Number of lines: 4
Line 1 – number of words: 4; number of syllables: 12
Line 2 – number of words: 6; number of syllables: 16
Line 3 – number of words: 4; number of syllables: 10
Line 4 – number of words: 6; number of syllables: 16
Line 1 and 3 rhyme with each other
Line 2 and 4 rhyme with each other
Keywords composition: 100%

Figure 6. Constraint configuration used for poem in Figure 5

Turing Test

Before conducting the main experiment to see how respondents' evaluated the computer generated poems in terms of various aspects, we first conducted a simple Turing Test-like experiment to to determine how the system is able to imitate human behavior, in this case writing poetry. For this experiment, we selected snippets from four poems created by famous Indonesian poets (such as Chairil Anwar, Sutardji Calzoum Bachri, and WS Rendra), four poems generated by the system with the full constraint configuration, and four poems generated by system with the loose constraint configuration.

For this Turing Test, the system only used poetic words as slot fillers so that the poetry does not specifically discuss a particular topic. These poems were randomized in the questionnaire and respondents were asked to annotate each poem by guessing whether the poem was written by a human or system. Figure 7 shows some poem examples for the Turing Test section.

The questionnaire results for the Turing Test are shown in Table 2. 74% of the respondents correctly identified human-authored poems, but 26% of the human-authored poem judgments were erroneous (i.e. deemed to be machine-authored). As for the poems generated with the full set of constraints, 57% of the judgments were erroneous, i.e. they were deemed to be human-authored, and for the poems generated with the loose constraints, in only 35% of the cases did respondents falsely identify them as human-authored.

Human authored (*Hilang* (Lost), by Sutardji Calzoum Bachri)	
batu kehilangan diam	*A stone loses silence*
jam kehilangan waktu	*A clock loses time*
pisau kehilangan tikam	*A knife loses stab*
mulut kehilangan lagu	*A mouth loses song*
Full constraint	
tak ada lagi pilu dan rindu	*no more pain and yearning*
dari rindu ke mentari	*from yearning to the sun*
ada yang terdiam	*some lay silent*
ada yang menunggu	*some lay in waiting*
Loose constraint	
cinta kau adalah sakit untuk kau	*your love is pain for you*
aku melayang, aku melayang	*I fly, I fly*
cinta kau adalah sakit untuk kau	*your love is pain for you*

Figure 7. Poem examples for Turing Test

Main experiment

For the main experiment, the three constraint configurations were each applied to three different news articles, resulting in 9 different poems being assessed. The poems were randomly obtained from the system output.

In this section of the experiment, we aim to analyze how the poems generated by the system under different configurations were appraised by respondents. The questionnaire randomly presents one of the three chosen news articles along with the three poems produced from that article under the previously discussed constraint configurations. Each poem is the result of concatenating three stanzas that were generated and selected randomly. Respondents were asked to give an assessment of the poems based on the following criteria:

1) **Structure**: a criterion to evaluate the overall structure of the poem, i.e. whether or not it fulfilled the respondent's subjective expectations of what constitutes a poem.

	Human authored	**Full constraints**	**Loose constraints**
Human	74%	57%	35%
Machine	26%	43%	65%

Table 2. Results for Turing Test experiment

2) **Diction**: a criterion to evaluate the choice of words used in the poetry generated.
3) **Grammar**: a criterion to evaluate how well the grammar was in the poem.
4) **Unity**: a criterion to evaluate the unity between the form and content of poetry produced.
5) **Message/theme**: a criterion to evaluate the suitability of the poetry content with the reference article.
6) **Expressiveness**: a criterion to evaluate the level of expression of the resulting poem.

An overview of the data analysis results of the questionnaire can be seen in Figure 8. The blue bar represents 50% keywords-full constraint, the red bar represents 100% keywords-full constraint, and the green bar represents loose constraint.

Every respondent's assessment is transformed to number scale with range of 0-3 then accumulated for the six criteria that have been mentioned previously. From the overview we can see that in general 50% keywords-full constraint and 100% keywords-full constraint parameter give better performance than loose constraint parameter in every criterion.

As can be seen from Figure 8, poems made with 50% keywords-full constraint and 100% keywords-full constraint have a better structure than loose constraint. The structure is evaluated from the number of lines, number of syllables, and rhyme in the poem. We can predict this result as the full constraint configuration is meant to give a strict rule for the system when composing poems that the loose constraint

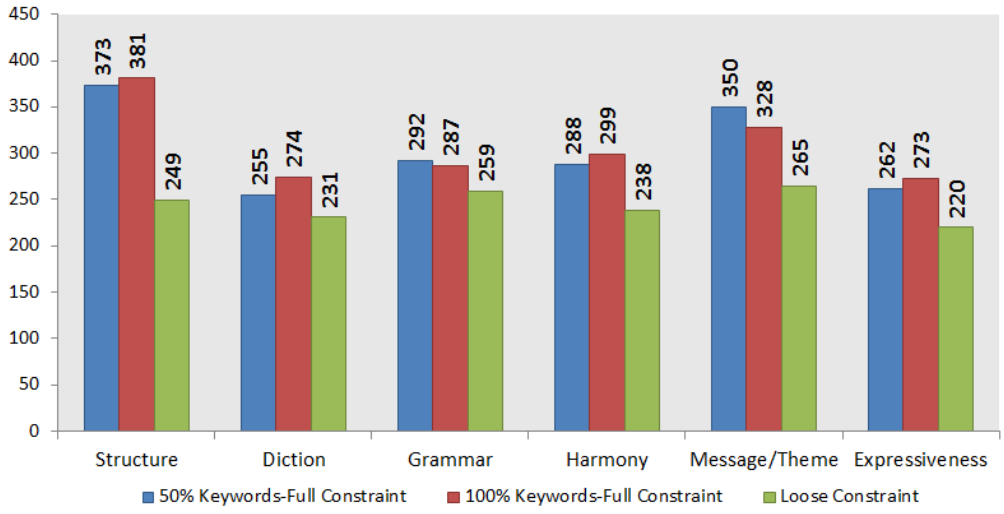


Figure 8. Overview of main experiment results

configuration does not have to obey. This phenomenon was also seen in unity and message aspect. Poems made with 50% keywords-full constraint and 100% keywords-full constraint seem to have a message and stay in specific theme/topic rather than loose constraint. The system is expected to achieve a good performance for discussing a specific theme given the way that the keywords are selected. The keyword ratio constrains the poems to remain on topic while the loose constraint configuration does not. However, it is important to remember that *Pemuisi* is not deliberately conveying a particular semantic message as it is simply constructing lines of poetry by randomly filling slots (given constraints). Thus, we claim that *Pemuisi* composes poems that can be said to be related to the article rather than faithful to the article. Tables 3 and 4 show the detail between topic and message aspect retrieved from the questionnaire response. 50%-FC stands for 50% keywords-full constraint, 100%-FC stands for 100% keywords-full constraint, and LC stands for loose constraint.

	50%-FC		**100%-FC**		**LC**	
	Topic	**Msg**	**Topic**	**Msg**	**Topic**	**Mes**
TA	29%	10%	11%	6%	5%	2%
A	59%	61%	76%	61%	54%	49%
D	10%	25%	11%	31%	34%	42%
TD	1%	4%	2%	3%	7%	8%

TA: Totally Agree; A: Agree; D: Disagree; TD: Totally Disagree

Table 3. The existence of topic and message

	50%-FC		**100%-FC**		**LC**	
	Topic	**Msg**	**Topic**	**Msg**	**Topic**	**Msg**
TA	24%	4%	11%	7%	5%	2%
A	64%	66%	68%	61%	46%	41%
D	12%	28%	20%	31%	42%	49%
TD	1%	2%	1%	1%	7%	8%

TA: Totally Agree; A: Agree; D: Disagree; TD: Totally Disagree

Table 4. The relation of topic and message with the article

The unity between the form and content is better in 50% keywords-full constraint and 100% keywords-full constraint than loose constraint. This aspect shows us about the unity of poem structure and content.

50% keywords-full constraint and 100% keywords-full constraint have a slight lead in expressiveness aspect. This could be due to the composition between poetic words and keywords that is regulated by the keywords ratio. While keeping the poem to stay on topic, we allow the system to also be expressive by using poetic words. Finally, an almost tie result is shown in diction and grammar aspect with 50% keywords-full constraint and 100% keywords-full constraint, with both yielding a slightly better result than loose constraint. We can infer that for every parameter we use the same templates set that already holds for grammaticality property.

Pemuisi: Up-to-date Poem Feed

We have developed a web application as a showcase to publish *Pemuisi* poems at http://budaya.cs.ui.ac.id/pemuisi. The core generation system runs as a background process of the site and is scheduled at noon everyday to crawl various news portals. In order to make *Pemuisi* up-to-date with the world situation, *Pemuisi* will find a recent article published by looking into the news portal RSS feed. The entire preprocessing work is automated.

Pemuisi composes a poem consisting of 3-4 stanzas about the chosen news article. As *Pemuisi* would produce all poem combinations which satisfy the set of given constraint, we demand a fast processing and relevant poem. We provide seven sets of constraints which represent various kinds of Indonesian traditional poem form structure. We also provide 22 templates and 50 poetic words as static language resources. These constraints and language resources can be added anytime later. The *Pemuisi* web application also randomly shuffles the order of all the language resources and set of constraints before generation commences in order to raise the diversity level of the output.

The poem produced by *Pemuisi* is then published to the site page. The first line of the poem is also tweeted by the *Pemuisi* Twitter account (@pemuisi) along with the website page link. In the site page connected with Twitter and Facebook, viewers can comment and share their thoughts about the poem to social media.

Conclusions and Future Work

We have developed an automatic poetry generation system that is capable of automatically generating poems in Indonesian based on specific context restrictions defined by existing constraints and reference news articles.

The system combines the general architecture of the Full-FACE system introduced in Colton et al. (2012), particularly the aspect that generated poems are based on current news articles, with the explicit treatment of the generation process as a constraint satisfaction problem as in Toivanen et al. (2013) (and to a lesser degree, Manurung (2003)), which affords a declarative formulation of the generation process, and the use of efficient off the shelf constraint solvers (although in our current system we use Prolog, we plan to use purpose-built constraint solvers such as ECLiPSe[1]).

The main contribution of this work, aside from this combined approach, and the adaptation to Indonesian, is the user evaluation that was conducted, as both Colton et al. (2012) and Toivanen et al. (2013) present no user evaluation. Lastly, *Pemuisi* is in effect a much more knowledge-poor system than Full-FACE, as there are far fewer lexical resources available for Indonesian as there are for English, in particular the Jigsaw Bard resource that appears to provide

[1] http://eclipseclp.org

a major contribution to the poeticness and coherence to the generated poems.

From the experimental results, it was found that when all the implemented constraints are applied the system is able to produce poetry that is deemed more similar to human-authored poetry rather than the poetry generated under the loosely-constrained configuration. They were also deemed to have better structure, more focus on a topic and conveyed the message from the reference article better.

Many aspects from the system are still rudimentary, and there are still many opportunities to improve the system, such as expanding the types of constraints that can be handled, developing a better interface for the user, and improving the language resources. A careful qualitative evaluation from poets and other poetry experts would be valuable in order to gain feedback about the output of the system. With the developed web application, viewers can leave comments about the generated poem, thus this provides a channel for collecting information for a deep analysis on human perception about the generated poems.

References

Colton, S., J. Goodwin, and T. Veale. 2012. Full-FACE Poetry Generation. In *Proceedings of the 3rd International Conference on Computational Creativity, 2012*. Dublin, Ireland.

Gervás, P. 2002. Exploring quantitative evaluations of the creativity of automatic poets. In *Proceedings of the 2nd. Workshop on Creative Systems, Approaches to Creativity in Artificial Intelligence and Cognitive Science, 15th European Conference on Artificial Intelligence (ECAI).* Lyon, France.

Manurung, H. 2003. *An evolutionary algorithm approach to poetry generation*. PhD. Dissertation, University of Edinburgh, Edinburgh, United Kingdom.

Quasthoff, U., M. Richter, and C. Biemann. 2006. Corpus Portal for Search in Monolingual Corpora. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006)*, Genoa, pp. 1799-1802

Toivanen, J. M., M. Järvisalo, and H. Toivonen. 2013. Harnessing Constraint Programming for Poetry Generation. In *Proceedings of the 4th International Conference on Computational Creativity 2013*. Sydney, Australia.

Musical Motif Discovery in Non-musical Media

Daniel Johnson and **Dan Ventura**
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
daniel.johnson@byu.edu
ventura@cs.byu.edu

Abstract

Many music composition algorithms attempt to compose music in a particular style. The resulting music is often impressive and indistinguishable from the style of the training data, but it tends to lack significant innovation. In an effort to increase innovation in the selection of pitches and rhythms, we present a system that discovers musical motifs by coupling machine learning techniques with an inspirational component. Unlike many generative models, the inspirational component allows the composition process to originate outside of what is learned from the training data. Candidate motifs are extracted from non-musical media such as images and audio. Machine learning algorithms select and return the motifs that most resemble the training data. This process is validated by running it on actual music scores and testing how closely the discovered motifs match the expected motifs. We examine the information content of the discovered motifs by comparing the entropy of the discovered motifs, candidate motifs, and training data. We measure innovation by comparing the probability of the training data and the probability of the discovered motifs given the model.

Introduction

Computational music composition is still in its infancy, and while numerous achievements have already been made, many humans still compose better than computers. Current computational approaches tend to favor one of two compositional goals. The first goal is to produce music that mimics the style of the training data. Approaches with this goal tend to 1) learn a model from a set of training examples and 2) probabilistically generate new music based on the learned model. These approaches effectively produce artefacts that mimic classical music literature, but little thought is directed toward expansion and transformation of the music domain. For example, David Cope (1996) and Dubnov et al. (2003) seek to mimic the style of other composers in their systems. The second goal is to produce music that is radically innovative. These approaches utilize devices such as genetic algorithms (Burton and Vladimirova 1999; Biles 1994) and swarms (Blackwell 2003). While these approaches can theoretically expand the music domain, they often have little grounding in a training data set, and their output often receives little acclaim from either music scholars or average listeners. A large portion of work serves one of these two goals, but not both.

While many computational compositions lack either innovation or grounding, great human composers from the period of common practice and the early 20th century composed with both goals in mind. For instance, Beethoven's music pushes classical boundaries into the beginnings of romanticism. The operas of Wagner bridge the gap between tonality and atonality. Schoenberg's twelve-tone music pushes atonality to a theoretical maximum. Great composers of this period produce highly creative work by extending the boundaries of the musical domain without completely abandoning the common ground of music literature. We must note that some contemporary composers strive to completely reject musico-historical precedent. While this is an admirable cause, we do not share this endeavor. Instead, we seek to compose music that innovates and extends the music of the period of common practice and the early 20th century.

Where do great composers seek inspiration in order to expand these boundaries in a musical way? They find inspiration from many non-musical realms such as nature, religion, relationships, art, and literature. Olivier Messiaen's compositions mimic birdsong and have roots in theology (Bruhn 1997). Claude Debussy is inspired by nature, which becomes apparent by scanning the titles of his pieces, such as *La mer* [The Ocean], *Jardins sous la pluie* [Gardens in the Rain], and *Les parfums de la nuit* [The Scents of the Night]. Debussy's *Prélude á l'aprés-midi d'un faune* [Prelude to the Afternoon of a Faun] is a direct response to Stéphane Mallarmé's poem, *L'aprés-midi d'un faune* [The Afternoon of a Faun]. Franz Liszt's programme music attempts to tell a story that usually has little to do with music. Many pop musicians are clearly inspired by relationships and social interactions. While it is essential for a composer to be familiar with music literature, it is apparent that inspiration extends to non-musical sources.

We present a computational composition method that serves both of the aforementioned goals rather than only one of them. This method couples machine learning (ML) techniques with an inspirational component, modifying and extending an algorithm introduced by Smith et al. (2012). The ML component maintains grounding in music literature and harnesses innovation by employing the strengths of genera-

tive models. It embraces the compositional approach found in the period of common practice and the early 20th century. The inspirational component introduces non-musical ideas and enables innovation beyond the musical training data. The combination of the ML component and the inspirational component allows us to serve both compositional goals.

Media Inspiration

Just as humans often rely on inspiration for their creative work, our motif discovery system relies on non-musical audio files for inspiration. Non-musical audio is a natural starting place for musical inspiration because audio and music both exist in the sound medium. We also generalize one step further by allowing our system to be inspired by other forms of media, specifically images. A human might look at a painting, understand its meaning, and compose a piece of music based on the way he feels about it. He might also feel inspired to compose a piece of music shortly after attending a speech, listening to a bird chirp, watching a movie, or reading poetry. Since computer technology has not yet matched the full capacity of humans in understanding events in the world, we begin with unsophisticated means for extracting musical inspiration from media (our precise methods are described in a later section).

Musical Motifs

We focus on the composition of motifs, the atomic level of musical structure. We use White's definition of motif, which is "the smallest structural unit possessing thematic identity" (1976). There are two reasons for focusing on the motif. First, it is the simplest element for modeling musical structure, and we agree with Cardoso et al. (2009) that success is more likely to be achieved when we start small. Second, it is a natural starting place to achieve global structure based on variations and manipulations of the same motif throughout a composition.

Since it is beyond the scope of this research to build a full composition system, we present a motif composer that performs the first compositional step. The motif composer trains an ML model with music files, it discovers candidate motifs from non-musical media, and it returns the motifs that are the most probable according to the ML model built from the training music files. It will be left to future work to combine these motifs into a full composition.

Related Work

A variety of machine learning models have been applied to music composition. Many of these models successfully reproduce credible music in a genre, while others produce music that is radically innovative. Since the innovative component of our algorithm is vastly different than the innovative components of other algorithms, we only review the composition algorithms that effectively mimic musical style.

Cope extracts musical signatures, or common patterns, from the works of a composer. These signatures are recombined into a new composition in the same style (1996). This process effectively replicates the styles of composers, but its novelty is limited to the recombination of already existing signatures. Aside from Cope's work, the remaining relevant literature is divisible into two categories: Markov models and neural networks.

Markov Models

Markov models are perhaps the most obvious choice for representing and generating sequential data such as melodies. The Markov assumption allows for inference and learning to be performed simply and quickly on large data sets. However, first-order Markov processes do not store enough information to represent longer musical contexts, while high-order Markov processes require intractable space and time.

This issue necessitates a variable order Markov model (VMM) in which variable length contexts are stored. Dubnov et al. (2003) implement a VMM for modeling music using a prediction suffix tree (PST). A longer context is only stored in the PST when 1) it appears frequently in the data and 2) it differs by a significant factor from similar shorter contexts. This allows the model to remain tractable without losing significant longer contextual dependencies. Begleiter et al. (2004) compare results for several variable order Markov models (VMMs), including the PST. Their experiments show that Context Tree Weighting (CTW) minimizes log-loss on music prediction tasks better than the PST (and all other VMMs in this experiment). Spiliopoulou and Storkey (2012) propose the Variable-gram Topic model for modeling melodies, which employs a Dirichlet-VMM and is also shown to improve upon other VMMs.

Variable order Markov models are not the only extensions explored. Lavrenko and Pickens (2003) apply Markov random fields to polyphonic music. In these models, next-note prediction accuracies improve when compared to a traditional high-order Markov chain. Weiland et al. (2005) apply hierarchical hidden Markov models (HHMMs) in order to capture long-term dependencies in music. HHMMs are used to model both pitch and rhythm separately.

Markov models generate impressive results, but the emissions rely entirely on the training data and a stochastic component. This results in a probabilistic walk through the training space without introducing any actual novelty or inspiration beyond perturbation of the training data.

Neural Networks

Recurrent neural networks (RNNs) are also effective for learning musical structure. However, similar to Markov models, RNNs still struggle to represent long-term dependencies and global structure due to the vanishing gradient problem (Hochreiter et al. 2001). Eck and Schmidhuber (2008; 2002) address the vanishing gradient problem for music composition by applying long short-term memory (LSTM). Chords and melodies are learned using this approach, and realistic jazz music is produced. Smith and Garnett (2012) explore different approaches for modeling long-term structure using hierarchical adaptive resonance theory neural networks. Using three hierarchical levels, they demonstrate success in capturing medium-level musical structures.

Like Markov models, neural networks can effectively capture both long-term and short-term statistical regularities in music. This allows for music composition in any genre given sufficient training data. However, few (if any) researchers have incorporated inspiration in neural network composition prior to Smith et al. (2012). Thus, we propose a novel technique to address this deficiency. Traditional ML methods can be coupled with sources of inspiration in order to discover novel motifs that originate outside of the training space. ML models can judge the quality of potential motifs according to learned rules.

Methodology

An ML algorithm is employed to learn a model from a set of music themes. Pitch detection is performed on a non-musical audio file, and a list of candidate motifs is saved. For our purposes, semantic content in the audio files is ignored. The candidate motifs that are most probable according to the ML model are returned. This process is tested using different ML model classes over various audio input files. A high-level system pipeline is shown graphically in Figure 1.

In order to generalize the concept of motif discovery from non-musical media, we also extend our algorithm to accept images as inputs. With images, we replace pitch detection with edge detection, and we iterate using a spiral pattern through the image in order to collect notes. This process is further explained in its own subsection.

The training data for this experiment are 9824 monophonic MIDI themes retrieved from The Electronic Dictionary of Musical Themes.[1] The training data consists of themes rather than motifs. We make this decision due to the absence of a good motif data set. An assumption is made that a motif follows the same general rules of a theme, except it is shorter. In order to better learn statistical regularities from the data set, themes are discarded if they contain at least one pitch interval greater than a major ninth. This results in a final training data set with 9383 musical themes. Themes and motifs are represented using the Phrase class from the jMusic library. We also utilize core functionality from jMusic for reading, writing, and manipulating musical structures.[2]

Machine Learning Models

A total of six ML model classes are tested. These include four VMMs, an LSTM RNN, and an HMM. These model classes are chosen because they are general, they represent a variety of approaches, and their performance on music data has already been shown to be successful. The four VMMs include Prediction by Partial Match, Context Tree Weighting, Probabilistic Suffix Trees, and an improved Lempel-Ziv algorithm named LZ-MS. Begleiter et al. provide an implementation for each of these VMMs,[3] an LSTM found on Github is used,[4] and the HMM implementation is found in the Jahmm library.[5]

Each of the learned ML models is used on both pitches and rhythms separately. Each model contains 128 possible pitches (0-127) and 32 possible note durations (32nd note multiples up to a whole note). The set of inputs in the RNNs represents which note is played, and the set of outputs represents the next note in the sequence to be played. The RNNs train for a fixed number of iterations before halting. The HMMs are trained using the Baum-Welch algorithm for a fixed number of iterations. The VMMs are trained according to the algorithms presented by Begleiter et al. (2004).

Audio Pitch Detection

Our system accepts an audio file as input. Pitch detection is performed on the audio file using an open source command line utility called Aubio.[6] More precisely, we use the *aubionotes* Windows binary from version 0.4.0 of Aubio, *schmitt* pitch detection, *kl* onset detection, and a threshold of 0.5. Aubio combines note onset detection and pitch detection in order to output a string of notes, in which each note is comprised of a pitch and duration. The string of detected notes is processed in order to make the sequence more manageable: given a tempo of 120 beats per minute, note durations are quantized to a 32nd note value; and note pitches are restricted to MIDI note values in the range [55, 85] by adding or subtracting octaves until each pitch is in range.

Image Edge Detection

Images are also used as inspirational inputs for the motif discovery system. We perform edge detection on an image using a Canny edge detector implementation,[7] which returns a new image comprised of black and white pixels. The white pixels (0 value) represent detected edges, and the black pixels (255 value) represent non-edges. We also convert the original image to a greyscale image and divide each pixel value by two, which changes the range from [0, 255] to [0, 127]. We simultaneously iterate through the edge-detected image and the greyscale image one pixel at a time using a spiral pattern starting from the outside and working its way inward. For each sequence of b contiguous black pixels (delimited by white pixels) in the edge-detected image, we create one note. The pitch of the note is the average intensity of the corresponding b pixels in the greyscale image, and the duration of the note is b 32nd notes. The pitches are restricted to MIDI note values in the range [55, 85] as they were for pitch-detected sequences. Quantization is not performed for edge-detected sequences, since all of the note durations are already multiples of 32nd notes.

Motif Discovery

After the string of notes are detected and processed, we extract candidate motifs of various sizes (see Algorithm 1). We define the minimum motif length as l_min and the maximum motif length as l_max. All contiguous motifs of length

[1] http://www.multimedialibrary.com/barlow/all_barlow.asp

[2] http://explodingart.com/jmusic

[3] http://www.cs.technion.ac.il/~ronbeg/vmm/code_index.html

[4] https://github.com/evolvingstuff/SimpleLSTM

[5] http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm/

[6] http://www.aubio.org

[7] http://www.tomgibara.com/computer-vision/canny-edge-detector

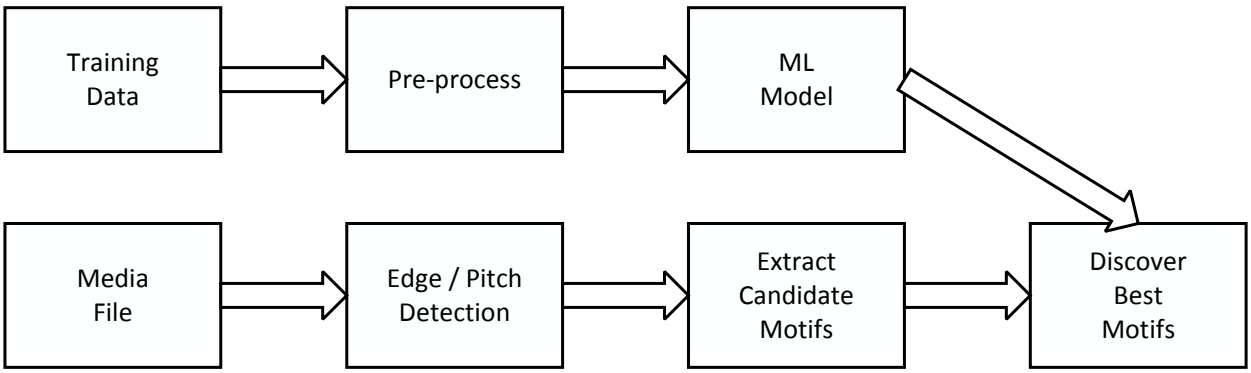


Figure 1: A high-level system pipeline for motif discovery. An ML model is trained on pre-processed music themes. Pitch detection is performed on an audio file or edge detection is performed on an image file in order to extract a sequence of notes. The sequence of notes is segmented into a set of candidate motifs, and only the most probable motifs according to the ML model are selected.

greater than or equal to l_min and less than or equal to l_max are stored. For our experiments, the variables l_min and l_max are set to 4 and 7 respectively.

After the candidate motifs are gathered, the motifs with the highest probability according to the model of the training data are selected (see Algorithm 2). The probabilities are computed in different ways according to which ML model is used. For the HMM, the probability is computed using the forward algorithm. For the VMMs, the probability is computed by multiplying all the transitional probabilities of the notes in the motif. For the RNN, the activation value of the correct output note is used to derive a pseudo-probability for each motif.

Pitches and rhythms are learned separately, weighted, and combined to form a single probability. The weightings are necessary in order to give equal consideration to both pitches and rhythms. In our system, a particular pitch is generally less likely than a particular rhythm because there are more pitches to choose from. Thus, the combined probability is defined as

$$P_{p+r}(m) = Pr(m_p){N_p}^{|m|} + Pr(m_r){N_r}^{|m|} \qquad (1)$$

where m is a motif, m_p is the motif pitch sequence, m_r is the motif rhythm sequence, N_p and N_r are constants, and $N_p > N_r$. In this paper we set $N_p = 60$ and $N_r = 4$. The resulting value is not a true probability because it can be greater than 1.0, but this is not significant because we are only interested in the relative probability of motifs. For convenience, in what follows, we will use the simpler notation $Pr(m)$ as a short hand for $P_{p+r}(m)$ as well as the conditional notation $Pr(m|M)$ as a shorthand for $P_{p+r}(m|M)$, where $P_{p+r}(m|M)$ is computed as in Eq. 1, replacing the independent probabilities with their respective conditional counterparts.

Since shorter motifs are naturally more probable than longer motifs, an additional normalization step is taken in Algorithm 2. We would like each motif length to have equal probability:

Algorithm 1 *extract_candidate_motifs*

1: **Input:** $notes$, l_min, l_max
2: *candidate_motifs* $\leftarrow \{\}$
3: **for** $l\_min \leq l \leq l\_max$ **do**
4: **for** $0 \leq i \leq |notes| - l$ **do**
5: *motif* $\leftarrow (notes_i, notes_{i+1}, ..., notes_{i+l-1})$
6: *candidate_motifs* $\leftarrow$ *candidate_motifs* $\cup$ *motif*
7: **return** *candidate_motifs*

Algorithm 2 *discover_best_motifs*

1: **Input:** $notes$, $model$, *num_motifs*, l_min, l_max
2: $C \leftarrow$ *extract_candidate_motifs*(*notes*, *l_min*, *l_max*)
3: *best_motifs* $\leftarrow \{\}$
4: **while** $|$*best_motifs*$| <$ *num_motifs* **do**
5: $m^* \leftarrow \underset{m \in C}{\text{argmax}}[norm(|m|)Pr(m|model)]$
6: *best_motifs* $\leftarrow$ *best_motifs* $\cup\, m^*$
7: **return** *best_motifs*

$$P_{equal} = \frac{1}{(l\_max - l\_min + 1)} \qquad (2)$$

Since the probability of a generative model emitting a motif of length *l* is

$$P(l) = \sum_{m \in C, |m|=l} Pr(m|model) \qquad (3)$$

we introduce a length-dependent normalization term that equalizes the probability of selecting motifs of various lengths.

$$norm(l) = \frac{P_{equal}}{P(l)} \qquad (4)$$

This normalization term is used in step 5 of Algorithm 2.

Validation and Results

We perform three stages of validation for this system. First, we compare the entropy of pitch-detected and edge-detected music sequences to comparable random sequences as a baseline sanity check to see if images and audio are better sources of inspiration than are random processes. Second, we run our motif discovery system on real music scores instead of media, and we validate the motif discovery process by comparing the discovered motifs to hand annotated themes for the piece of music. Third, we evaluate the structural value of the motifs. This is done by comparing the entropy of the discovered motifs, candidate motifs, and themes in the training set. We also measure the amount of innovation in the motifs by measuring the probability of the selected motifs against the probability of the training themes according to the learned ML model.

Preliminary Evaluation of Inspirational Sources

Although pitch detection is intended primarily for monophonic music signals, interesting results are still obtained on non-musical audio signals. Additionally, interesting musical inspiration can be obtained from image files. We performed some preliminary work on fifteen audio files and fifteen image files and found that these pitch-detected and edge-detected sequences were better inspirational sources than random processes. This evaluation was performed as a sanity check, and we did not select motifs or use machine learning at this stage. Instead, we compared the $entropy$ (see Equation 5) of pitch-detected and edge-detected sequences against comparable random sequences and found that there was more rhythm and pitch regularity in the pitch-detected and edge-detected sequences. In our data, the sample space of the random variable X is either a set of pitches or a set of rhythms, so $Pr(x_i)$ is the probability of observing a particular pitch or a rhythm.

$$H(X) = -\sum_{i=1}^{n} Pr(x_i) \log_b Pr(x_i) \tag{5}$$

More precisely, for one of these sequences we found the sequence length, the minimum pitch, maximum pitch, minimum note duration, and maximum note duration. Then we created a sequence of notes from two uniform random distributions (one for pitch and one for rhythm) with the same length, minimum pitch, maximum pitch, minimum note duration, and maximum note duration. The average pitch and rhythm entropy measures were lower for pitch-detected and edge-detected sequences. A homoscedastic, two-tailed Student's t-test on the data shows statistical significance with p-values of 1×10^{-5} for pitches from images, 1×10^{-23} for rhythms from images, and 0.0003 for rhythms from audio files. In addition, although the p-value for pitches from audio files is not statistically significant (0.175), it is still fairly low. This suggests that there is potential for interesting musical content (Wiggins, Pearce, and Müllensiefen 2009) in the pitch-detected and edge-detected sequences even though the sequences originate from non-musical sources.

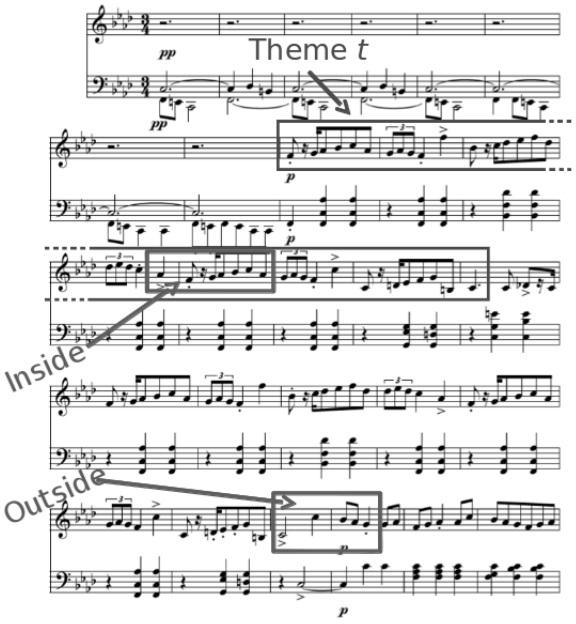


Figure 2: An example of a motif inside the theme and a motif outside the theme for a piece of music. The average normalized probability of the motifs inside the theme are compared to the average normalized probability of the motifs outside the theme.

Evaluation of Motif Discovery Process

A test set consists of 15 full music scores with one or more hand annotated themes for each score. The full scores are fetched from KernScores,[8] and the corresponding themes are removed from the training data set (taken from the aforementioned Electronic Dictionary of Musical Themes). Each theme effectively serves as a hand annotated characteristic theme from a full score of music. This process is done manually due to the incongruence of KernScores and The Electronic Dictionary of Musical Themes. In order to ensure an accurate mapping, full scores and themes are matched up according to careful inspection of their titles and contents. We attempt to choose a variety of different styles and time periods in order to adequately represent the training data.

For each score in the test set, candidate motifs are gathered into a set C by iterating through the full score, one part at a time, using a sliding window from size l_min to l_max. This is the same process used to gather candidate motifs from audio and image files. C is then split into two disjoint sets, where C_t contains all the motifs that are subsequences of the matching theme(s) for the score, and C_{-t} contains the remaining motifs. See Figure 2 for a visual example of motifs that are found inside and outside of the theme.

A statistic Q is computed which represents the mean normalized probability of the motifs in a set S given a model M:

[8] http://kern.ccarh.org/

Algorithm 3 *evaluate_discovery_process*
T is the set of all 9383 themes, V and S are sets of scores. Each $r \in V$ contains a set of themes $\{t_1...t_n\}$, $t_i \in T$ and each $s \in S$ contains a set of themes $\{u_1...u_k\}$, $u_i \in T$. $V \cap S = \emptyset$ and $\forall s \in S$ and $\forall r \in V$, $s \cap r = \emptyset$

1: **Input:** T, V, S
2: **for** each ML model class $\mathcal{M}$ **do**
3: $best = -\infty$
4: **for** each setting p of $\mathcal{M}$'s hyperparameters **do**
5: $ave = 0$
6: **for** each score $s \in V$ **do**
7: learn M_p using $T - s$ as training data
8: $ave = ave + U(s|M_p)$
9: $ave = ave/|V|$
10: **if** $ave > best$ **then**
11: $best = ave$
12: $p_{best} = p$
13: $p^*_{\mathcal{M}} = p_{best}$
14: **for** each ML model class $\mathcal{M}$ **do**
15: **for** each score $r \in R$ **do**
16: learn $M_{p^*_{\mathcal{M}}}$ using $T - r$ as training data
17: *results* $\leftarrow U(r|M_{p^*_{\mathcal{M}}})$
18: **return** *results*

$$Q(S|M) = \frac{\sum_{m \in S} norm(|m|)Pr(m|M)}{|S|} \quad (6)$$

$Q(C_t|M)$ informs us about the probability of thematic motifs being extracted by the motif discovery system. $Q(C_{-t}|M)$ informs us about the probability of non-thematic motifs being discovered. A metric U is computed in order to measure the ability of the motif discovery system to discover desirable motifs.

$$U(C|M) = \frac{Q(C_t|M) - Q(C_{-t}|M)}{\min\{Q(C_t|M), Q(C_{-t}|M)\}} \quad (7)$$

U is larger than zero if the discovery process successfully identifies motifs that have motivic or thematic qualities according to the hand-labeled themes.

Given our collected set T of 9383 themes, we use leave-one-out cross validation on a set V of music scores and their hand-labeled themes in order to fine-tune the ML model class hyperparameters to maximize U, as shown in Algorithm 3. For each score $s \in V$, we learn an ML model M from the model class $\mathcal{M}$ using $T - s$ as training data (line 7), and using the learned model we calculate the average U value for the set V (lines 8-9). We perform this validation under various hyperparameter configurations for all $s \in V$ for each ML model class (lines 2-6). After this is done, we select the hyperparameter configuration that results in the highest average value for U (lines 10-13). Finally, after these hyperparameters are tuned, we calculate U over a separate test set S of scores and themes (disjoint from V) for each model class (lines 14-17). The results are shown in Table 1.

Algorithm 4 *evaluate_motif_quality*
T is the set of all 9383 themes, F is a non-musical (inspirational) media file, $M_{p^*_{\mathcal{M}}}$ is a learned model

1: **Input:** T, F, $M_{p^*_{\mathcal{M}}}$
2: *allmotifs* $\leftarrow$ *extract_candidate_motifs* from T
3: $H_m = average\_entropy(allmotifs)$
4: *candidates* $\leftarrow$*extract_candidate_motifs* from F
5: $H_c = average\_entropy(candidates)$
6: *best* $\leftarrow$*discover_best_motifs* from *candidates* using model $M_{p^*_{\mathcal{M}}}$
7: $H_b = average\_entropy(best)$
8: *results* $\leftarrow R(T, best|M_{p^*_{\mathcal{M}}})$
9: **return** H_m, H_c, H_b, *results*

Given the data in the table, a case can be made that certain ML model classes can effectively discover thematic motifs with a higher probability than other motif candidates. Four of the six ML model classes have an average U value above zero. This means that an average theme is more likely to be discovered than an average non-theme for these four classes. PPM and CTW have the highest average U values over the test set. LSTM has the worst average, but this is largely due to one outlier of -91.960. Additionally, PST performs poorly mostly due to two outliers of -24.363 and -31.614. Except for LSTM and PST, all of the models are fairly robust by keeping negative U values to a minimum.

Evaluation of Structural Quality of Motifs

We also evaluate both the information content and the level of innovation of the discovered motifs, as shown in Algorithm 4. First, we measure the information content by computing *entropy* as we did before. We compare the entropy of the discovered motifs (lines 6-7) to the entropy of the candidate motifs (lines 4-5). We also segment the actual music themes from the training set into a set of motifs using Algorithm 1, and we add the entropy of these motifs to the comparison (lines 2-3). In order to ensure a fair comparison, we perform a sampling procedure which requires each set of samples to contain the same proportions of motif lengths, so that our entropy calculation is not biased by the length of the motifs sampled. The results for two image input files and two audio input files are displayed in Table 2, with each column for each input file the result of running Algorithm 4 twice, once for pitch and once for rhythm. The images and audio files are chosen for their textural and aural variety, and their statistics are representative of other files we tested. Bioplazm2.jpg is a computer-generated fractal while Landscape.jpg is a photograph, and Lightsabers.wav is a sound effect from the movie *Star Wars* while GalwayKinnell-Neverland.wav is a recording of a person reading poetry.

The results are generally as one would expect. The average pitch entropy is always lowest on the training theme motifs, it is higher for the discovered motifs, and higher again for the candidate motifs. With the exception of Landscape.jpg, the average rhythm entropy follows the same pattern as pitch entropy for each input. One surprising ob-

Score File Name	CTW	HMM	LSTM	LZMS	PPM	PST
BachBook1Fugue15.krn	4.405	4.015	3.047	2.896	11.657	4.951
BachInvention12.krn	-2.585	-5.609	26.699	1.078	0.534	13.191
BeethovenSonata13-2.krn	1.065	-0.145	7.769	8.876	4.973	9.182
BeethovenSonata6-3.krn	-0.715	-5.320	2.874	0.832	1.283	4.801
ChopinMazurka41-1.krn	6.902	0.808	-7.690	3.057	18.965	-24.363
Corelli5-8-2.krn	-6.398	-1.270	-0.692	-2.395	-1.166	1.690
Grieg43-2.krn	2.366	1.991	-2.622	0.857	8.800	-7.740
Haydn33-3-4.krn	14.370	2.370	1.189	6.155	8.475	0.841
Haydn64-6-2.krn	1.266	2.560	-1.092	0.855	1.809	-0.133
LisztBallade2.krn	-0.763	-0.610	-1.754	-0.046	1.226	0.895
MozartK331-3.krn	0.838	0.912	3.829	0.756	3.222	5.413
MozartK387-4.krn	-4.227	-0.082	-91.960	-2.127	-3.453	-31.614
SchubertImpromptuGFlat.krn	49.132	3.169	0.790	8.985	59.336	1.122
SchumannSymphony3-4.krn	0.666	2.825	-2.154	0.289	1.560	-6.830
Vivaldi3-6-1.krn	7.034	2.905	0.555	7.055	9.633	-0.367
Average	4.890	0.568	-4.081	2.475	8.457	-1.931

Table 1: U values for various score inputs and ML model classes. Positive U values show that the average normalized probability of motifs inside themes is higher than the same probability for motifs outside themes. Positive U values suggest that the motif discovery system is able to detect differences between thematic motifs and non-thematic motifs.

servation is that the rhythm entropy for some of the ML model classes is sometimes higher for the discovered motifs than it is for the candidate motifs. This suggests that thematic rhythms are often less predictable than non-thematic rhythms. However, the pitch entropy almost always tends to be lower for the discovered motifs than the candidate motifs. This suggests that thematic pitches tend to be more predictable.

Next, we measure the level of innovation of the best motifs discovered (line 8). We do this by taking a metric R (similar to U) using two Q statistics (see equation 6), where A is the set of 9383 themes from the training database and E is the set of discovered motifs.

$$R(A, E|M) = \frac{Q(A|M) - Q(E|M)}{\min\{Q(A|M), Q(E|M)\}} \quad (8)$$

When R is greater than zero, A is more likely than E given the ML model M. In this case, we assume that there is a different model that would better represent E. If there is a better model for E, then E must be novel to some degree when compared to A. Thus, If R is greater than zero, we infer that E innovates from A. The R results for the same four input files are shown along with the entropy statistics in Table 2. Except for PPM, all of the ML model classes produce R values greater than zero for each of the four inputs.

While statistical metrics provide some useful evaluation in computationally creative systems, listening to the motif outputs and viewing their musical notation will also provide valuable insights for this system. We include six musical notations of motifs discovered by this system in Figure 3, and we invite the reader to listen to sample outputs at `http://axon.cs.byu.edu/motif-discovery`.

Conclusion and Future Work

The motif discovery system in this paper composes musical motifs that demonstrate both innovation and value. We show that our system innovates from the training data by extracting candidate motifs from an inspirational source without generating data from a probabilistic model. This assumption is validated by observing high R values.

Additionally, the motif discovery system maintains compositional value by grounding it in a training data set. The motif discovery process is tested by running it on actual music scores instead of audio and image files. The results show that motifs found inside of themes are on average more likely to be discovered than motifs found outside of themes.

Improvements and modifications can be made in the analysis and methodology of our system. We are currently preparing another manuscript which evaluates the difference between motifs discovered by our system and comparable random motifs. The results show that using (non-musical) media as inspiration for the motif discovery process is more efficient at producing "musical" motifs than is randomly generating "reasonable" motifs.

The discovered motifs are the contribution of this system. While work presented here is a proof-of-concept for the use of non-musical media sources as inspiration in creating musical motifs, more sophisticated techniques should be explored. In the future, we plan to utilize machine vision to extract meaning from images; we plan to study saccades from human subjects on various images in order to train the computer to see them in a more human, natural way; and we plan to incorporate digital signal analysis on audio files in order to hear audio more like a human would hear it. (While it is certainly not necessary for a computer to be inspired in the same way as a human might be, if the goal is to compose music that people can appreciate, it seems worthwhile to explore human-centric models of musical inspiration.)

In addition to improving the motif creation process, future work will investigate combining these motifs, adding harmonization, and creating full compositions. This work is simply the first step in a novel composition system. While there are a number of directions to take with this system as

Bioplazm2.jpg	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
pitch entropy training motifs	1.894	1.979	1.818	1.816	1.711	1.536	1.793
pitch entropy discovered motifs	2.393	2.426	1.944	1.731	2.057	1.759	2.052
pitch entropy candidate motifs	2.217	2.328	2.097	2.104	1.958	1.784	2.081
rhythm entropy training motifs	1.009	1.051	0.976	0.970	0.927	0.822	0.959
rhythm entropy discovered motifs	2.110	2.295	1.789	2.212	0.684	1.515	1.767
rhythm entropy candidate motifs	2.387	2.466	2.310	2.309	2.132	1.934	2.256
R	7.567	13.296	20.667	4.603	-0.276	7.643	8.917
Landscape.jpg	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
pitch entropy training motifs	1.894	1.979	1.818	1.816	1.711	1.536	1.793
pitch entropy discovered motifs	1.974	2.074	2.143	1.833	2.027	1.675	1.954
pitch entropy candidate motifs	2.429	2.531	2.598	2.341	2.271	2.028	2.367
rhythm entropy training motifs	1.009	1.051	0.976	0.970	0.927	0.822	0.959
rhythm entropy discovered motifs	1.984	1.863	2.175	1.983	0.727	1.455	1.698
rhythm entropy candidate motifs	1.549	1.712	1.810	1.509	1.396	1.329	1.551
R	0.805	0.236	1.601	0.429	4.624	1.283	1.496
Lightsabers.wav	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
pitch entropy training motifs	1.894	1.979	1.818	1.816	1.711	1.536	1.793
pitch entropy discovered motifs	2.076	1.884	1.881	1.652	2.024	1.586	1.850
pitch entropy candidate motifs	2.225	2.097	2.217	1.876	2.115	1.755	2.048
rhythm entropy training motifs	1.009	1.051	0.976	0.970	0.927	0.822	0.959
rhythm entropy discovered motifs	1.534	1.309	2.024	1.623	0.860	1.225	1.429
rhythm entropy candidate motifs	1.540	1.524	1.541	1.502	1.548	1.276	1.489
R	5.637	0.793	27.227	4.812	6.768	7.540	8.796
GalwayKinnell-Neverland.wav	**CTW**	**HMM**	**LSTM**	**LZMS**	**PPM**	**PST**	**Average**
pitch entropy training motifs	1.894	1.979	1.818	1.816	1.711	1.536	1.793
pitch entropy discovered motifs	1.823	2.480	2.132	1.773	1.997	1.701	1.984
pitch entropy candidate motifs	2.153	2.248	2.250	2.141	2.242	1.839	2.146
rhythm entropy training motifs	1.009	1.051	0.976	0.970	0.927	0.822	0.959
rhythm entropy discovered motifs	1.550	1.587	1.560	1.779	0.289	1.128	1.315
rhythm entropy candidate motifs	1.472	1.469	1.471	1.477	1.469	1.226	1.431
R	1.520	10.163	24.968	4.283	0.257	6.865	8.010

Table 2: Entropy and R values for various inputs. We measure the pitch and rhythm entropy of motifs extracted from the training set, the best motifs discovered, and all of the candidate motifs extracted. On average, the entropy increases from the training motifs to the discovered motifs, and it increases again from the discovered motifs to the candidate motifs. The R values are positive when the training motifs are more probable according to the model than the discovered motifs. Higher R values represent higher amounts of innovation from the training data.

a starting point, we are inclined to compose from the bottom up. Longer themes can be constructed by combining the motifs from this system using evolutionary or other approaches. Once a set of themes is created, then phrases, sections, and multiple voices can be composed in a similar manner. Contrastingly, another system could compose from the top down, composing the higher level features first and using the motifs from this system as the lower level building blocks. This system could also be extended by including additional modes of inspirational input such as text or video. Our intent is for this system to be the starting point for an innovative, high quality, well-structured system that composes pieces which a human observer could call creative.

References

Begleiter, R.; El-Yaniv, R.; and Yona, G. 2004. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research* 22:385–421.

Biles, J. 1994. GenJam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, 131–137.

Blackwell, T. 2003. Swarm music: improvised music with multi-swarms. In *Proceedings of AISB Symposium on Artificial Intelligence and Creativity in Arts and Science*, 41–49.

Bruhn, S. 1997. *Images and Ideas in Modern French Piano Music: the Extra-musical Subtext in Piano Works by Ravel, Debussy, and Messiaen*, volume 6. Pendragon Press.

Burton, A. R., and Vladimirova, T. 1999. Generation of

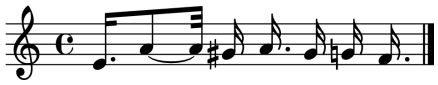
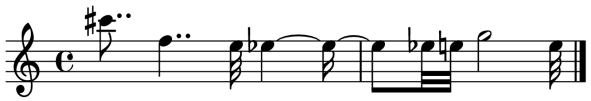
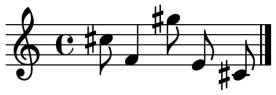
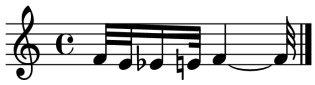
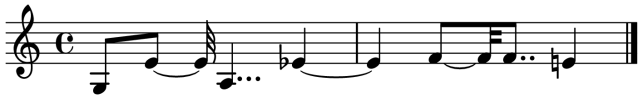
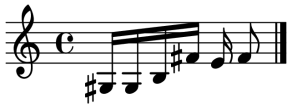
ML Model	Input File	Motif Discovered
CTW	*MLKDream.wav*	
HMM	*Birdsong.wav*	
LSTM	*Pollock-Number5.jpg*	
LZMS	*Lightsabers.wav*	
PPM	*Bioplazm2.jpg*	
PST	*GalwayKinnell-Neverland.wav*	

Table 3: Six motifs discovered by our system.

musical sequences with genetic techniques. *Computer Music Journal* 23(4):59–73.

Cardoso, A.; Veale, T.; and Wiggins, G. A. 2009. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine* 30(3):15–22.

Cope, D. 1996. *Experiments in Musical Intelligence*, volume 12. AR Editions Madison, WI.

Dubnov, S.; Assayag, G.; Lartillot, O.; and Bejerano, G. 2003. Using machine-learning methods for musical style modeling. *Computer* 36(10):73–80.

Eck, D., and Lapalme, J. 2008. Learning musical structure directly from sequences of music. Technical report, University of Montreal, Department of Computer Science.

Eck, D., and Schmidhuber, J. 2002. Learning the long-term structure of the blues. In *Proceedings of the International Conference on Artificial Neural Networks*. 284–289.

Hochreiter, S.; Bengio, Y.; Frasconi, P.; and Schmidhuber, J. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press. 237–244.

Lavrenko, V., and Pickens, J. 2003. Music modeling with random fields. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 389–390.

Smith, B. D., and Garnett, G. E. 2012. Improvising musical structure with hierarchical neural nets. In *Proceedings of Eighth Artificial Intelligence and Interactive Digital Entertainment Conference*, 63–67.

Smith, R.; Dennis, A.; and Ventura, D. 2012. Automatic composition from non-musical inspiration sources. In *Proceedings of International Conference on Computational Creativity*, 160–164.

Spiliopoulou, A., and Storkey, A. 2012. A topic model for melodic sequences. *ArXiv E-prints*.

Weiland, M.; Smaill, A.; and Nelson, P. 2005. Learning musical pitch structures with hierarchical hidden Markov models. Technical report, University of Edinburgh.

White, J. D. 1976. *The Analysis of Music*. Prentice-Hall.

Wiggins, G. A.; Pearce, M. T.; and Müllensiefen, D. 2009. Computational modelling of music cognition and musical creativity. *Oxford handbook of computer music* 383–420.

Non-Conformant Harmonization: the Real Book in the Style of Take 6

François Pachet, Pierre Roy
Sony CSL
Paris, France
pachetcsl@gmail.com

Abstract

We address the problem of automatically harmonizing a leadsheet in the style of any arranger. We model the arranging style as a Markov model estimated from a corpus of non-annotated MIDI files. We consider a vertical approach to harmonization, in which chords are all taken from the arranger corpus. We show that standard Markov models, using various vertical viewpoints are not adapted for such a task, because the problem is basically over constrained. We propose the concept of *fioriture* to better capture the subtleties of an arranging style. Fioritures are ornaments of the given melody during which the arranging style can be expressed more freely than for melody notes. Fioritures are defined as random walks with unary constraints and can be implemented with the technique of Markov constraints. We claim that fioritures lead to musically more interesting harmonizations than previous approaches and discuss why. We focus on the style of Take 6, arguably the most sophisticated arranging style in the jazz genre, and we demonstrate the validity of our approach by harmonizing a large corpus of standard leadsheets.

Introduction

Automatic harmonization has been addressed for decades by computer music research (see Steels, 1986 for an early attempt at machine-learning of harmonization and Fernandez and Vico, 2013 for a survey). One reason for the success of this problem in the research community is that it can be considered, in first approximation, as a *well-defined problem*, a crown jewel in computer music. Automatic harmonization denotes in practice many different problems, depending on the nature of the input (melody, chord labels, bass, song structure given or not) and of the output (chord labels, chord realizations, contrapuntal voices), the constraints concerning the nature of the targeted harmonization (number of voices) and the way the targeted style is modeled (programmed explicitly or learned from examples). A widely studied variant of the automatic harmonization problem is the generation of a four-part (or more) harmonization of a given melody. Such a problem has been tackled in a variety of contexts, though mostly for classical music, Bach chorales in particular, and using virtually all the technologies available including rules, functions (Koops et al. 2013), grammars, constraints (Anders and Miranda, 2011), and statistical models of all types (Paiement et al. 2006).

Today, there are many approaches that work satisfactorily to produce harmonizations in the Classical style with reasonable musical quality. It is remarkable that automatic harmonization has achieved such a status of well-definedness that many papers in this domain consist in variations of existing algorithms, with little or no musical output (a sign, probably of the maturity of the field). However, there is no system, to our knowledge, that is able to produce truly musically interesting harmonizations, at least for the ears of musically trained listeners such as the first author of this paper. In the context of computational creativity, we claim that there are two problems with the current state of the art which limit their quality, and therefore their possibility for generating creative outputs: *excess of conformance* and *excess of agnosticism*.

Conformance. Automatic harmonization has so far been envisaged solely under the viewpoint of harmonic conformance: the main criterion of success is that the generated material has to conform to the harmonic constraints of the problem. For instance, a harmonic label of C minor (either imposed or inferred from, say, a soprano) should produce chord realizations that *conform* to C minor, for instance, chords composed of important notes of the scale. Conformance yields indeed a well-defined measure to evaluate systems, because there are well-defined harmonic distances (see Section Harmonic Distance), but tends to go in the way of creativity, since the best a system can do is to paraphrase harmonic labels. Such a skill can be impressive for non-musicians, but not for experts. Consequently, many harmonization systems give the impression that they are essentially filling the blanks (inner voices) with correct but uninteresting musical excipients. This is sometimes referred to as the "correct" versus "good" problem, but in fact, such harmonizers are basically unable to produce interesting solutions, because of excess in conformance.

Agnosticism (excess of generality). Most works, with the exception of (Ebcioglu, 1986), attempt to model a given style using general methods (such as Markov models, rules, etc.). General methods can be good in general, but are rarely very good in particular. Similarly to the famous "glass ceiling" problem occurring in MIR (Casey et al., 2008), there seems to be a glass ceiling concerning the musical quality of automatic harmonization. In our view this is caused by the use of too general methods and by the absence of consideration for the details of what makes a specific style interesting or creative. Most often, these details are not captured by general methods.

In this study, we focus on the harmonization style of the American six-voice *a cappella* band Take 6. Take 6 is the most awarded vocal group in history. Since their first two albums (Take 6, 1988; 1990) they renewed the genre of gospel barbershop-like harmonization by pushing it to its harmonic and vocal limits. Their style of arranging is considered unanimously as extraordinarily inventive, recognizable, and very difficult to imitate. Even the transcription of their performances is a very difficult task that only harmony experts can perform correctly (see Section Acknowledgements). Most of their works consist in 6-voice note-to-note harmonization of traditional songs, with many dissonances and bold voice movements typical of jazz big bands. The creativity of Take 6, if any, consists precisely in the use of those dissonances and digressions. Of course, their style and specificity is arguably also dependent on the quality of the singing voices (notably the bass), but this dimension is outside the scope of this paper, and we consider here only the symbolic aspects of their arranging style.

Most knowledgeable listeners of Take 6 enjoy "wow" effects due to their spectacular use of harmonic surprises. Figure 1 shows an excerpt of a harmonization by Take 6 of the traditional "Hark the Herald Angels Sing". Figure 2 shows an estimation of the corresponding excerpt of the leadsheet (end of section A). It can be seen clearly that the chords used to harmonize the note B♭ do not conform to the expected harmony of B♭ major: although the performance of Take 6 are not labeled, we can estimate the last realization of the B♭ as an instance of a *C7dim9#11* (*C E G Bb Db F#*), which is very far from the expected Bb major scale, or of any scale close by (such as relative minors). Such a harmonic surprise is typical of the style of Take 6. By definition, conformant methods in automatic harmonization are not able to capture this kind of knowledge, especially from non-labeled training data.

Our goal is to produce six-voice harmonization in that style that triggers the same kinds of "wow" effects as the originals. The key idea of our approach is that most wow effects are obtained by *non-conformant harmonizations*, i.e., harmonizations that do not conform to the harmonic labels of the original leadsheet, but stay within well-defined constraints. The technical claim of this paper it that the technology of Markov constraints (Pachet et al., 2011) is particularly well suited for such a task, thanks to the possibility of generating creative sequences within well-defined constraints.

Problem Statement

The problem we address constitutes a variation on standard harmonization problems such as melody or bass given. It can be defined in terms of inputs/outputs as follows:

Inputs:

- A leadsheet representing the target melody to harmonize, as well as chord labels in a known syntax (i.e., we know their pitch constituents),
- A harmonization style represented by a set of non-annotated scores containing polyphonic content. No annotation of these scores is needed. In practice, arbitrary MIDI files may be used, including files without a fixed tempo coming from, e.g., recordings of real-time performances.

The expected output is a fully harmonized score, in the given style, i.e. a polyphonic score that maintains the the soprano of the leadsheet, and whose harmonies fit with the leadsheet chord labels.

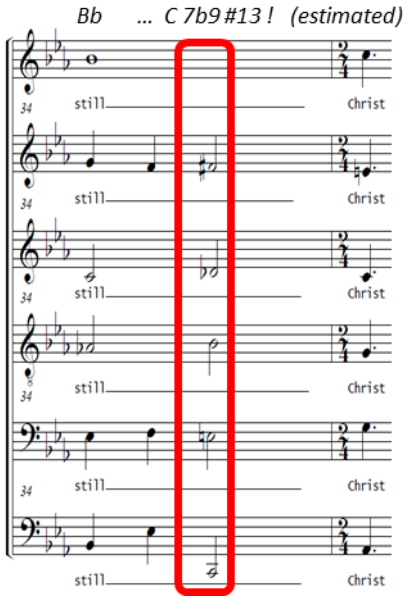


Figure 1. Example of a typical non-conformant harmonization by Take 6. Harmonies used (estimated from the score) go from Bb (which conforms to the leadsheet) to a surprising, non-conformant C7dim9#11 (Transcription by A. Dessein).

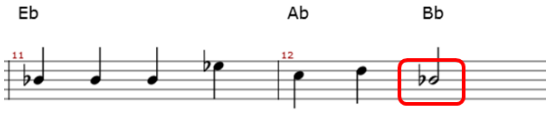


Figure 2. Extract of a leadsheet for "Hark the Herald Angels Sing" (end of section A). The last B*b* is supposed to be harmonized in Bb (shortcut for Bb major).

Musically, the goal is to produce a harmonization that is *reminiscent* of the style, i.e., such that knowledgeable listeners can recognize the authors. However this is not a well-defined problem, for several reasons: listeners may not recognize a style because they do not know the arranger well enough, or because they give more importance to the sound than to the notes, or for many other reasons, including that the arranger may not have any definite style *per se*. In this paper, we do not attempt to solve the harmonization problem in many styles (though the system can, as exemplified in Section Applications to Other Styles). Rather, we attempt to convince ourselves, as knowledgeable Take 6 listeners, that our system grasps some of their subtle arranging tricks and reproduce them in unknown situations. A scientific evaluation of the system based on style recognition is in progress but is not the subject matter of this paper.

Corpora Used

The experiments we describe use a comprehensive database of jazz leadsheets described in (Pachet et al., 2013). For each leadsheet we have a melody (monophonic sequence of notes) and chord labels. For each chord label, the database provides the set of pitch-classes of the chord, in ascending order (that is, the formal definition of the chord, not its realization). In this study we used the Real Book (illegal edition), the most widely used jazz fake book. The Real book contains about 400 songs, 397 of which are parsed correctly (a few songs with no harmony or no melody are ruled out for instance).

For the harmonization style, we have selected a number of composers including classical ones (Wagner, Debussy, etc.) and jazz (Take 6 notably, and Bill Evans). Each composer is represented by a set of MIDI files of some of their compositions or performances. All MIDI files have been found on the web, except the 10 MIDI files of Take 6 that were provided to us by a human transcriber (A. Dessein). The Take 6 MIDI files are of excellent quality (i.e., there are virtually no transcription errors). The other MIDI files are of varying quality. Some of them correspond to actual scores (Wagner), others to performances (Bill Evans). IN order to cope with the diversity of tonalities and pitch ranges encountered in the leadsheet melodies, we have transposed systematically the corpus in all 12 keys.

Homophonic Harmonization

The approach we follow consists in considering the harmonization problem as a *vertical* problem, as opposed to voice-leading approaches (such as Whorley et al., 2013), and following an older tradition initiated by (Pachet and Roy, 1995) on constraint-based 4-voice harmonization in the Classical style. To compensate for the monotony of strict vertical harmonization, we complement this step by a smoothing procedure that somehow reestablishes voice-leading *a posteriori* from the vertical skeleton structure, by joining contiguous notes with the same pitch. This second step is completely deterministic, and the central issue we address is the production of the chordal skeleton.

Before describing the harmonization process, we introduce a measure of *harmonic conformance*, which is at the core of the whole process.

Harmonic Conformance

Because the scores of arrangers are not labeled, we need a way to relate chord realizations found in the arranger corpus to chord labels of a leadsheet. In order to avoid the pitfalls of chord recognition (which works well for simple chords, but much less for the complex chords as found in jazz), we use a simple but robust measure of the harmonic conformance between unlabeled chords. This measure, called ε-conformance is based on pitch class histograms.

For any chord realization C_i, i.e., a set of MIDI pitches, we build a pitch class histogram as an array of 12 integers, where each integer represents the number of occurrences of the corresponding pitch class in the chord (starting with C up to B), normalized by the total number of pitches. For instance, the circled chord in Figure 1 has a pitch-class frequency count $f = [0,1,1,0,1,0,1,0,0,1,0,0]$. The histogram is the frequency count divided by its module $h_i = \frac{f_i}{\sum_{k=1}^{12} {f_k}^2}$. The harmonic distance between two chords C_1 and C_2 can then be defined as the scalar product of the pitch class histograms:

$$D(C_1, C_2) = 1 - \sum_{k=1}^{12} h_1^k \times h_2^k.$$

where h_1(resp. h_2) is the pitch-class histogram of chord C_1 (resp. C_2). Such a distance takes its values in $[0, 1]$.

In practice, this distance enables us to categorize chord realizations appearing in the arranger corpus with regards to a given chord label. For each chord label, we can define an ideal prototype consisting of its pitch class definition, and then consider the ball centered on this ideal prototype of radius ε. ε represents the "harmonic conformance" of a chord realization to a chord label. Increasing values of ε provide increasingly large sets of chords, that are more or less conformant to the label. Figure 3 shows examples of chords at various distances to "C 7" for various values of ε in the Take 6 corpus.

Another way to relate chord realizations to chord labels is to consider the *best match* for a given corpus: the chord in the arranger corpus with the minimal harmonic distance to the ideal realization of the label. We then consider the ball centered around this best match, of radius ε. In any case, pitch class histograms provide us with a robust way to fetch chord realizations for any chord label, in non-annotated corpora.

Unary Markov Constraints

Equipped with a harmonic distance, we can generate new chordal skeletons. The idea is to estimate a Markov model of the sequences of chord realizations from the arranger corpus. The leadsheet (soprano movement and chord labels) is represented as a set of unary constraints holding on the sequence to generate. The framework of Markov constraints (Pachet et al., 2011), is precisely designed to handle such cases, and provides an efficient algorithm to generate those sequences, as well as a guarantee that all sequences satisfying the constraints will be found, with their correct probability in the original model. Solving a Markov constraint problem is strictly equivalent to sampling the sequences in the space of solutions. Each sequence $s = s_1, .., s_n$, has a probability $p(s) = p(s_1) \times \prod_{i=1}^{n-1} p(s_{i+1}|s_i)$ according to the considered Markov model (see next section). The unary Markov constraint algorithms guarantee that all sequences satisfying the constraints are drawn with their probability in the original model.

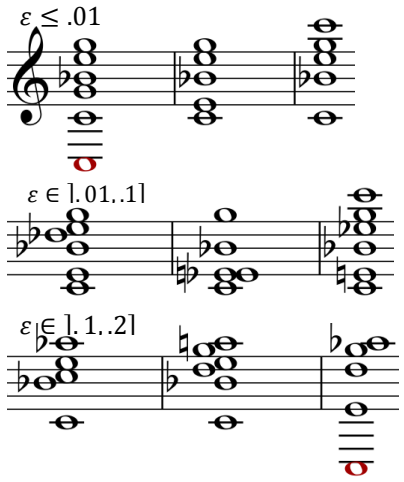


Figure 3. Various chord realizations from the Take 6 corpus for several values of ε (0.01, 0.1 and 0.2), representing increasing harmonic distance to a *C 7* chord label. As ε increases, more notes outside of the legal notes of C 7 (C, E, G, Bb) are added. For $\varepsilon = 1$ (maximum distance) all possible chords of the corpus are considered. In practice, reasonable, conformant realizations lie within a distance of about .15.

Viewpoints

Such a process raises an important issue concerning the choice of the viewpoint, i.e. the actual data used to estimate the Markov model. The most demanding viewpoint is the actual set of notes (Midi pitches) of the chord. This is called here the *Identity* viewpoint, since it contains all the information we have on a chord. Degraded viewpoints are also considered: *BassTenorSoprano* is the viewpoint consisting of the bass, tenor and soprano pitches (and ignoring the others). We define similarly the *BassSoprano* and *Soprano* viewpoints. For the sake of comparison, we also introduce the *Constant* viewpoint, which assigns a constant value to any chord (and serves as a base line for our experiments). Note that we do not consider duration information, as we do not want to rely on the quality of the MIDI Files.

Of course there is a tradeoff here between 1) harmonic conformance, represented here by ε, and 2) style conformance, which manifests itself by the presence of chord transitions that actually occurred in the corpus. Such a tradeoff between adaptation and continuity is not novel, and has been studied in automatic accompaniment (Cabral et al., 2006; Marchini and Purwins, 2010). In our context, it is formulated as a tradeoff between ε and viewpoint selectiveness. The most demanding viewpoint generate chord sequences that sound more natural in the given style, since they replicate actual transitions of chord realizations occurring in the corpus. However, such chord transitions will generate a sparse Markov model. The consequence is that only a very small number of leadsheets can be harmonized in that way for small values of ε. By degrading the viewpoints, more transitions will be available, so smaller (more conformant) values of ε can be considered.

Harmonizing the Real Book

In order to illustrate the harmonic conformance / viewpoint tradeoff, we describe a basic experiment that has, to our knowledge, never been conducted, at least on such a scale. For several values of ε we study the sparsity of the four viewpoints introduced above, by counting how many songs from the Real Book can be harmonized entirely with the viewpoint.

More precisely, for each leadsheet taken from the Real Book (397), we build a Markov Constraint problem consisting of the following constraints:

- Generate a sequence of chord realizations taken exclusively from the Take 6 corpus, transposed in all 12 pitches (variable domains),
- Each note of the leadsheet is harmonized by one chord realization (homophonic note-to-note harmonization),
- Transitions between 2 chord realizations c_i and $c_{i+1}$are all Markovian for the considered viewpoint, i.e. $p(c_{i+1}|c_i) > 0$,
- Each chord c_{i+1} has a soprano which is the leadsheet note
- Each chord realization c_{i+1} must be ε-conformant to the corresponding leadsheet chord label, for the chosen value of ε.

These constraints can all be implemented as a unary Markov constraint problem. The experiment consists in counting, for each value of ε in $[0, 1]$ and for each of the four viewpoints how many songs from the Real Book can be fully harmonized. The results are presented in Figure 4. It can be seen clearly that with non-trivial viewpoints (i.e. all viewpoints but soprano), solutions are found only for high values of ≤ε. For those values, harmonic conformance is lost. Only the basic *Soprano* viewpoint leads to many solutions (160, a value insensitive to ε). It can be noted that the *Constant* viewpoint (a trivial viewpoint that consists in basically removing the Markovian constraint), solutions are found for 262 songs. This means that there are 102 songs for which the *Soprano* viewpoint does not lead to any solution, for any value of ε. This corresponds to songs that contain pitch transitions that never occur between two consecutive realized chords in the Take 6.

It is important to note here that when no solution is found for a given leadsheet / viewpoint combination / value of ε, this does *not* necessarily implies that the leadsheet contains a transition for which there is no match in the corpus (for the given viewpoint). It means that there is no *complete* solution, i.e. transitions compatible with each other so as to make up a complete solution sequence.

This experiment shows clearly that harmonic conformance is somewhat incompatible with precise Markov models of chord realizations, for a realistic corpus (Take 6) on a realistic test database (the Real Book). However, we can use the Soprano viewpoint as a basis for producing interesting harmonization of most "reasonable" leadsheets, with a clear control on harmonic conformance.

Figure 5, Figure 6 and Figure 7 show homophonic harmonization of the four first bars of Giant Steps with various values of ε. It can be noted that while harmonic conform-

ance can be used as a parameter to generate more or less conformant realization, the results are academically correct, but rarely very interesting musically. The style of the arranger is hard to recognize, because there are not enough actual transitions that are being reused from the corpus. The control of harmonic conformance can generate surprises, but at the price of losing the essence of the style.

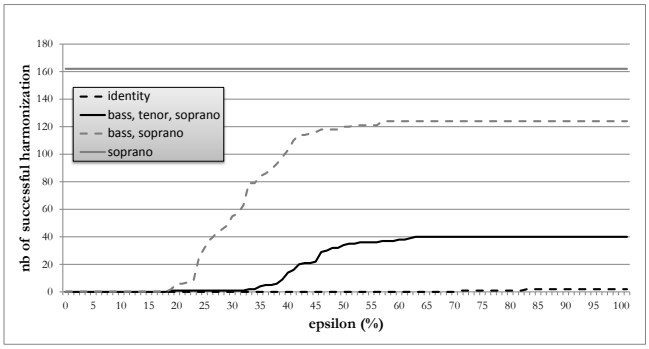


Figure 4. Graph showing the number of successful harmonization from the Real Book (illegal edition) using a Markov model of chord realizations, and various viewpoints of decreasing precision (identity, bass/tenor/soprano, bass/soprano, soprano).

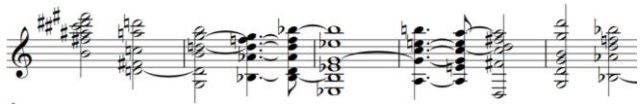


Figure 5. The beginning of Giant Steps harmonized with a value of $\varepsilon \in [0, 0.01]$. All realizations come from the Take 6 corpus satisfy exactly the chord labels. The overall harmonization is conformant but not very interesting.

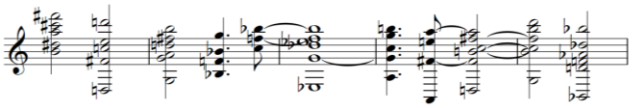


Figure 6. The beginning of Giant Steps with $\varepsilon \in [0, 0.2]$. The chords are less conformant and more interesting, but the whole harmonization still lacks surprise.

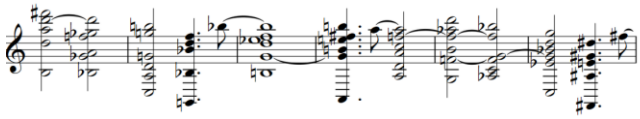


Figure 7. The beginning of Giant Steps with $\varepsilon \in [.3, .4]$. Chords are clearly farther away from the label, while retaining some flavor of the labels. However the decrease in harmonic conformance is musically not very interesting.

In order to express the harmonization style more clearly, and simultaneously bring creativity in the harmonization process, we introduce the concept of Fioriture.

Fioritures as a stylistic device

The idea of fioriture comes from a simple observation of polyphonic scores written by masters: It is difficult to be inventive on short duration notes. However, long notes raise opportunities to express a style: the longer a note is, the more possibilities of invention the arranger has. In the context of leadsheet based harmonization, we therefore introduce the concept of fioriture as a free variation, in the style of the arranger, occurring exactly during a *long* note, and making sense with its context.

A Simple Fioriture Example

We illustrate the concept of fioriture on a simple example. The task is to harmonize the melody shown in Figure 8: two notes with simple chords labels (both notes belong to the chord triads).

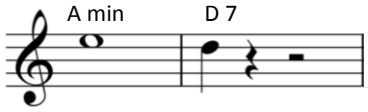


Figure 8. A simple melody to harmonize with fioritures.

This melody can be harmonized homophonically as described above, as illustrated in Figure 9.

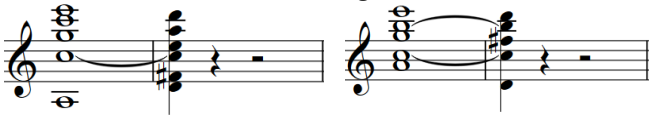


Figure 9. Two homophonic harmonization of the melody in Figure 10, with $\varepsilon \in [0, .1]$and $\varepsilon \in [.1, .2]$ respectively. A higher value of ε, the second one is more jazzy with a 9th added to the first chord and a 6th to the second one.

We can generate here a fioriture on the first note, since its duration is 4 beats. The Markov constraint problem corresponding to this fioriture is the following:

- First, select a rhythm for a note starting on the first beat of a 4/4 bar, and lasting 4 bars (rhythm selection is described in the next section). Let n be the number of notes, we generate $n + 1$ chord realizations to include the chord on the following note (here a D).
- The domain of the first chord contains only chords whose soprano is the first melody note (here, A).
- We can choose here a demanding viewpoint such as the *identity* viewpoint because in most cases the constraints above are not too hard.

Figure 10 shows various solutions, with increasing number of notes in the fioriture. It should be noted that all fioritures start from a soprano *A* on a *Amin* chord and end on a soprano *D* on a *D7* chord. However, some of them, in partic-

ular the last ones, deviate substantially from the chord labels. In short, they achieve musically meaningful harmonic non conformance. To our knowledge, only Markov constraints can compute quickly distributions of solutions of such problems.

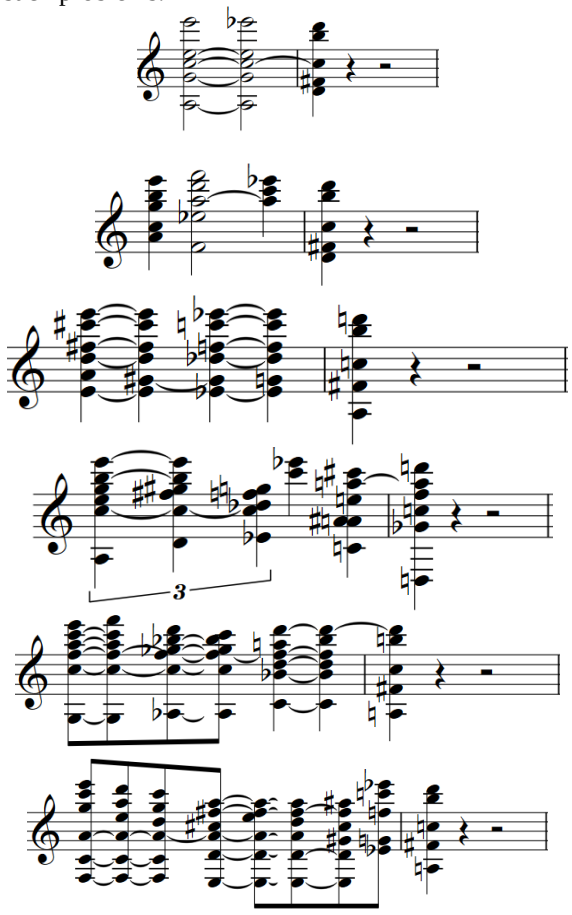


Figure 10. Fioritures with various numbers of notes. First one introduces an interesting chromaticism (E to Eb then to D); second example (3 notes) introduce a clearly non conformant chord, that resolves nicely to the D; third example (4 notes) consists in a bold chromatic descent from A minor to D; fourth example (5 notes) uses an interesting triplet-based rhythm that also departs substantially from the A minor chord label; last example is a remarkable jazzy sequence of chords.

Common-sense rhythms

One difficulty that arises when creating fioritures is to find an adequate rhythm for the generated chords. One solution would be to try to imitate rhythm as found in the arranger corpus, but this implies that the corpus used is perfectly reliable, and that metrical information is provided, which is not the case with MIDI files obtained from performances. More importantly, generating Markov sequences with durations raise sparsity issues that do not have general solutions. Another argument is that the rhythm of the fioriture should comply with the genre of the leadsheet more than of the arranger's corpus.

In this study, we have exploited the statistical properties of the leadsheet database to find *commonsense* rhythms that fit with the leadsheet to harmonize. For each rhythm to generate, we query the database to retrieve all the "melodic rhythms" that occur in all jazz standards, at the given metrical position. For a given leadsheet note to harmonize, we retrieve all melodic extracts starting at the same metrical position in the bar, and of the same duration. We then draw a rhythm at random, weighted by its probability in the database. Such a method can be parameterized in many ways (imposing the number of notes, the presence of rests, filter out by composer, genre, etc.). Figure 11 and Figure 12 show the most frequent rhythms found by such a query on the Real book, for 2 different configurations (starting beat in bar and duration).

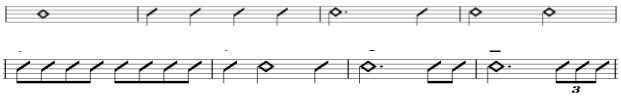


Figure 11. The 8 most frequent rhythms for a note starting on the first beat of a 4/4 bar with a 4 beat duration, from the Real Book, with their respective frequencies. Query returned 6 062 occurrences of 670 different rhythms.

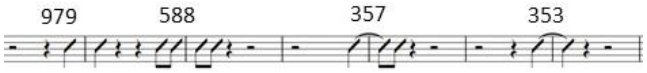


Figure 12. The four most frequent rhythms for a note starting on the last beat of a 4/4 bar with a 2 beat duration, and their respective frequencies. Query found 3943 occurrences of 111 different rhythms.

Full Examples

Two examples of Giant Steps harmonized with fiortiures are given in annex. One in the style of Take 6, and another one in the style of Richard Wagner's tetralogy. In both cases, it can be said that the musical quality is high, compared to previous approaches in automatic harmonization. Preliminary experiments were conducted by playing some harmonizations to highly trained experts (a world famous Brazilian composer, a harmony professor at Goldsmiths College, a talented jazz improviser and teacher, a professional UK jazz pianist): all of them acknowledge that the system produces highly interesting outputs. A full evaluation is under study to try to evaluate precisely the impact of fioritures on the perception of the piece, but is seems reasonable to say that they increase the musical creativity of the software in a significant manner.

Applications to Other Styles

This paper has focused on the style of Take 6, because of the acknowledged difficulty in modeling their productions. Our approach clearly improves on previous attempts at modeling barbershop harmonization such as (Roberts,

2005), who concludes his study by: "although it is possible to formalize the creative process into rules, it does not yield 'good' arrangements". We think we have reached a reasonable level of musical quality here. Our approach, however, is applicable to other styles, as this paper shows with the case of Wagner. Technically our approach is able to harmonize most leadsheets in any style defined by at least one more polyphonic MIDI files, but we did not conduct any specific musical evaluation in other styles yet.

Conclusion

We have introduced the concept of fioriture to harmonize leadsheets in the style of any arranger. Fioritures are controlled random walks within well-defined boundaries defined by long notes in the melody to harmonize. Fioritures could be envisaged under the framework of HMM (as in Farbood and Schoner, 2001). However, HMMs use chord labels as hidden states so we would need an annotated corpus, which is not the case. Furthermore, annotating Take 6 scores with chord labels is in itself an ill-defined problem. Finally, HMM cannot be controlled as precisely and meaningfully as Markov constraints.

Our approach works with non-annotated, non voice-separated corpora for modeling the arranging style. It only requires a definition of chord labels used in the leadsheet (as sets of pitch classes).

Like all music generation systems a rigorous evaluation of our approach is difficult. We claim that our system works remarkably well for most cases, as it rarely makes blatant musical errors, and most often produces musically interesting and challenging outputs. Beyond automatic harmonization, the possibility to control manually fioritures (when, with which parameters) paves the way for a new generation of assisted composition systems. Our approach could be easily extended to exploit social preferences, to help the system choose chords that "sound right" to listeners and ruling out the ones that do not.

Fioritures can also be used as a creative device. By forcing fioritures to have many notes, or by manually substituting chosen leadsheet notes by others, one can generate harmonizations in which the original melody become less and less recognizable, and the style of the arranger becomes increasingly salient. Finally we want to stress that using fioritures to express style is a paradox: fioritures (from italian *fioritura*, flowering) are supposed to be decorative, as opposed to core melody notes, i.e. are not considered primary musical elements. But in our highly constrained context, they can become a device for creative expression.

Acknowledgements

This research is conducted within the Flow Machines project, which received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 291156. We thank A. Dessein for providing us with perfect transcriptions of Take 6 recordings. The accompanying web site[1] gives examples of harmonization of jazz standards.

References

Anders, T. and Miranda, E. R. 2011. Constraint programming systems for modeling music theories and composition. *ACM Comput. Surv.* 43(4): 30.

Ebcioglu, K. 1986. An Expert System for Chorale Harmonization. *Proceedings of AAAI*: 784-788.

Cabral, G. Briot, J.-P., Pachet, F. 2006. Incremental Parsing for Real-Time Accompaniment Systems. *Proc. of 19th FLAIRS Conference*, Melbourne Beach, USA.

Casey, M. A., Veltkamp, R., Goto, M., Leman, M., Rhodes, C., & Slaney, M. 2008. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, *96*(4), 668-696.

Farbood, M. M. and Schoner, B. 2001. Analysis and Synthesis of Palestrina-Style Counterpoint Using Markov Chains. *Proceedings of the 2001 International Computer Music Conference*: 111-117.

Fernández, . D. and Vico, F. J. 2013. AI Methods in Algorithmic Composition: A Comprehensive Survey. *J. Artif. Intell. Res. (JAIR)* 48: 513-582

Koops, H. V. and Magalhães, J. P. and de Haas, W. B. 2013. A functional approach to automatic melody harmonisation. *Proceedings of the first ACM SIGPLAN workshop on Functional art, music, modeling & design (FARM '13)*, ACM, New York, NY, USA, 47-58.

Marchini, M. and Purwins, H. 2010. Unsupervised Analysis and Generation of Audio Percussion Sequences. Proceedings of CMMR: 205-218.

Pachet, F. Roy, P. 1995. Integrating constraint satisfaction techniques with complex object structures, 15th Annual Conf. of the British Comp. Society, Cambridge, pp. 11-22.

Pachet, F., Roy, P. and Barbieri, G. 2011. Finite-Length Markov Processes with Constraints. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*: 635-642, Barcelona.

Pachet, F., Suzda, J., and Martin, D. 2013. A Comprehensive Online Database of Machine-Readable Lead Sheets for Jazz Standards. *Proceedings of ISMIR: 275-280*, Curitiba (Brazil).

Paiement, J.-F. Eck, D. and Bengio, S. 2006. Probabilistic Melodic Harmonization. Proc. of *Canadian Conference on AI: 218-229.*

Roberts, S. 2005. Automated Harmonisation of a Melody into a Four Part Barbershop Arrangement, *Bsc thesis report*, University of Bath.

Steels, L. 1986. Learning the Craft of Musical Composition, Proc. of ICMC.

Take 6. 1988. *Take 6*, Warner Bros. 1988.

Take 6. 1988. *So Much 2 say*, Warner Bros. 1990.

Whorley, R. Rhodes, C. Wiggins, G. and Pearce, M. 2013. Proceedings of the Fourth International Conference on Computational Creativity: 79-86.

[1] http://www.flow-machines.com/harmonization

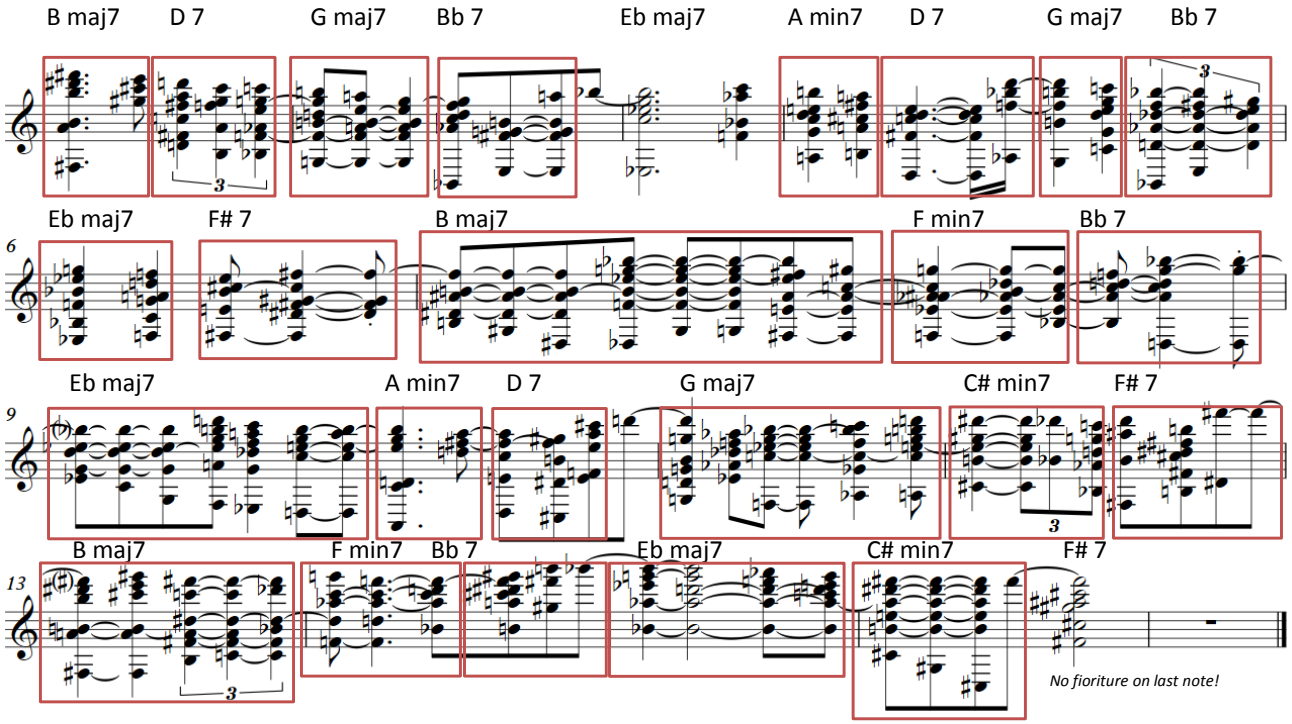


Figure 13. Giant Steps in the style of Take 6 with fioritures of various lengths. Fioritures are indicated by boxes. Note the use of new rhythms and interesting harmonies.

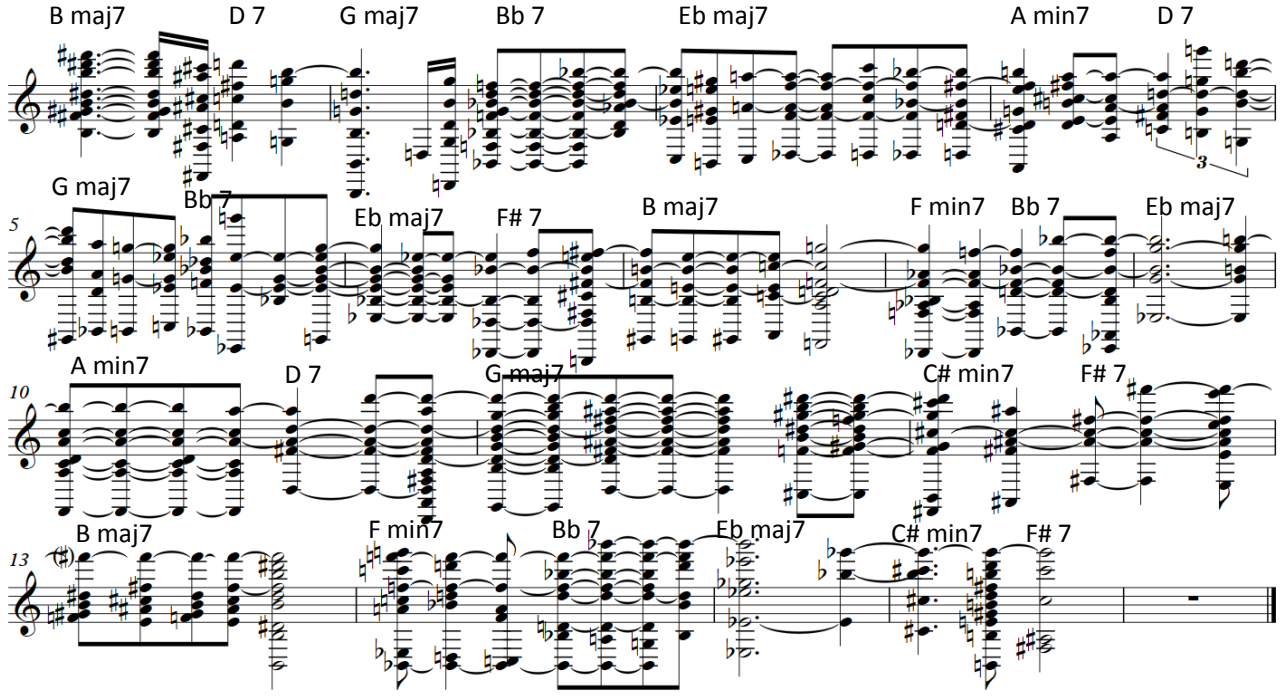


Figure 14. Giant steps with fioritures, in the style of Wagner (training corpus consists of the scores of the Ring tetralogy). The musical output definitely sounds Wagnerian yet follows strictly the Giant Steps leadsheet. Musical comments are available on the accompanying web site.

A Musical Composition Application Based on a Multiagent System to Assist Novel Composers

Maria Navarro
Computer Science Department
Salamanca University
Pza Merced, Salamanca 37005 Spain
mar90ali94@usal.es

Juan Manuel Corchado
Computer Science Department
Salamanca University
Pza Merced, Salamanca 37005 Spain
corchado@usal.es

Yves Demazeau
Laboratoire d'Informatique de Grenoble
110 avenue de la Chimie
Domaine Universitaire de Saint-Martin-dHres
BP 53 - 38041 Grenoble cedex 9 France
Yves.Demazeau@imag.fr

Abstract

This paper presents a solution to help new composers make harmonies. A multiagent approach based on virtual organizations has been used to construct this application. This model is built by using a multiagent system. This study presents a Multi-Agent System (MAS) built with PANGEA, a platform to develop different multiagent systems, capable of composing music following the HS algorithm. The results show the success of this application in correctly composing a classical harmony.

Introduction

Interest in computational creativity has been increasing in the scientific community. Although this interest is recent, there are a number of algorithms, schemas and procedures to develop an intelligent machine, capable of creating new ideas or new artistic compositions.

Many music students, or even musicians, have problems composing or improvising melodies with their own instrument. They may find it difficult to practice their improvisation or to compose their own melody because they usually need to work with other musicians who are too busy to collaborate with them. This system was designed to assist these music students in improving their abilities.

The goal of the system is to show that a simple and general agent framework such as PANGEA (Platform for Automatic coNstruction of orGanizations of intElligents Agents) (Zato and others 2012) can build a proper and scalable music composition system. A multiagent system based on virtual organizations is used because it permits making changes in the problem specification, and can modify the music style or add new rules without altering the structural composition. Only the agents behavior needs modification. The BDI architecture was chosen for these reasons.

We will evaluate the results by considering two types of criteria. First, we will consider mathematical criteria, which include an optimization function to minimize. The smaller the value in this function for one chord, the better the chord. This function considers constraint rules that evaluate the chord obtained. These rules and the evaluation method will be detailed in Section 3.

In Western music, dissonance is the quality of sounds that seem unstable and have a need to resolve to a stable sound called consonance. The definition of dissonance is culturally conditioned, which is why a classical and an occidental music culture is considered for the evaluation of consonance. According to this criterion, we can consider these consonance intervals (in order of consonance):

- Octaves
- Perfect fourths and perfect fifths
- Major thirds and minor sixths
- Minor thirds and major sixths

We will also evaluate whether the system helps composers to make their melodies or to improvise a melody by just listening to the harmonies. This evaluation consists of evaluating the system with a number from 1 to 10.

The second section contains a brief review of algorithms in music composition, Multiagent Systems and basic concepts of virtual organizations. The third section presents our model, and our particular solution, attempting to solve the problem of harmony composition with an unknown melody, and how Virtual Organizations (VO) can help to improve this system. The last section shows some results of the system, and proposes new lines of improvement.

Background

This section presents general information about composition algorithms, concepts about MAS and VO and a brief explanation about the background of agents.

Review in composition algorithms

While grammar-based systems were initially widely used in composition tasks, today there are many other algorithms attempting to compose music. Some of these are called live algorithms (Bown 2011).

One of the most successful algorithms involves Markov models (Eigenfeldt and Pasquier 2013).There are also algorithms that uses lyrics as a variable into their compositions, as for example (Monteith, Martinez, and Ventura 2012). One interesting and notable study is that of F. Pachet (Pachet 2003).

(Hoover, Szerlip, and Stanley 2011) focused on evolving a single monophonic accompaniment for a multipart MIDI by using a compositional pattern producing network (CPPN), a special type of artificial neural network (ANN).

Agents and creativity are two disciplines that have interacted in several case studies (Martin, Jin, and Bown 2011; Lacomme, Demazeau, and Dugdale 2010).

Harmony Search Algorithm Algorithm Music improvisation aims to produce an ideal state determined by aesthetic parameters, i. e., consonance or sound balance.

The procedure has five steps described here (Geem and Choi 2007). First, it is necessary to choose the optimization function and to consider a "memory" called Harmony Memory (HM, a matrix filled with as many generated solution vectors as HMS (harmony memory size)). The new harmony is generated by a random selection, a memory consideration, by using HM and a pitch adjustment (Geem and Choi 2007). The choice of one or another is conditioned by two probabilistic parameters: PAR (Pitch Adjustment Rate) and HMCR (Harmony memory Considering Rate).

Although the new harmony is built, the constraint rules that evaluate the obtained chord must also be taken into account. For this, a threshold is established. If the chord exceeds this value, it is dismissed, and the process starts again with a new chord that replaces the rejected chord.

Finally, if the new harmony vector x has a better value for the fitness function than the worst harmony in the HM, the new harmony is included in the HM. This process is repeated over and over until the stopping criterion (maximum number of improvisations) is reached.

Virtual Organizations

In the initial development of multiagent systems, the agents were seen as autonomous and dynamic entities that evolve according to their own objectives, without external explicit restrictions on their behavior and communications (Demazeau and Müller 1990). In recent years, developers have directed their interest to the organizational aspects of the society of agents (Hübner et al. 2010). Thus, two descriptive levels are set: the organization and the agent. Agents are now seen as dynamic entities that evolve within organizations.

The following sections present a description of the system, as well as the algorithm, and the agent structures used to solve the problem.

Classical Harmony Composition

Modeling musical composition is difficult because musical objects do not have any pre-assigned connotation. That means there are as many definitions of the same object as there are belief systems in musical history. For this reason, our efforts were centered on composing music from the classical period. In this period, there were many rules for composing classical music. In particular, the following main norms are considered.

- **R1** - 8th and 5th parallels: these are produced when the interval between the i-note and the j-note of the chord n and the interval between the (i+1)-note and the (j+1)-note of the chord n+1 are both 5th or 8th.
- **R2** - Leading-note resolution. There is a rule that requires a resolution of the leading-note in the tonic.
- **R3** - Voices crossing. An ideal harmony must avoid voice i getting above voice j, when j=i+1.
- **R4** - Movements between *tension*. Each chord has a peculiar role that produces stability or instability, depending on the functions (tonic, dominant and subdominant). It is the *tension* that permits the music to evolve in the composition. For this reason, our desire is to produce a movement between chords, to prevent the music from becoming boring. Thus, the repetition of the same function over time must be penalized in some way.
- **R5** - Avoid a large interval between two pitches in a chord. This is important because if we have a big pitch in the same chord, the connection between all pitches can break.
- **R6** - Avoid a large interval between two pitches in the same voice. This rule allows building more "cantabile" melodies, in general.

With all of these constraints and rules, the following optimization equation was built to minimize:

$$\sum_{i=1}^{N}\sum_{j=1}^{3} Rank(x_{ij}) + \sum_{i=1}^{N}\sum_{j=1}^{3} Penalty(x_{ij}) \quad (1)$$

Where:

$$Rank(x_{ij}) = iRank(x_{ij}, x_{i(j-1)} + \quad (2)$$

$$ln(Tension_i) + x_{ij} - x_{(i-1)j} \quad (3)$$

$Tension(x)$ values are considered with a discrete scale from 1 to 3, depending of the tension role. If the chord is Subdominant, the tension is 1, if it is dominant, tension is 3, and if it is tonic, tension would be 2. The values of $iRank(x)$ for a specific harmonic interval are:

- 3rd, 8th interval: Value of 1
- 6th interval: Value of 1.5
- 4th interval: Value of 2
- 5th interval: Value of 2.5
- Unisone interval: Value of 3
- 2nd, 7th interval: Value of 4

$Penalty(x)$ are shown in equations 4,5,6 and 7, keeping in mind the constraints considered previously.

$$x_{(i-1)j} \equiv SI \wedge x_{ij} \neq DO \Rightarrow Penalty(x_{ij}) = 5 \quad (4)$$

$$x_{i(j-1)} \geq x_{ij} \Rightarrow Penalty(x_{ij}) = 4 \quad (5)$$

$$Tension_{i-1} = 3 \wedge Tension_i = 1 \Rightarrow Penalty(x_{ij}) = 2 \quad (6)$$

$$x_{(i-1)j} - x_{(i-1)(j-1)} = x_{ij} - x_{i(j-1)} = 5 \vee 8 \quad (7)$$

$$\Rightarrow Penalty(x_{ij}) = 3 \quad (8)$$

The algorithm starts with an initialization of the Harmony Memory (HM) matrix that is stored in the repository. Several PAR and HMCR were also tested, and we chose the best ones: 0.3 to PAR and 0.2 to HMCR. In the next section, both the structure of MAS based on VO and its advantages will be explained.

Multiagent System Structure

Virtual organizations were used to implement and develop our model. Virtual organizations provide a certain number of roles easily replaceable by an agent, depending on the context. This allows the system to be very flexible. Besides, a methodology based on VO can provide us with a global vision of the problem, the model and the possible solutions.

To design the virtual organization it is necessary to analyze the needs and expectations of the system. The result of this analysis will be the roles of the entities involved in the proposed system. The following specific roles were found:

- Composer Role: This role creates the harmonic music following their rules to achieve a goal (desire).
- Evaluator Role: This role evaluates the result of the composer role and decides if it is good enough to present it to the user.
- Interface Role: This role allows the user to interact with the system.
- Data Supplier Role: This role is an agent that accesses and stores all or most of the information needed to manage the actions that govern this system.
- Control Role: The agents that exercise this role will have overall control of the system.

To implement the roles of the VO we chose to develop a MAS. For the composer and evaluator agents, we chose a BDI agent architecture (Corchado et al. 2004), for two reasons: firstly, it is the most common deliberative agent architectures, and one of the simplest; and secondly, this structure is perfectly adapted to our requirements. The BDI agent process involves two fundamental activities: a) determining which goals should be achieved (deliberation) and b) deciding how to reach these goals (planning). Both processes should be carried out by taking into account the limited resources of each agent.

The schema in Figure 1 shows how client agents are connected to model our problem.

To begin, the composer agent has as a goal or "desire" to minimize the value of the optimization function. To achieve this goal, it has to make some rules or "intentions" (that is, the algorithm), starting from its "beliefs" or its initial stage. As we can see, the BDI architecture is perfectly suited to the agent.

Additionally, the composer agent has as a "desire" to classify the chord made by the composer agent. To achieve this goal, it has to follow its "intentions", starting with its "beliefs". Finally, the remaining agents are given communication, coordination and representation tasks.

The system was developed on PANGEA (Zato and others 2012), which provides us with certain advantages. PANGEA is a service-oriented platform that allows the open multiagent system to take maximum advantage of the distribution of the resources. With PANGEA, we can change our musical agent in order to change the composition algorithm or behavior. We can even change an agent and replace it with a multiagent system capable of communicating to compose a new music. Second, we can change our Constraint Agent.

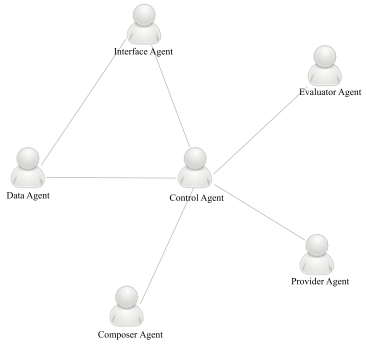


Figure 1: A global view of multiagent system interactions and communications among only client agents.

This means that different styles can be composed with this system and we only have to incorporate new behavior or update it to create jazz, rock, romantic, baroque or medieval music. We also have a database with classic styling features. The user can change these features and behaviors at any moment to permit or forbid a parallel 5th or 8th, study the leading-note resolution, etc.

Results and Conclusions

With a general framework of a MAS structure such as PANGEA, we have built a model able to compose different harmonies in order to help students new to the art of composing. However, the fitness of the results is evaluated by studying the way the rules and constraints are followed.

After the first iterations, we did not get a proper chord line, as shown in Figure 2. The first chord is perfect, taking the intervals between the notes into account. After analysing the transition between chord 1 and chord 2, we can see that the intervals are not so perfect (between Do and Re there is a 2nd interval, which is considered as dissonance). Between chord 2 and 3, the R3 is violated, as Do is becoming Mi, and the intervals again are not so perfect. Chord 4 has consonant intervals (although they might be better) but in the third voice rule R6 is violated (Sol becomes Do, and this is a little big interval.) Finally, chord 5 is better for rule R6.

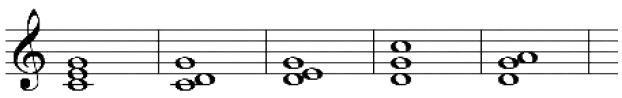


Figure 2: Harmony achieved with 45 iterations

However, the more iterations we performed, the better the results we obtained. We have a new line with 200 iterations, noticeably better than the previous one (See Figure 3). The first chord is perfect, taking the intervals between the notes

into account. Analysing chord 2, we can see that the intervals are almost as perfect as chord 1 (we have a 3rd interval and a 4th interval). Chord 3 is a chord with perfect consonance. Chord 4 has a consonant 4th interval and a dissonant interval of 2nd. Finally, chord 5 is consonant with a 3rd and 4th interval.

Rules R3, R5 and R6 are respected throughout the experiment.

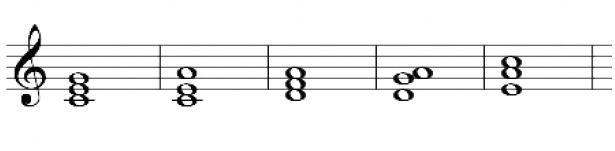


Figure 3: Harmony achieved after 200 iterations

This means that we have an evolutionary algorithm. This depends not only on the iterations we perform, but also on the parameters PAR or HMCR, which indicate the probability of making a random value for a pitch in a chord, as explained in the previous section. The fitness of the results is evaluated by studying the way the rules and constraints are followed. In other words, the more the rules are followed, the better the harmony will sound. The mathematical evaluation is to study the value of the optimization function as well as the number of the constraints that are violated.

Nevertheless in music, there is also a qualitative form to evaluate the model. This method of evaluation is based on acoustic perception, and therefore depends on the listener. We conducted tests with two experts in classical music (composers) and two non-experts in classical music to punctuate both harmonies above. The evaluation criteria was: "completely dissonant", "dissonant", "a bit consonant", "consonant", "completely consonant". The experts number 1 and number 2 evaluated the first harmony between "a bit consonant" and "dissonant", and the others evaluated as "dissonant". In the second harmony all four rated it as "consonant".

In our small study, two composers used our method and evaluated the results on a scale of 1-10. The first evaluated the result with a 6 and the second with a 7,5, which we consider as acceptable in our first approach to the system.

With regards to the virtual organization, the process of identifying and organizing roles helped to improve the management and thus to improve efficiency. The MAS structure allows us to make an extensible and scalable system as we change rules, constraints and behavior, with little effort, searching new ways of mixing different techniques, or even tools in the composition. The BDI architecture is perfectly suited for the solution we were seeking. BDI has a clear methodology that facilitates the development stage, with many theories that suit our problem. This architecture enables us to easily introduce a learning mechanism, as we can see in our case study. Moreover, using PANGEA as the platform allowed fluid communication between agents, which is evident in the design of the application, improving the modularity and the separation between client and provider as well.

As a future work, we propose incorporating rhythms. This model can also evolve to learn and self-check its own mistakes in harmony composition.

References

Bown, O. 2011. Experiments in modular design for the creative composition of live algorithms. *Computer Music Journal* 35.

Corchado, J. M.; Pavón, J.; Corchado, E. S.; and Castillo, L. F. 2004. Development of cbr-bdi agents: a tourist guide application. In *Advances in case-based reasoning*. Springer. 547–559.

Demazeau, Y., and Müller, J.-P. 1990. *Decentralized Ai*. Elsevier.

Eigenfeldt, A., and Pasquier, P. 2013. Considering vertical and horizontal context in corpus-based generative electronic dance music. In *Proceedings of the Fourth International Conference on Computational Creativity*, 72.

Geem, Z. W., and Choi, J.-Y. 2007. *Music composition using harmony search algorithm.* Springer Berlin Heidelberg. 593–600.

Hoover, A. K.; Szerlip, P. A.; and Stanley, K. O. 2011. Interactively evolving harmonies through functional scaffolding. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, 387–394. ACM.

Hübner, J. F.; Boissier, O.; Kitio, R.; and Ricci, A. 2010. Instrumenting multi-agent organisations with organisational artifacts and agents. *Autonomous Agents and Multi-Agent Systems* 20(3):369–400.

Lacomme, L.; Demazeau, Y.; and Dugdale, J. 2010. Clic: an agent-based interactive and autonomous piece of art. In *Advances in Practical Applications of Agents and Multiagent Systems*. Springer. 25–34.

Martin, A.; Jin, C. T.; and Bown, O. 2011. A toolkit for designing interactive musical agents. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, 194–197. ACM.

Monteith, K.; Martinez, T.; and Ventura, D. 2012. Automatic generation of melodic accompaniments for lyrics. In *Proceedings of the International Conference on Computational Creativity*, 87–94.

Pachet, F. 2003. The continuator: Musical interaction with style. *Journal of New Music Research* 32(3):333–341.

Zato, C., et al. 2012. Pangea–platform for automatic construction of organizations of intelligent agents. In *Distributed Computing and Artificial Intelligence*. Springer. 229–239.

Empirically Grounding the Evaluation of Creative Systems: Incorporating Interaction Design

Oliver Bown
Design Lab, University of Sydney,
NSW, 2006, Australia
oliver.bown@sydney.edu.au

Abstract

In this paper I argue that the evaluation of artificial creative systems in the direct form currently practiced is not in itself empirically well-grounded, hindering the potential for incremental development in the field. I propose an approach to evaluation that is grounded in thinking about *interaction design*, and inspired by an anthropological understanding of human creative behaviour. This requires looking at interactions between systems and humans using a richer cultural model of creativity, and the application of empirically better-grounded methodological tools that view artificial creative systems as situated in cultural contexts. The applicability of the concepts 'usability' and 'user experience' are considered for creative systems evaluation, and existing evaluation frameworks including Colton's creativity tripod and Ritchie's 18 criteria are reviewed from this perspective.

Introduction: Evaluation, Creativity and Empiricism

This paper is concerned with the evaluation of creative systems, specifically in the area of artistic creativity (not to be confused with evaluation *by* creative systems). Whilst AI researchers in other application domains are able to observe and measure incremental improvements in their algorithms, computational creativity researchers are burdened by the inherent ambiguity in the field regarding whether algorithm or system X is better than algorithm or system Y. Incremental developments in the field are also relatively obscure to the outsider: the figurative artworks created by Harold Cohen's celebrated automated artist, AARON, in the 1980s[1] *look* like the work of a competent and creative artist. As far as the artwork itself is concerned, this would appear to be as good as it gets – problem solved. But most in the field believe that we are only just beginning to develop good creative systems. Such appearances foster confusion about where we are at in the development of significant artistic creativity in computers, between a far-off goal on the one hand, and a solved problem on the other.

Cardoso, Veale, and Wiggins (2009) characterise the field as taking a pragmatic, demonstrative approach to computational creativity practice, "which sees the construction of working models as the most convincing way to drive home a point" (Cardoso, Veale, and Wiggins, 2009, p. 19). This tradition has kept the focus on innovation, distinguishing it from more theoretical studies of creativity. Nevertheless the discussions and demonstrations that surround such an approach depend on a firm relationship between empirical observations and what we claim about systems. Hence, understandably, a significant portion of the literature in the field focuses on the 'necessary theoretical distraction' of how to go about evaluating systems. Wiggins' notion of evaluation (Wiggins, 2006), widely adopted in the field, requires that a system performs tasks in a way that would be deemed creative if performed by a human. But whilst simple to state, the task of concretely drawing such a conclusion about a given system maintains an opaque and vexing relationship to the various forms of empirical observations available to us.

In light of these issues, the purpose of this paper is to examine the empirical grounding underlying the evaluation of systems. Empirical grounding is defined as the practice of anchoring theoretical terms to scientifically measurable events, and is necessary for the "effectiveness of the application of knowledge" (Goldkuhl, 2004), that is essential for transforming discussions about system designs and methods into incremental scientific progress.

I argue that whilst the essential incompatibility between evaluation in computational creativity and the objective nature of optimisation found in AI may have been acknowledged from the outset, there remains a gap that has still not yet been plugged by a positive theory of evaluation in computational creativity. Further to this, I propose that the standard model of creativity in art, derived largely from Boden's concepts, has not provided a suitable framework for thinking about where and how the evaluation of creativity applies in human artistic behaviour. To address this, it is proposed that a human-centred view, specifically the use of design-based approaches such as interaction design, can give computational creativity a thorough empirical grounding. An interaction design approach can be applied easily to existing work in computational creativity, viewing the understanding and measurement of system behaviours in terms of their interaction with human 'users'. It offers a practical route to bringing a much-needed human and social dimension to studies of creative systems without rejecting aspirations to-

[1] See AARON's online biography at http://www.usask.ca/art/digital_culture/wiebe/moving.html.

wards autonomy in computational creativity software.

The Soft Side of Computational Creativity

The adjectives 'hard' and 'soft' have been used, controversially, to refer to different areas of scientific enquiry (as a precaution, they remain in quotes throughout this paper!). Diamond (1987) explains that some "areas are given the highly flattering name of hard science, because they use the firm evidence that controlled experiments and highly accurate measurements can provide," whereas "soft sciences, as they're pejoratively termed, are more difficult to study for obvious reasons... You can't start... and stop [experiments] whenever your choose. You can't control all the variables; perhaps you can't control any variable. You may even find it hard to decide what a variable is" (Diamond, 1987, p. 35). Although many theoreticians such as Diamond reject the tone of the terms (here Diamond is arguing that soft sciences are in fact harder than hard sciences), the definitions given here usefully describe a continuum of what he understands as 'degrees of operationalisation'. Whilst the terms may connote 'tough' and 'weedy' respectively, they also connote 'rigid' and well-defined levels of operationalisation versus more 'flexible' and loosely-defined levels of operationalisation. This distinction remains useful. A key point is that there are appropriate ways to deal with 'soft' concepts, above all of which is to acknowledge them as such in order to apply suitable methods. A popular perception is that 'soft' sciences harden as their theory and practice coevolve, with psychology and sociology given as typical examples (Nature, 2005). But doing quality 'soft science' would appear to be the first step towards this ambition. Computational creativity necessarily deals with both sorts of concepts, and researchers must therefore know how to work across this spectrum.

I discuss as an example Colton's 'creativity tripod' (2008). Colton proposes to include in his formulation of evaluation a set of internal properties of systems, due to the limited information available when using only the end products of an automated creative process to evaluate that process (as advocated by Ritchie (2007)). He proposes that we look inside the system itself in order to gain a fuller description of the system's processes along with its products, and thus make a more informed decision about the creativity of the system. This, he argues, is more in line with how we evaluate human creativity:

> "A classic example... is Duchamp's displaying of a urinal as a piece of art. In situations like these, consumers are really celebrating the creativity of the artist rather then the value of the artefact" (Colton, 2008, p. 15)

Colton suggests breaking down creativity into three components – a 'creativity tripod' of skill, appreciation and imagination – that can be sought in creative systems. He defines each of these as necessary conditions for the identification of creativity, and proposes that creativity evaluation could be built around an analysis of these properties. He performs such an analysis of his own systems, *HR* and *The Painting Fool*, and identifies the existence of each component in both systems (although he clarifies that they do not occur simultaneously in the same version of the *Painting Fool* system).

In Colton's analysis, skill, appreciation and imagination are not formalised, and are treated as intuitive ideas taken in the manner of Wiggins' 'creativity as recognised by a human' criterion. Accordingly, Colton's application of the terms is impressionistic. For example, he says of *The Painting Fool*'s imagination that "we wrote a scene generation module that uses an evolutionary approach to build scenes containing objects of a similar nature, such as city skylines and flower arrangements" (Colton, 2008, p. 21). From this, the reader has little hope of determining whether the 'imagination' criterion has been satisfied, let alone what the subcriteria are for imagination. A further problem is that, in empirical terms, the expected order of knowledge discovery has clearly been put in reverse: imagination has been defined first as a kind of internal scene generation process, then implemented into the system, the conclusion being drawn that the system contains imagination. This abandons the critical step of enquiry into whether, having defined imagination as such and implemented it accordingly, this is *actually* a sufficient definition of imagination.

Under these circumstances, the concepts skill, appreciation and imagination cannot be distinguished from trivial pseudo-versions of themselves. Accordingly, reduction to triviality provides an easy rebuttal to such claims, and this has been performed by Ventura on Colton's criteria (Ventura, 2008). Ventura presents a clearly trivial, unanimously uncreative computer program, and applies a similar analysis to that performed originally by Colton, concluding that the mock system has skill, appreciation and imagination. If Venutra's system has these features, and they are sufficient for the attribution of creativity, then we must either accept the system as creative or reject the criteria as they currently stand.

Can such vague concepts be used at all, or should they dropped altogether if they can't be precisely formalised? I prefer to support both Colton's initial premise – that an understanding of the inner workings of systems is as necessary to evaluating creativity as the outputs the system produces – and his identification of skill, appreciation and imagination as critical features of advanced creative systems. They are things that we would expect to see well implemented in our finest systems and there is nothing wrong with making this intuitive step. But unfortunately they are clumsy terms, and as Ventura's analysis demonstrates, don't look like hopeful performers at a formal level. In Diamond's terms, they are far from being effectively operationalised, and they may never be operationalised, because in the process we would reasonably expect to device concepts that are far removed from folk terminology, just as physicists and neuroscientists have done.

A more generic scientific strategy for how to work with both rigid and flexible objects alike comes from the definitive hard scientist Richard Feynman (1974) who makes a simple appeal to what he describes as an unspoken law of science, "a kind of utter honesty–a kind of leaning over backwards" to face the problem of "how not to fool ourselves" (Feynman, 1974). He draws an analogy between

forms of habitual scientific practice and the famed cargo cults of the South Pacific, who carved wooden headphones and bamboo antennae in the hope of attracting cargo planes to land, imitating the troops they had seen during WWII. He calls upon scientists across disciplines to ask themselves, Am I making symbolic wooden headphones or real working headphones?

In the spirit of Feynman's call to 'utter honesty', an overlooked first step is to acknowledge that these terms, given our current knowledge, are extremely flexible and far-from-operationalised, which places very different demands on how we address and manipulate them as concepts. Their treatment is implicitly argument-based, meaning that no neat proof or direct basis in evidence is available to us. This makes for a very messy equivalence to the process of checking the steps of a proof or repeating a simulation experiment, with each step containing unknowns and vagaries: flexible rather than rigid science. Computational creativity needs to learn to work with vague concepts that are not easily subject to formal treatment.

Other examples of slips into the space of soft science that are likely to occur in computational creativity discourse include describing a system as 'doing something on its own' when discussing the autonomy of systems, but remaining imprecise about what the 'it' and the 'doing' specify (e.g., to say a program composes a piece of music 'on its own' requires quite a detailed analysis of the sequence of events leading to the specific configuration of musical content), and cases of comparing exploratory and transformational creativity in an interpretive manner (e.g., to classify any historical creative act as transformational requires the imposition of our own chosen categories onto incomplete historical data) (see Ritchie, 2006, for an interesting discussion).

For this reason 'soft sciences', such as social anthropology, subject the use of language to great scrutiny. The meaning of terms that cannot easily be made measurable or mathematically manipulable are instead treated with an acknowledgement of their fragility. As a part of their data gathering, anthropologists immerse themselves in cultural situations in order to be able to fully understand and successfully interpret what they observe. Immersion is necessary in order to expose the cultural content of these situations, which is not directly accessible through 'hard science' methods such as surveys, lab tests or recordings. For example, the difference between a twitch of the eye, a wink, a fake wink, a parodied wink, a burlesque of a parodied wink, and so on, might only be fully accessible to someone who has an intimate understanding of the sociocultural context in which the act occurs (Geertz, 1973). Misinterpretation of such acts is a clear source of error in the development of theory. In the 1980s, borrowing from philosopher Gilbert Ryle, anthropologist Clifford Geertz (Geertz, 1973) developed these practices into a method of 'thick description' that gave new impetus to, and validation of, the interpretative ('soft') side of anthropology as a science. Such thinking is more relevant to computational creativity than it may appear. The empirical material underlying Wiggins' 'creativity as recognised by a human' criterion, is in the first instance anthropological rather than psychological, revolving around interpretations of culturally-situated human behaviour: in particular that we establish a shared understanding of what 'creative' means.

Geertz' advice on grounding methodology is that "if you want to understand what a science is, you should look in the first instance not at its theories or findings, and certainly not at what its apologists say about it; you should look at what the practitioners of it do" (Geertz, 1973, p. 5). This is a call to work the science's methods around the data and practices that are practically available. This may be helpful given what computational creativity practitioners do. Cardoso, Veale and Wiggins' characterisation of computational creativity practice as the construction of "working models as the most convincing way to drive home a point" (Cardoso, Veale, and Wiggins, 2009, p. 19), breaks down into two parts: the engineering excellence to create advanced creative systems, and the analysis of human social interaction in creative contexts that will be used to round off the argument. Thus a necessary direction for computational creativity is to fuse excellence in the 'hard science' area of algorithms and the 'soft science' of understanding human social interaction. The terms skill, appreciation and imagination are things that we should be seeking to better define through ('soft') computational creativity research, and cannot at the same time be used as the basis for a ('hard') test for creativity.

Characterising Artistic Creativity Using Generative and Adaptive Creativity

Value or utility is included in the vast majority of definitions of creativity (most notably (Boden, 1990)), and is critical to many applications of creativity research, such as improving organisational creativity and building creative cities. But non-cognitive processes such as biological evolution are also viewed as creative. Here, value cannot have the same meaning as it does in the context of human cogintion-based creativity, because there is no agent to do the valuing. And yet this difference has not been explored in any depth. The application of theoretical concepts has tended to focus on Boden's (1990) two key distinctions in her analysis of creativity: between personal and historical creativity as indications of scope; and between combinatorial, exploratory and transformational creativity as forms of creative succession. From this point of view, creativity is tightly bound to individual human goals, and is primarily conceived of as a cognitive process that is used to discover new things of value.

This lack of attention to the variable nature of value in creativity causes confusion and has led to a poor empirical grounding for evaluation in computational creativity, precisely because much creativity occurs outside of the process of human creative cognition (in the narrower sense given above). A distinction based on different relations to value has not been taken up by the community. I draw on a distinction (Bown, 2012) between 'generative' and 'adaptive' creativity, and argue that this distinction clarifies and resolves the confusion about how value is manifest in the arts.

In (Bown, 2012) I propose a distinction between two forms of creativity based on their relationship to value: generative and adaptive creativity. Generative creativity is defined with a very broad scope, it occurs wherever new types

of things come into existence. It does not require cognition: non-human processes such as biological evolution are capable of creating new types of things, and, I argue, there are also examples of human activity in which things emerge 'autopoietically' without being planned or conceived of by individual humans. The role of generative creativity in art will be discussed below.

Generative creativity offers an expanded view of creativity in which the production of new types of thing is the sole criterion for creativity to have occurred, and the process by which those things are produced – whether by deities, human minds or autopoietic processes – is secondary. In human creativity, this liberates us from the possibly misleading premise that the 'creative mind' is necessary and sufficient for the 'act of creation'. A framework that distinguishes between those entities can properly address the issue of when and how human thinking is associated with new things coming into existence.

Adaptive creativity on the other hand is that in which something is created by an intelligent agent in response to a need or opportunity. The distinguishing feature here is that of value or benefit – generative creativity is 'value free'. In adaptive creativity, the agent doing the creation stands to benefit from the creative act: a link must exist between the creative agent and the beneficial return of the creative act in order for adaptive creativity to have occurred. Uncontroversial examples include solving everyday problems, such as using a coat-hanger to retrieve something from behind a wardrobe. Adaptive creativity is understood as requiring certain cognitive abilities such as mental representation, whereas generative creativity is completely blind, as in biological evolution.

Generative and adaptive creativity are not extremes at either ends of a continuum, but distinct and mutually exclusive categories – either there was a preceding purpose or there was not. However, the appearance of new things may be the sum of different episodes of generative and adaptive creativity.

Given these terms, I argue that the existing notion of the evaluation of creative systems is entirely – indeed inherently – geared towards adaptive creativity, and is unable to accommodate generative creativity at all. Adaptive creativity alone is compatible with computational creativity's AI legacy, which preferences an optimisation or search approach to discovering valuable artefacts. This is not without powerful applications. Evolutionary optimisation regularly discovers surprising designs in response to engineering problems. Thaler's "Creativity Machine", for example, was used to discover novel toothbrush designs using a relatively traditional optimisation approach involving a clear objective function (Plotkin, 2009). It is only generative creativity that is incompatible with optimisation.

Adaptive and Generative Creativity in the Arts

For the purpose of evaluating creative systems, it has been considered reasonable to assume that we can treat artistic domains entirely in terms of adaptive creativity, and that the act of creating artworks is an adaptively creative act. Accordingly one can view the production of an artwork as an optimisation or search problem. This simplification is built in to the premise of an agent designed to evaluate its output in order to find good solutions. For such an agent to incorporate generative creativity into its behaviour would mean that the value of its output was indeterminate and evaluation would be frustrated.

But evidence suggests that this view of art does not hold when one considers its social functions. I will focus on music for the purpose of this discussion, and take what I believe is an uncontroversial understanding of music insofar as sociologists of music are concerned. Hargreaves and North (1999) identify three principal social functions for music: self-identity, interpersonal relationships and mood. These in turn, they argue, shape musical preference and practice. For example, "research on the sociocultural functions of music suggests that it provides a means of defining ethnic identity" (Hargreaves and North, 1999, p. 79).

The evidence they gather shows the perceived aesthetic value of music not to be determined purely by exposure to a corpus or 'inspiring set', but also by a set of existing social relationships. More recent research in experimental psychology reveals an increasingly complex story behind how we give value to creative artefacts. Salganik, Dodds, and Watts (2006), for example, show that music ratings are directly influenced by one's perception of how others rated the music, not just in the long term but at the moment of making the evaluation. Newman and Bloom (2012) examine the underlying causes of the attachment of value to originals rather than copies, finding, amongst other things, that the value given to an original is associated with its physical contact with the artist. Both studies suggest a form of winner-takes-all process whereby success begets further success. Such phenomena place limits on the importance of the creative content in evaluation. Admittedly artistic success is not the same as artistic creativity, but the overlap is great enough, in any practical sense of evaluating creativity, to carry the argument from one domain to the other.

Csikszentmihalyi's (1999) domain-individual-field theory has long held that individuals influence domains and alter fields, but such observations have on the whole been only been acknowledged, not actually applied in computational creativity. Coming close, Charnley, Pease, and Colton (2012) present 'framing' as a way to deal with the process of adding additional information that may influence the value of a creative output. According to the idea of framing, I might provide information alongside an artwork, such as an exhibition catalogue entry, that influences its perception. In its simple form framing would embellish an artwork, perhaps explaining some hidden symbolism behind the materials used. But in this sense it is simply a part of the system output along with the artwork. By comparison, verbal statements, and other social actions, can have effects with respect to value that are categorically different from this, for example by provoking people to alter their perception of value in general. Framing takes steps towards the idea that value can be manipulated, even 'created', but continues to assume a fixed frame of reference.

Taking these additional processes into account, when an individual produces an artwork, some amount of the value

of that artwork may have already been determined by factors that are not controlled by the individual, or be later determined by factors that are unrelated to the content of the work. The creativity invested in the creation is not entirely the product of the individual, whose artistic behaviour may be more associated with habit and enculturation than discovery, but is imposed upon the individual through their context and life history. The anthropological notion of the 'dividual', or 'porous subject' (Smith, 2012) has been used capture this idea of a person as being composed of cultural influences, indicating their ongoing permeability to influence. According to this view, the flux of influence between individuals may have an equivalence to the interaction between submodules within a single brain, meaning that isolating individuals as units of study is no better a division than focusing on couples, tuples, larger groups or cognitive submodules. Given this understanding of individual human behaviour in relation to culture in general, and the arts in particular, computational creativity can be seen to place too much emphasis on the idea of individuals being independent creators.

From this alternative point of view it is argued that artistic behaviour has a significant generative creativity element by which new forms 'spring up', not because individuals think of them, but through a jumble of social interaction. Such emergent forms may have structural properties related to the process that produced them, but they were not made with purpose. By analogy, consider a classic debate about adaptationism and form in evolutionary theory: the shape of a snail shell, as described in Thompson's *On Growth and Form* (Thompson, 1992) comes about through the process of evolutionary adaptation. But this is not purely a product of the selective pressures acting on the species. It results from an interaction between selective pressures and naturally-occurring structure. Likewise, human acts of creation are constrained by structural factors that guide the creator, augmenting agency.

The notion that a system possesses a level of creativity is riddled with complexity, owing to the fact that creativity is as much something that is enacted upon individual systems as enacted by them. In computational creativity, this means that the goal of evaluating virtual autonomous artists is not empirically well-grounded when performed in isolation. Empirical grounding requires a strong coherence between our theories and practices, and the things we can observe. In the following section, I will argue that an interaction design approach delivers this coherence, bringing together system development with a thorough understanding of the culturally-situated human. I will suggest that interaction design shouldn't be viewed merely as an add-on or a form of research used only at the application stage, but that it has a central role to play in improving methodology in computational creativity.

Towards Empirical Grounding

To reiterate the argument so far, empirical grounding is defined as the process of anchoring theoretical terms to scientifically measurable events. Computational creativity characteristically employs a makers' approach to innovating new ideas and building better systems, but the idea of asking how creative these systems are is not empirically well-grounded. Then what can we ask? I have examined the need simply to elaborate on terms and concepts during the process of evaluation, adopting approparite 'soft science' ways of thinking alongside the existing engineering mindset, but although a well-grounded approach needs to take this into account, it does not provide a grounding itself.

Two research methodologies already well integrated into computational creativity offer a basis for empirically well-grounded research. These are interaction design and multi-agent systems modelling. In both cases the imbalance between generative creativity and adaptive creativity is addressed. In the interaction design approach, creative systems are treated as objects that are inevitably situated in interaction with humans. The nature of that interaction, including its efficacy, is treated as the primary concern. Here the empirical grounding comes from the fact that properties of interaction and experience related to the analysis of usability and user experience can be observed and measured, whilst existing notions of creativity evaluation can easily be incorporated into theories of interaction design. This need not be limited to a creative professional working with a piece of creative software, but could apply to any form of interaction between person and creative system. In the modelling approach, artificial creative systems are treated as models of human creative systems. For the reasons discussed above, it does not suffice to test the success of model systems by attempting to evaluate their output, but many other observable and measurable aspects of human creativity can be studied. Multi-agent models of social networks are particularly appealing in this regard because generatively creative processes fall inside the scope of the system being studied, alleviating the tension between adaptive and generative creativity.

In this paper I only elaborate on the interaction design approach, firstly because it is more immediately applicable to computational creativity practice, and secondly because much of what can be said about empirically grounded modelling is well-known to researchers.

Interaction Design

Discussions of humans evaluating machines are commonplace in the computational creativity literature. But a lot less attention is paid to the wider range of ways in which humans can interact with creative systems. The word 'interaction', applied in the context of humans interacting with creative systems, was only used in three out of 41 papers in the 2013 ICCC proceedings (and six papers out of 46 in 2012).

Interaction design is a large field of research and is not presented in any depth here (a good introduction is the textbook by Rogers, Preece, and Sharp (2007)). The following discussion considers computational creativity in light of some core topics from the field, and looks beyond to how a study of interaction in its widest sense could be usefully applied to computational creativity.

A number of computational creativity studies are already explicitly user-focused owing to their specific research goals. For example, DiPaola et al. (2013) examined the use

of evolutionary design software in the hands of professional designers, looking at usability through the integration with the creative process, and ultimate creative productivity.

A human-centred approach to the evaluation of creative systems shifts the nature of the enquiry very slightly, by asking not how creative a system is, or whether it is creative by some measure, but how its creative potential is practically manifest in interactions with people. However, this does not require researchers to repurpose their systems as tools for artists, designers or end users, or abandon the goal of automating creativity, but to take a pluralistic approach to the application of creativity as something that is realised through interaction. As addressed in the work of DiPaola et al. (2013), described above, an obvious instance is to look at usability in the case of creativity support tools.

This is the classical locus of interaction between interaction design and computational creativity. But even researchers working towards fully autonomous 'artificial artists' are building systems that will ultimately interact with people, albeit in non-standard ways. Examples include artists such as Paul Brown (Brown, 2009), who has wrestled with the notion of maximising the agency of a system to the exclusion of the human artists' signature. As the discussions surrounding such system design shows, there is no shortage of interaction between systems and the social worlds they inhabit, any of which can be considered a source of rich data.

Beyond usability a key concept in interaction design is 'user experience' (Hassenzahl and Tractinsky, 2006). User experience looks beyond efficacy with respect to function to consider a host of subjective qualities to do with interaction more generally, such as desirability, credibility, satisfaction, accessibility, boredom and so on (Rogers, Preece, and Sharp, 2007). Analysis of user experience includes understanding users' desires, expectations and assumptions, and their overall conceptual model of the system. These diverse and quite vague concepts in user experience are arguably of greater importance than usability in a wide number of circumstances, and can also be at odds with it. For example, in game development pleasure can be seen to be contrary to usability (Rogers, Preece, and Sharp, 2007): dysfunctional ways of doing things, as embodied in interface design choices, may be more fun than more functional choices. By comparison, computational creativity need not be reduced to issues of function.

Such analytical concepts present a striking match with the most ambitious goals of computational creativity. Returning to Wiggins' definition, it would not be surprising to find that a human's appraisal of machine creativity is subject to a complex of user-experience design factors. Concepts such as surprise are already established in computational creativity theory, whereas other notions, such as the role of music and art in the development of social identification, are not, but may form part of the design of a successful 'computational creativity' experience.

To acknowledge and make explicit the design component in creating autonomous systems may help remove the perceived paradox that the system is an autonomous agent supposedly independent of its creators, by examining what 'designed autonomy' would actually mean.

Often successful computationally creative systems involve some kind of puppetry, such as the subtleties of 'fine tuning' described by Colton, Pease, and Ritchie (2001). Many working in this area have embraced the idea of creative software either as a tool, a collaborator that is not capable of full autonomy, or as as creative *domain* in its own right. In these cases the interaction between human artist and software agent is treated as a persevering and explicitly acknowledged state of affairs, rather than as a temporary stop on the way to fully autonomous creative systems.

In such cases it is again fruitful to think of the relationship between the developer/artist and the system in terms of usability, even if the working interface is simply a programming environment. Such a view may lead to better knowledge about effective development practices that in turn speed up the creation of more impressive creative systems. Accepting the role of developers and artists also enables a better grasp of the attribution of authorship and agency, asking instead a question of degree – how much and in what way the system contributed to the creative outputs – rendering unimportant the ideal of 'full autonomy'.

From Evaluating System Creativity to Analysing Situated Creativity

Taking an interaction design approach reveals a wealth of empirically grounded questions that can be asked about creative systems without changing the basic designs and objectives of practitioners, and without an overly narrow focus on the question of how creative the system is.

But in order not to throw out the baby with the bathwater, since our interest is in systems that act creatively, then the creativity of systems must remain the focus of an interaction design approach. We require enriched ways to question the nature of creative efficacy and creative agency in systems. For example, an interaction design approach can better frame our evaluation of the issue of the software's autonomy, which might otherwise be occluded.

A number of existing approaches to evaluation already give ample space for domain-specific and application-dependent variation in their use, but do not go so far as to preference design and interaction studies over direct evaluation in computational creativity. Jordanous' (2011) proposal for creativity evaluation measures that are domain-specific suggests a design approach which is targeted at specific user-groups and specific needs, rather than an objective notion of what creativity is. A number of other researchers have proposed objective or semi-objective (depending on human responses) measures that are associated with creativity (they are not necessarily measures *of* creativity). Kowaliw, Dorin, and McCormack (2012), for example, compare formal definitions of creativity, written into a system, with human evaluations, so as to examine the accuracy of these definitions.

One of the most widely applied and discussed examples is Ritchie's (2001; 2007) set of criteria. Ritchie proposes 18 criteria for "attributing creativity to a computer program". The criteria derive from two core pieces of information that apply wherever a machine produces creative outputs: the inspiring set I (the input to the system) and the system's output R. An evaluation scheme (often multi-person surveys

in the implementations examined by Ritchie) is then used to form two key measures for each output in R: *typicality* is a measure of how typical the output is of the kind of artefact being produced; *quality* is a measure of the perceived or otherwise computed quality of that artefact. From these scores, Ritchie organises the outputs into sets according to whether they fall into given ranges of typicality and quality. These sets are then applied in various ways in the calculation of the resulting Boolean criteria. For example, criterion 5 states that the number of outputs that are both high-quality and typical, divided by the number of outputs that are just typical, is greater than some given threshold (this, plus the thresholds required to determine the 'high-quality' and 'typical' sets, are left to the implementer to specify). As with all of Ritchie's criteria, criterion 5 corresponds to a natural usage of the term creativity, in this case that a system whose set of typical outputs rarely includes valuable outputs is in some sense creatively lacking.

One practical problem with Ritchie's criteria, as illustrated by the examples of their application to creative systems reported in (Ritchie, 2007), is the difficulty with which implementers establish their evaluation scheme. For example, Pereira et al. (2005) measure typicality based on closeness to I, calculated using edit distance. The appropriateness of this choice is hard to determine. Others use human responses to surveys, providing a form of empirical grounding. But such surveys may have wide variance, and the formulations of the criteria have no way of incorporating variance, which would represent a more complex model of the social system in which the creative agent operates. This belies the fact that typicality is a slippery, 'soft science' concept in reality and its relationship to a measure of quality more so, despite the clarity of Ritchie's mathematics. Thus, as with Colton's tripod, Ventura (2008) points to shortcomings in the criteria by showing that trivial programs can reveal instances of inherent insufficiency in their outcomes when compared with intuitive analysis of the same systems. The underlying problem is that of how to empirically ground the choice of evaluation scheme itself, such that it might provide an empirical grounding for the criteria, suggesting that the mathematics has simply shifted the hard problem from one place to another. The best we can do is to see how the various evaluation schemes and criteria relate in practice to other observables, thus the critical point: using human responses about creativity or related features of a system, alone, does not itself provide an empirical grounding for understanding the system, but rather a data point about the wider interaction. Further studies of behaviour are required to empirically ground our understanding of what these human responses mean.

A related issue in the discussion surrounding Ritchie's criteria, is what to do with the results obtained. The criteria have, in Ritchie's view, often been misunderstood as some sort of multivariate test for creativity. Confusingly, Ritchie unintentionally encourages this misunderstanding in his description of them as "criteria for attributing creativity to a computer program" (Ritchie, 2007). In fact he cautions against their direct use in this way.

Thus the criteria offer different analytical windows onto the creative nature of systems. We are invited to preference some criteria over others, but given no advice on how to. However, from the point of view of interaction design, such ambiguity is expected and desirable. In application, we may value systems that are good at producing a high ratio of quality to overall output, or typicality to overall output, or quality within the typical set. Alternatively, other approaches to creativity may suggest counter-intuitive additions or alterations to Ritchie's criteria, such as novelty search (e.g., Lehman and Stanley, 2011), which attempts to chart an output space by relentlessly searching for atypicality. The result of this is a broad representative spread of prototypes, not a concentration of high-value or typical outputs, so would score low on many of Ritchie's criteria but may prove to be the basis for powerful automated creativity. An interaction design approach is implicit in Ritchie's treatment of systems as tools. For example, in defining typicality, he refers to the system as having a job to do "producing artefacts of the required sort" (Ritchie, 2007, p. 73). This is not, on reflection, a requirement associated with being creative, but with performing some function required by the user or designer.

With this in mind, is it possible that the final step of "attributing creativity to a computer program" has caused more confusion than clarity, and should be quietly dropped? I suggest that it should and, echoing Jordanous (2011), that the criteria are better suited to specific creative scenarios. A jingle composer may preference typicality and require only an average degree of value, whereas an experimental artist may have little or no interest in typicality but is willing to hold out for rare instances of exceptional value. Both have different time-demands, resources, goals, aesthetic preferences and notions of the role of creativity in their work. It would not be unusual to view the experimental artist as the more creative of the two, but this is clearly only an assumption given our present theoretical understanding of creativity. The same applies to end users. Even a consumer may want typicality sometimes, and extraordinary experiences at other times.

Thus the problems raised concerning Ritchie's criteria and their application are very easily addressed by taking a human-centred view of creative systems. Applications in the domain of both generative and adaptive creativity can be devised, and examination of the creative behaviour of systems can then be empirically well-grounded in the methods of interaction design.

Conclusion

The main argument of this paper is that the evaluation of systems as it is currently typically conceived in the computational creativity literature is not in itself empirically well-grounded. The data provided by performing human evaluations should instead be understood as one potential source of information that can feed into studies of the interaction between creative systems and people in order to be well-grounded. Systems may only be understood as creative by looking at their interaction with humans using appropriate methodological tools. A suitable methodology would include, (i) the recognition and rigorous application of 'soft science' methods wherever vague unoperationalised terms

and interpretative language is used, and (ii) an appropriate model of creativity in culture and art that includes the recognition of humans as 'porous subjects', and the significant role played by generative creativity in the dynamics of artistic behaviour. For the time being at least, terms such as 'creativity' and 'imagination' do not describe things that we can readily measure or objectively identify, they are concepts that frame other kinds of measurable and objectively identifiable things, as part of a loose theoretical framework.

References

Boden, M. 1990. *The Creative Mind*. George Weidenfeld and Nicholson Ltd.

Bown, O. 2012. Generative and adaptive creativity. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*. Berlin: Springer. 361–381.

Brown, P. 2009. Autonomy, signature and creativity. In McCormack, J., and d'Inverno, M., eds., *Dagstuhl Seminar Proceedings 09291: Computational Creativity: An Interdisciplinary Approach*, 1–7.

Cardoso, A.; Veale, T.; and Wiggins, G. A. 2009. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine* 30(3):15.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In *Proceedings of the Third International Conference on Computational Creativity*, 77–82.

Colton, S.; Pease, A.; and Ritchie, G. 2001. The effect of input knowledge on creativity. In *Case-based reasoning: Papers from the workshop programme at ICCBR*, volume 1.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, 14–20.

Csikszentmihalyi, M. 1999. Implications of a systems perspective for the study of creativity. In Sternberg, R. J., ed., *The Handbook of Creativity*. New York: Cambridge University Press. 313–335.

Diamond, J. 1987. Soft sciences are often harder than hard sciences. *Discover* 8(8):34–39.

DiPaola, S.; McCaig, G.; Carlson, K.; Salevati, S.; and Sorenson, N. 2013. Adaptation of an autonomous creative evolutionary system for real-world design application based on creative cognition. In *Proceedings of the Fourth International Conference on Computational Creativity*, 40–48.

Feynman, R. 1974. Cargo cult science. Available from http://neurotheory.columbia.edu/ken/cargo_cult.html.

Geertz, C. 1973. *The Interpretation of Cultures*. New York: Basic Books.

Goldkuhl, G. 2004. Design theories in information systems-a need for multi-grounding. *Journal of Information Technology Theory and Application (JITTA)* 6(2):7.

Hargreaves, D. J., and North, A. C. 1999. The functions of music in everyday life: Redefining the social in music psychology. *Psychology of Music* 27(1):71–83.

Hassenzahl, M., and Tractinsky, N. 2006. User experience-a research agenda. *Behaviour & Information Technology* 25(2):91–97.

Jordanous, A. 2011. Evaluating evaluation: Assessing progress in computational creativity research. In *Proceedings of the second international conference on computational creativity (ICCC-11). Mexico City, Mexico*, 102–107.

Kowaliw, T.; Dorin, A.; and McCormack, J. 2012. Promoting creative design in interactive evolutionary computation. *IEEE transactions on evolutionary computation* 16(4):523.

Lehman, J., and Stanley, K. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation* 19(2):189–223.

Nature. 2005. In praise of soft science. *Nature* 6(23):1003.

Newman, G. E., and Bloom, P. 2012. Art and authenticity: The importance of originals in judgments of value. *Journal of Experimental Psychology: General* 141(3):558.

Pereira, F. C.; Mendes, M.; Gervás, P.; and Cardoso, A. 2005. Experiments with assessment of creative systems: an application of Ritchie's criteria. In *Proceedings of the workshop on computational creativity, 19th international joint conference on artificial intelligence*.

Plotkin, R. 2009. *The genie in the machine: how computer-automated inventing is revolutionizing law and business*. Stanford University Press.

Ritchie, G. 2001. Assessing creativity. In Wiggins, G. A., ed., *Proc. of AISB'01 Symposium*.

Ritchie, G. 2006. The transformational creativity hypothesis. *New Generation Computing* 24(3):241–266.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Rogers, Y.; Preece, J.; and Sharp, H. 2007. Interaction design.

Salganik, M. J.; Dodds, P. S.; and Watts, D. J. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *science* 311(5762):854–856.

Smith, K. 2012. From dividual and individual selves to porous subjects. *The Australian Journal of Anthropology* 23(1):50–64.

Thompson, D. W. 1992. *On Growth and Form*. Dover.

Ventura, D. 2008. A reductio ad absurdum experiment in sufficiency for evaluating (computational) creative systems. In *Proceedings of the 5th international joint workshop on computational creativity. Madrid, Spain*, 11–19.

Wiggins, G. A. 2006. Searching for computational creativity. *New Generation Computing* 24(3):209–222.

What to expect when you're expecting: The role of unexpectedness in computationally evaluating creativity

Kazjon Grace and Mary Lou Maher
{k.grace,m.maher}@uncc.edu
The University of North Carolina at Charlotte

Abstract

Novelty, surprise and transformation of the domain have each been raised – alone or in combination – as accompaniments to value in the determination of creativity. Spirited debate has surrounded the role of each factor and their relationships to each other. This paper suggests a way by which these three notions can be compared and contrasted within a single conceptual framework, by describing each as a kind of *unexpectedness*. Using this framing we argue that current computational models of novelty, concerned primarily with the *originality* of an artefact, are insufficiently broad to capture creativity, and that other kinds of expectation – whatever the terminology used to refer to them – should also be considered. We develop a typology of expectations relevant to computational creativity evaluation and, through it describe a series of situations where expectations would be essential to the characterisation of creativity.

Introduction

The field of computational creativity, perhaps like all emergent disciplines, has been characterised throughout its existence by divergent, competing theoretical frameworks. The core contention – unsurprisingly – surrounds the nature of creativity itself. A spirited debate has coloured the last several years' conferences concerning the role of *surprise* in computational models of creativity evaluation. Feyerabend (1963) argued that scientific disciplines will by their nature develop incompatible theories, and that this theoretical pluralism beneficially encourages introspection, competition and defensibility. We do not go so far as to suggest epistemological anarchy as the answer, but in that pluralistic mindset this paper seeks to reframe the debate, not quell it.

We present a way by which three divergent perspectives on the creativity of artefacts can be placed into a unifying context[1]. The three perspectives on evaluating creativity are that, in addition to being valuable, 1) creative artefacts are novel, 2) creative artefacts are surprising, or 3) creative artefacts transform the domain in which they reside. We propose that these approaches can be reconceptualised to all derive from the notion of *expectation*, and thus be situated within a framework illustrating their commonalities and differences.

Creativity has often been referred to as the union of novelty and value, an operationalisation first articulated (at least to the authors' knowledge) in Newell, Shaw, and Simon (1959). Computational models of novelty (eg. Berlyne, 1966, 1970; Bishop, 1994; Saunders and Gero, 2001b) have been developed to measure the *originality* of an artefact relative to what has come before. Newell and others (eg. Abra, 1988) describe novelty as necessary but insufficient for creativity, forming one half of the novelty/value dyad.

Two additional criteria have been offered as an extension of that dyad: surprisingness and transformational creativity. Surprise has been suggested as a critical part of computational creativity evaluation because computational models of novelty do not capture the interdependency and temporality of experiencing creativity (Macedo and Cardoso, 2001; Maher, 2010; Maher and Fisher, 2012), but has also been considered unnecessary in creativity evaluation because it is merely an observer's response to experiencing novelty (Wiggins, 2006b). Boden's transformational creativity (Boden, 2003) (operationalised in Wiggins, 2006a) has been offered as an alternative by which creativity may be recognised. In both cases the addition is motived by the insufficiency of originality – the comparison of an artefact to other artefacts within the same domain – as the sole accompaniment to value in the judgement of creativity.

Thus far these three notions – novelty, surprise and transformativity – have been considered largely incomparable, describing different parts of what makes up creativity. There has been some abstract exploration of connections between the two – such as Boden's (2003) connection of "fundamental" novelty to transformative creativity – but no concrete unifying framework. This paper seeks to establish that there is a common thread amongst these opposing camps: *expectations* play a role in not just surprise but novelty and transformativity as well.

The foundation of our conceptual reframing is that the notions can be reframed thusly:

- *Novelty* can be reconceptualised as occurring when an observer's expectations about the continuity of a domain are violated.
- *Surprise* occurs in response to the violation of a confident expectation.
- *Transformational creativity* occurs as a collective reaction to an observation that was unexpected to participants in a domain.

We will expand on these definitions through this paper. Through this reframing we argue that *unexpectedness is involved in novelty, surprise and domain transformation, and is thus a vital component of computational creativity evaluation.* The matter of where in our field's pluralistic and still-emerging theoretical underpinnings the notion of unexpectedness should reside is – for now – one of terminology

[1] Creative processes are another matter entirely, one beyond the scope of this paper.

alone. This paper sidesteps the issue of whether expectation should primarily be considered the stimulus for surprise, a component of novelty, or a catalyst for transformative creativity. We discuss the connections between the three notions, describe the role of expectation in each, and present an exploratory typology of the ways unexpectedness can be involved in creativity evaluation.

We do not seek to state that novelty and transformativity should be subsumed within the notion of surprise due to their nature as expectation-based processes. Instead we argue that the notions of novelty, surprise and transformativity are all related by another process – expectation – the role of which we yet know little. We as a field have been grasping at the trunk and tail of the proverbial poorly-lit pachyderm, and we suggest that expectation might let us better face the beast.

The eye of the beholder

Placing expectation at the centre of computational creativity evaluation involves a fundamental shift away from comparing artefacts to artefacts. Modelling unexpectedness involves comparing the reactions of observers of those artefacts to the reactions of other observers. This reimagines what makes a creative artefact different, focussing not on objective comparisons but on subjective perceptions. This "eye of the beholder" approach framing is compatible with formulations of creativity that focus not on artefacts but on their artificers and the society and cultures they inhabit (Csikszentmihalyi, 1988). It should be noted that no assumptions are made about the nature of the observing agent – it may be the artefact's creator or not, it may be a participant in the domain or not, and it may be human or artificial.

The observer-centric view of creativity permits a much richer notion of what makes an artefact different: it might relate to the subversion of established power structures (Florida, 2012), the destruction of established processes (Schumpeter, 1942), or the transgression of established rules (Dudek, 1993; Strzalecki, 2000). These kinds of cultural impacts are as much part of an artefact's creativity as its literal originality, and we focus on expectation as an early step towards their computationally realisation.

The notion of transformational creativity (Boden, 2003) partially addresses this need by the assumption that cultural knowledge is embedded in the definition of the conceptual space, but to begin computationally capturing these notions in our models of evaluation we must be aware of how narrowly we define our conceptual spaces. The notion common to each of subversion, destruction and transgression is that *expectations about the artefact are socio-culturally grounded*. In other words, we must consider not just how an artefact is described, but its place in the complex network of past experiences that have shaped the observing agent's perception of the creative domain. A creative artefact is *unexpected* relative to the rules of the creative domain in which it resides. To unravel these notions and permit their operationalisation in computational creativity evaluation we focus not on novelty, surprise or transformativity alone but on the element common to them all: *the violation of an observer's expectations*.

Novelty as expectation

Runco (2010) documents multiple definitions of creativity that give novelty a central focus, and notes that it is one of the only aspects used to define creativity that has been widely adopted. Models of novelty, unlike models of surprise, are not typically conceived of as requiring expectation. We argue that novelty can be described using the mechanism of expectation, and that doing so is illuminative when comparing novelty to other proposed factors.

Novelty can be considered to be expectation-based if the knowledge structures acquired to evaluate novelty are thought of as a model with which the system attempts to predict the world. While these structures (typically acquired via some kind of online unsupervised learning system) are not being built for the purpose of prediction, they represent assumptions about how the underlying domain can be organised. Applying those models to future observations within the domain is akin to expecting that those assumptions about domain organisation will continue to hold, and that observations in the future can be described using knowledge gained from observations in the past. The *expectation of continuity* is the theoretical underpinning of computational novelty evaluation, and can be considered the simplest possible creativity-relevant expectation.

Within the literature the lines between novelty and surprise are not always clear-cut, a conflation we see as evidence of the underlying role of expectation in both. Novelty in the Creative Product Semantic Scale (O'Quin and Besemer, 1989), a creativity measurement index developed in cognitive psychology, is defined as the union of "originality" and "unexpectedness". The model of interestingness in Silberschatz and Tuzhilin (1995) is based on improbability with respect to confidently held beliefs. The model of novelty in Schmidhuber (2010) is based on the impact of observations on a predictive model, which some computational creativity researchers would label a model of transformativity, while others would label a model of surprise. Each of these definitions suggests a complex relationship that goes beyond the notion of originality as captured by simple artefact-to-artefact comparisons.

Surprise as expectation

Many models of surprise involve the observation of unexpected events (Ortony and Partridge, 1987). In our previous work we give a definition of surprise as *the violation of a confidently-held expectation* (Maher and Fisher, 2012; Grace et al., 2014a), a definition derived from earlier computational models both within the domain of creativity (Macedo and Cardoso, 2001) and elsewhere (Ortony and Partridge, 1987; Peters, 1998; Horvitz et al., 2012; Itti and Baldi, 2005).

Models of surprise have previously looked at a variety of different kinds of expectation: predicting trends within a domain (Maher and Fisher, 2012), predicting the class of an artefact from its features (Macedo and Cardoso, 2001) or the effect on the data structures of a system when exposed to a new piece of information (Baldi and Itti, 2010). The first case concerns predicting attributes over time, and involves an expectation of continuity of trends within data, the second case concerns predicting attributes relative to a classification, and is an expectation of continuity of the relationships within data, and the third case concerns the size of the change in a predictive mechanism, and is based on an expectation of continuity, but measured by the post-observation change rather than the prediction error. In each of these cases it is clear that a related but distinct expectation is central to the judgement of surprisingness, but as of yet no comprehensive typology of the kinds of expectation relevant to creativity evaluation exists. The *expectations of continuity* that typically make up novelty evaluation can be extended to cover the above cases This paper investigates the kinds of expectation that are relevant to creativity evaluation independent of whether they

are an operationalisation of surprise or some other notion.

Transformativity as expectation

Boden's *transformational creativity* can be reconceptualised as unexpectedness. We develop a notion of transformativity grounded in *an observer's expectations that their predictive model of a creative domain is accurate.* This requires a reformulation of transformation to be *subjective* to an observer – Boden wrote of the transformation of a domain, but we are concerned with the transformation of an observer's knowledge about a domain. To demonstrate the role of expectation in this subjective transformativity, we consider the operationalisation of Boden's transformative creativity proposed by Wiggins (2006b,a), and extend it to the context of two creative systems rather than one.

One system, the *creator*, produces an artefact and chooses to share it with the second creative system, the *critic*. For the purposes of this discussion we investigate how the critic evaluates the object and judges it transformative. In Wiggins' formalisation the conceptual space is defined by two sets of rules: $\mathscr{R}$, the set of rules that define the boundaries of the conceptual space, and $\mathscr{T}$, the set of rules that define the traversal strategy for that space. Wiggins uses this distinction to separate Boden's notion of transformational creativity into *$\mathscr{R}$-transformational*, occurring when a creative system's rules for bounding a creative domain's conceptual space are changed, and *$\mathscr{T}$-transformational*, occurring when a creative system's rules for searching a creative domain's conceptual space are changed.In the case of our critic it is the set $\mathscr{R}$ that we are concerned with – the critic does not traverse the conceptual space to generate new designs, it evaluates the designs of the creator.

Once we assume the presence of more than one creative agent then $\mathscr{R}$, the set of rules bounding the conceptual space, cannot be ontological in nature – it cannot be immediately and psychically shared between all creative systems present whenever changes occur. $\mathscr{R}$ must be mutable to permit transformation and individual to permit situations where *critic* and *creator* have divergent notions of the domain. Divergence is not an unusual case: If a transformational artefact is produced by *creator* and judged *$\mathscr{R}$-transformational* by it, and then shared with *critic*, there must by necessity be a period between the two evaluations where the two systems have divergent $\mathscr{R}$ – even with only two systems that share all designs. With more systems present, or when creative systems only share selectively, divergence will be greater. To whom, then, is such creativity transformational?

To reflect the differing sets belonging to the two agents we refer to $\mathscr{R}$ as it applies to the two agents as $critic_{\mathscr{R}}$ and $creator_{\mathscr{R}}$. If a new artefact causes a change in $critic_{\mathscr{R}}$, then we refer to it as *$critic_{\mathscr{R}}$-transformational*. This extends Boden's distinction between P- and H-creativity: A creative system observing a new artefact (whether or not it was that artefact's creator) can change only its own $\mathscr{R}$, and thus can exhibit only P-transformativity. We distinguish "P-transformativity" from "P-creativity" to permit the inclusion of other necessary qualities in the judgement of the latter: novelty, value, etc.

We can now examine the events that lead *critic* to judge a new artefact to be *$critic_{\mathscr{R}}$-transformational*. The rules that make up $critic_{\mathscr{R}}$ cannot have been prescribed, they must have developed over time, changing in response to the perception of P-transformational objects. The rules that make up Wiggins' set $\mathscr{R}$ must be inferred from the creative system's past experiences. The rules in $critic_{\mathscr{R}}$ cannot be descriptions of the domain as it exists independently of the critic system, they are merely *critic*'s current best guess at the state of the domain. The rules in $\mathscr{R}$ are learned estimates that make up a *predictive model* of the domain – they can only be what the creative system *critic expects* the domain to be.

A kind of expectation, therefore, lies at the heart of both the transformational and the surprise criteria for creativity. The two approaches both concern the *un*-expectedness of an artefact. They differ, however, in how creativity is measured with respect to that unexpectedness. Transformational creativity occurs when a creative system's expectations about the boundaries of the domain's conceptual space – Wiggins' $\mathscr{R}$ – are updated in response to observing an artefact that broke those boundaries. Surprisingness occurs when a creative system's expectations are violated in response to observing an artefact. Transformation, then, occurs in response to surprisingness, but both can occur in the same situations. This is not to say that all expectations are alike: "surprise" as construed by various authors as a creativity measure has involved a variety of kinds of expectation. The purpose of this comparison is to demonstrate that there is a common process between the two approaches, and we suggest that this commonality offers a pathway for future research.

From individual to societal transformativity

A remaining question concerns the nature of H-transformativity in a framework that considers all conceptual spaces to be personal predictive models. This must be addressed for an expectation-based approach to model transformation at the domain level – that which Boden originally proposed. If all $\mathscr{R}$ and transformations thereof occur within a single creative system, then where does the "domain" as a shared entity reside? Modelling creativity as a social system (Csikszentmihalyi, 1988) is one way to answer that question, with the notion that creativity resides in the interactions of a society – between the creators, their creations and the culture of that society. This approach argues that the shared domain arises emergently out of the interactions of the society (Saunders and Gero, 2001b; Sosa and Gero, 2005; Saunders, 2012), and that it is communicated through the language and culture of that society. The effect of this is that overall "historical" creativity can be computationally measured, but only if some bounds are placed on history. Specifically, the transformativity of an artefact can be investigated with respect to the history of a defined society, not all of humanity.

One approach to operationalising this socially-derived H-creativity would be through a multi-agent systems metaphor: for an artefact to be judged H-creative it would need to receive a P-creative judgement from a majority of the pool of influence within the society, assuming that each agent possesses personal processes for judging the creativity of artefacts and the influentialness of other creative agents. This very simple formalisation does not model any of the influences discussed in Jennings (2010), but is intended to demonstrate how it would be possible to arrive at H-transformativity within a society given only P-transformativity within individual agents.

A framework for creative unexpectedness

The notion of expectation needs to be made more concrete if it is to be the basis of models of creativity evaluation. We develop a framework for the kinds of expectation that are

relevant to creativity evaluation, and situate some prior creativity evaluation models within that framework. The framework is designed to describe *what to expect* when modelling expectation for creativity. The framework is based on six dichotomies, an answer to each of which categorises the subject of an expectation relevant to the creativity of an artefact. These six questions are not intended to be exhaustive, but they serve as a starting point for exploration of the issue.

First we standardise a terminology for describing expectations:

- The *predicted* property is what is being expected, the dependent variable(s) of the artefact's description. For example, in the expectation "it will fit in the palm of your hand" the size of artefact is the predicted property.
- The *prediction* property is the information about the predicted, such as a range of values or distribution over values that is expected to be taken by artefacts. For example, in the expectation "the height will be between two and five metres" the prediction is the range of expected length values.
- The *scope* property defines the set of possible artefacts to which the expectations apply. This may be the whole domain or some subset, for example "luxury cars will be comfortable".
- The *condition* property is used to construct expectations that predict a relationship between attributes, rather than predict an attribute directly. These expectations are contingent on a relationship between the predicted property and some other property of the object – the condition. For example, the expectation "width will be approximately twice length" predicts a relationship between those two attributes in which the independent variable length affects the dependent variable width. In other expectations the prediction is unconditional and applies to artefacts regardless of their other properties.
- The *congruence* property is the measure of fit between an expectation and an observation about which it makes a prediction – a low congruence with the expectation creates a high unexpectedness and indicates a potentially creative artefact. Examples of congruence measures include proximity (in attribute space) and likelihood.

Using this terminology an expectation makes a *prediction* about the *predicted* given a *condition* that applies within a *scope*. An observation that falls within that scope is then measured for *congruence* with respect to that expectation. The six dichotomies of the framework categorise creativity-relevant expectations based on these five properties.

1. Holistic vs. reductionist

Expectations can be described as either *holistic*, where what is being predicted is the whole artefact, or *reductionist*, where the expectation only concerns some subset of features within the artefact. Holistic expectations make predictions in aggregate, while reductionist expectations make predictions about one or more attributes of an artefact, but less than the whole.

An example of a holistic expectation is "I expect that new mobile phones will be similar to the ones I've seen before". This kind of expectation makes a prediction about the properties of an artefact belonging to the creative domain in which the creative system applies. The attribute(s) of all artefacts released within that domain will be constrained by that prediction. In this case what is being *predicted* is the whole artefact and the *prediction* is that it will occupy a region of conceptual space. The *scope* is all possible artefacts within the creative domain of the system. The *congruence* measure calculates distance in the conceptual space.

This kind of expectation is typically at the heart of many computational novelty detectors – previously experienced artefacts cause a system to expect future artefacts to be similar within a conceptual space. One example is the Self-Organising Map based novelty detector of (Saunders and Gero, 2001a), where what is being predicted is the whole artefact, the scope is the complete domain, the prediction is a hyperplane mapped to the space of possible designs, and the congruence is the distance between a newly observed design and that hyperplane.

An example of a reductionist expectation is "I expect that new mobile phones will not be thinner than ones I've seen before". This is a prediction about a single attribute of an artefact, but otherwise identical to the holistic originality prediction above: it is an expectation about all members of a creative domain, but about only one of their attributes. What is being *predicted* is the "depth" attribute, the form of that *prediction* is an inequality over that attribute, and the *scope* is membership in the domain of mobile phones.

Macedo and Cardoso (2001) use reductionist expectations in a model of surprise. An agent perceives some attributes of an artefact and uses these in a predictive classification. Specifically the agent observes the façades of buildings and constructs an expectation about the kind of building it is observing. The agent then approaches the building and discovers its true function, generating surprise if the expectation is violated. In this case the *predicted* property is the category to which the building belongs and the prediction is the value that property is expected to take.

2. Scope-complete vs. scope-restricted

Expectations can also be categorised according to whether they are *scope complete*, in which case the scope of the expectation is the entire creative domain (the universe of possibilities within the domain the creative system is working), or *scope-restricted*, where the expectation applies only to a subset of possible artefacts. The subset may be defined by a categorisation that is exclusive or non-exclusive, hierarchical or flat, deterministic or stochastic, or any other way of specifying which designs are to be excluded.

The mobile phone examples in the previous section are scope-complete expectations. An example of a scope restricted expectation would be "I expect smartphones to be relatively tall, for a phone". In this case the *predicted* property is device height (making this a reductionist expectation) and the *prediction* is a region of the height attribute bounded by the average for the domain of phones. The scope of this expectation, however, is artefacts in the category "smartphones", a strict subset of the domain of mobile phones in which this creative system operates. This kind of expectation could be used to construct hierarchical models of novelty.

Peters (1998) uses this kind of hierarchy of expectations in a model of surprise – each level of their neural network architecture predicts temporal patterns of movement among the features identified by the layers below it, and surprise is measured as the predictive error. At the highest level the expectations concern the complete domain, while at lower levels the predictions are spatially localised.

3. Conditional vs. unconditional

Conditional expectations predict something about an artefact contingent on another attribute of that artefact. Unconditional expectations require no such contingency, and predict something about the artefacts directly. This is expressed in our framework via the condition property, which contains an expectation's independent variables, while the predicted property contains an expectation's dependent variable(s). A conditional expectation predicts some attribute(s) of an artefact conditionally upon some other attribute(s) of an artefact, while an unconditional expectation predicts attribute(s) directly. In a *conditional* expectation the prediction is that there will be a relationship between the independent attributes (the condition) and the dependent attributes (the predicted). When an artefact is observed this can then be evaluated for accuracy.

Grace et al. (2014a) details a system which constructs conditional expectations of the form "I expect smartphones with faster processors to be thinner". When a phone is observed with greater than average processing power and greater than average thickness this expectation would be violated. In this case the *predicted* property is the thickness (making this a reductionist expectation), the *prediction* is a distribution over device thicknesses, and the *scope* is all smartphones (making this a scope-restricted expectation given that the domain is all mobile devices). The difference from previous examples is that this prediction is *conditional* on another attribute of the device, its CPU speed. Without first observing that attribute of the artefact the expectation cannot be evaluated. In Grace et al. (2014a) the *congruence measure* is the unlikelihood of an observation: the chance, according to the prior probability distribution calculated from the prediction, of observing a device at least as unexpected as the actual observation.

4. Temporal condition vs. atemporal condition

A special case of conditional expectations occurs when the conditional property concerns time: the age of the device, its release date, or the time it was first observed. While all expectations are influenced by time in that they are constructed about observations in the present from experiences that occurred in the past, temporally conditional expectations are expectations where time is the contingent factor. Temporal conditions are used to construct expectations about trends within domains, showing how artefacts have changed over time and predicting that those trends will continue.

Maher, Brady, and Fisher (2013) detail a system which constructs temporally conditional expectations of the form "I expect the weight of more newly released cars to be lower". Regression models are constructed of the how the attributes of personal automobiles have tended to fluctuate over time. In this case the *predicted* property is the car's weight, the *prediction* is a weight value (the median expected value), and the *scope* is all automobiles in the dataset. The *conditional* is the release year of the new vehicle: a weight prediction can only be made once the release year is known. The *congruence* measure in this model is the distance of the new observation from the expected median.

5. Within-artefact temporality vs. within-domain temporality

The question of temporally conditional expectations requires further delineation. There are two kinds of temporally contingent expectation: those where the time axis concerns the whole domain, and those where the time axis concerns the experience of an individual artefact. The above example of car weights is the former kind – the temporality exists within the domain, and individual cars are not experienced in a strict temporal sequence. Within-artefact temporality is critically important to the creativity of artefacts that are perceived sequentially, such as music and narrative. In this case what is being predicted is a component of the artefact yet to be experienced (an upcoming note in a melody, or an upcoming twist in a plot), and that prediction is conditional on components of the artefact that have been experienced (previous notes and phrases, and previous plot events).

Pearce et al. (2010) describes a computational model of melodic expectation which probabilistically expects upcoming notes. In this case the *predicted* property is the pitch of the next note (an attribute of the overall melody), the *prediction* is a probability distribution over pitches. While the *scope* of the predictive model is all melodies within the domain (in that it can be applied to any melody), the *conditional* is the previous notes in the current melody. Only once some notes early in the sequence have been observed can the pitch of the next notes be estimated.

6. Accuracy-measured vs. impact-measured

The first five categorisations in this framework concern the expectation itself, while the last one concerns how unexpectedness is measured when those expectations are violated. Expectations make predictions about artefacts. When a confident expectation proves to be incorrect there are two strategies for measuring unexpectedness: how incorrect was the prediction, and how much did the predictive model have to adjust to account for its failure? The first strategy is *accuracy-measured* incongruence, and aligns with the probabilistic definition of unexpectedness in Ortony and Partridge (1987). The second strategy is *impact-measured* incongruence, and aligns with the information theoretic definition of unexpectedness in Baldi and Itti (2010). In the domain of creativity evaluation the accuracy strategy has been most often invoked in models of surprise, while the impact strategy has been most associated with measures of transformativity.

Grace et al. (2014b) proposes a computational model of surprise that incorporates impact-measured expectations. Artefacts are hierarchically categorised as they are observed by the system, with artefacts that fit the hierarchy well being neatly placed and artefacts that fit the hierarchy poorly causing large-scale restructuring at multiple levels. The system maintains a stability measure of its categorisation of the creative domain, and its expectation is that observations will affect the conceptual structure proportional to the current categorisation stability (which can be considered the system's confidence in its understanding of the domain). Measuring the effect of observing a mobile device on this predictive model of the domain is a measure of impact. These expectations could be converted to a measure of accuracy by instead calculating the classification error for each observation, not the restructuring that results from it. The system would then resemble a computational novelty detector.

Experiments in expectability

To further illustrate our framework for categorising expectation we apply it to several examples from our recent work modelling surprise in the domain of mobile devices (Grace et al., 2014b,a). This system measures surprise by constructing expectations about how the attributes of a creative artefact

relate to each other, and the date which a particular artefact was released is considered as one of those attributes. Surprise is then measured as the unlikelihood of observing a particular device according to the predictions about relationships between its attributes. For example, mobile devices over the course of the two decades between 1985 and 2005 tended, on average, to become smaller. This trend abruptly reversed around 2005-6 as a result of the introduction of touch screens and phone sizes have been increasing since. The system observes devices in chronological order, updating its expectations about their attributes as it does so. When this trend reversed the system expressed surprise of the form "The height of device A is surprising given expectations based on its release date". Details of the computational model can be found in earlier publications.

Figure 1 shows a plot of the system's predictions about device CPU speed the system made based on year of release. At each date of release the system predicts a distribution over expected CPU clock speeds based on previous experiences. The blue contours represent the expected distribution, with the thickest line indicating the median. The white dots indicate mobile devices. The gradient background indicates hypothetical surprise were a device to be observed at that point, with black being maximally surprising. The vertical bands on the background indicate the effect of the model's confidence measure – when predictions have significant error the overall surprise is reduced as the model is insufficiently certain in its predictions, and may encounter unexpected observations because of inaccurate predictions rather than truly unusual artefacts. An arrow indicates the most surprising device in the image, the LG KC-1, released in 2007 with a CPU speed of 806Mhz, considered by the predictive model to be less than 1% likely given the distribution of phone speeds before that observation. Note that after soon after 2007 the gradient of the trend increases sharply as mobile devices started to become general-purpose computing platforms. The KC-1 was clearly ahead of its time, but without the applications and touch interface to leverage its CPU speed it was never commercially successful.

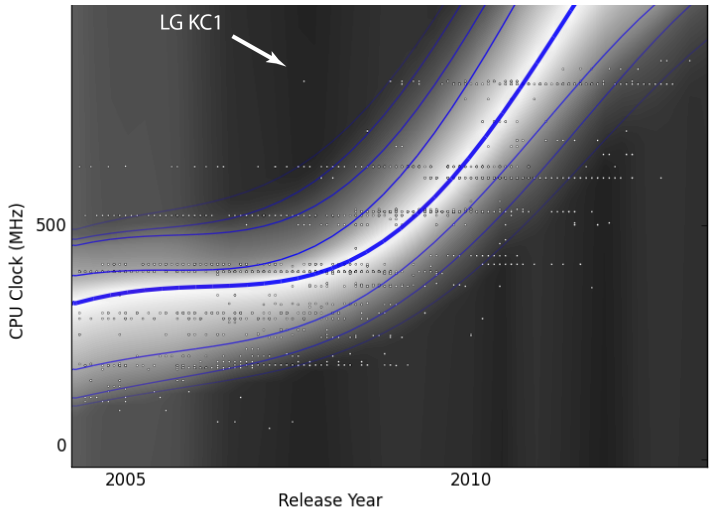


Figure 1: Expectations about the relationship between release year and CPU speed within the domain of mobile devices. The LG KC-1, a particularly unexpected mobile device, is marked.

This is a *reductionist, scope-complete, within-domain temporally conditional* expectation, with congruence measured by *accuracy*. It is reductionist as the predicted attribute is only CPU speed. It is scope-complete because CPU speeds are being predicted for all mobile devices, the scope of this creative system. It is conditional because it predicts a relationship between release year and CPU speed, rather than predicting the latter directly, and that condition is temporal as it is based on the date of release. It is within-domain temporal, as the time dimension is defined with respect to the creative domain, rather than within the observation of the artefact (mobile phones are typically not experienced in a strict temporal order, unlike music or narrative). It is accuracy-measured as incongruence is calculated based on the likelihood of the prediction, not the impact of the observation on the predictive model.

Figure 2 shows another expectation of the same kind as in Figure 1, this time plotting a relationship between device width and release year. The notation is the same as in Figure 1 although without the background gradient. The contours represent the expected distribution of device masses for any given device volume. Here, however, the limits of the scope-complete approach to expectation are visible. Up until 2010 the domain of mobile devices was relatively unimodal with respect to expected width over time. The distribution is approximately a Poisson, tightly clustered around the 40-80mm range with a tail of rare wider devices. Around 2010, however, the underlying distribution changes as a much wider range of devices running on mobile operating systems are released. The four distinct clusters of device widths that emerge – phones, "phablets" (phone/tablet hybrids), tablets and large tablets – are not well captured by the scope-complete expectation. If a new device were observed located midway between two clusters it could reasonably be considered unexpected, but under the unimodality assumption of the existing system this would not occur. A set of *scope-restricted temporally conditional* expectations could address this by predicting the relationship between width and time for each cluster individually. Additionally a measure of the *impact* of the devices released in 2010 on this predictive model could detect the transformational creativity that occurred here.

Figure 3 shows a plot of the system's predictions about device mass based on device volume. Note that – unsurprisingly – there is a strong positive correlation between mass and volume, and that the distribution of expected values is broader for higher volumes. Two groups of highly unexpected devices emerge: those around 50-100cm^3 in volume but greater than 250gr in mass, and those in the 250-500cm^3 range of volumes but less than 250gr mass. Investigations of the former suggest they are mostly ruggedised mobile phones or those with heavy batteries, and investigations of the latter suggest they are mostly dashboard-mounted GPS systems (included in our dataset as they run mobile operating systems).

This is a *reductionist, scope-complete, atemporal condition*, with congruence measured by *accuracy*. By our framework, the difference between the expectations modelled in Figure 1 and Figure 3 are that the former's conditional prediction is contingent on time, while the latter's is contingent on an attribute of the artefacts.

Figure 4 shows the results of a different model of surprise, contrasted with our earlier work in Grace et al. (2014b). An online hierarchical conceptual clustering algorithm (Fisher, 1987) is used to place each device, again observed chronologically, within a hierarchical classification tree that evolves and restructures itself as new and different devices are observed.

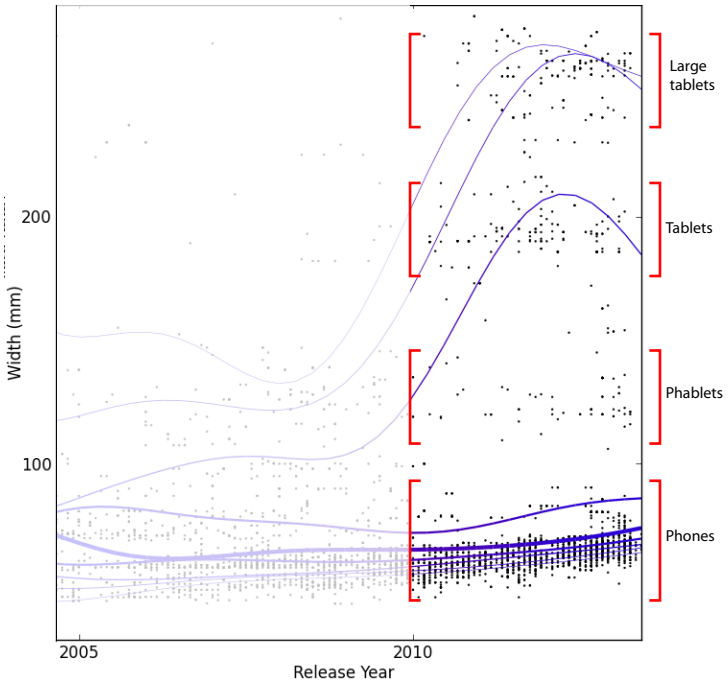


Figure 2: Expectations about the relationship between the release year and width of mobile devices. Note that the distribution of widths was roughly unimodal until approximately 2010, when four distinct clusters emerged.

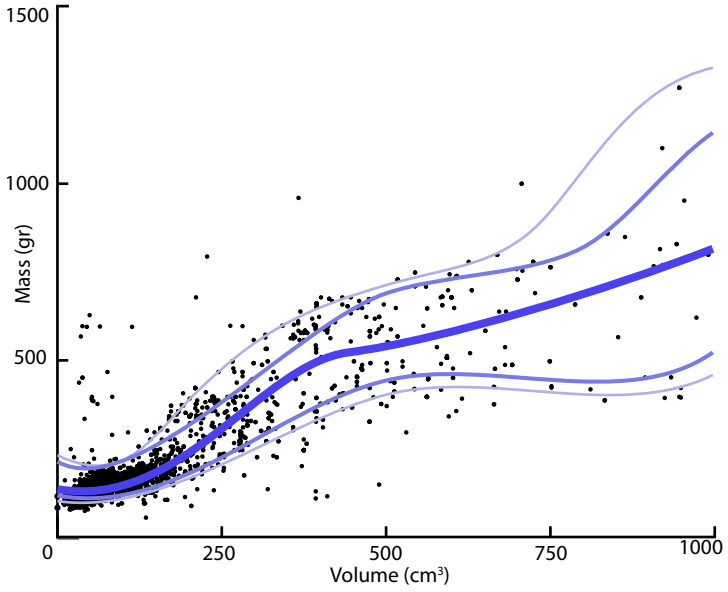


Figure 3: Expectations about the relationship between volume and mass within the domain of mobile devices.

The degree to which a particular device affects the tree structure can then be measured, indicating the amount by which it transformed the system's knowledge of the domain. The most unexpected device according to this measure were the Bluebird Pidiom BIP-2010, a ruggedised mobile phone which caused a redrawing of the physical dimensions based boundary between "tablet" and "phone" and caused a large number of devices to be recategorised as one or the other (although it must be noted that such labels are not known to the system). The second most unexpected device was the ZTE U9810, a 2013 high-end smartphone which put the technical specs of a tablet into a much smaller form factor, challenging the system's previous categorisation of large devices as also being powerful. The third most unexpected device was the original Apple iPad, which combined high length and width with a low thickness, and had more in common internally with previous mobile phones than with previous tablet-like devices.

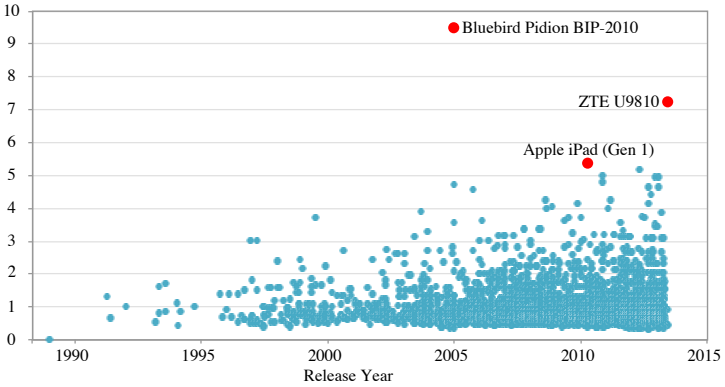


Figure 4: Incongruence of mobile devices with respect to their impact on learnt conceptual hierarchy. Three particularly unexpected devices are labelled.

This is a *reductionist, scope-complete, unconditional* expectation with congruence measured by *impact*. It is reductionist it does not predict all attributes of the device, only that there exists certain categories within the domain. It is scope-complete as it applies to all devices within the domain. It is unconditional as the prediction is not contingent on observing some attribute(s) of the device. The primary difference from the previous examples of expectation is the congruence measure, which measures not the accuracy of the prediction (which would be the classification error), but the degree to which the conceptual structure changes to accommodate the new observation.

Novelty, surprise, or transformativity?

Our categorisation framework demonstrates the complexity of the role of expectation in creativity evaluation, motivating the need for a deeper investigation. We argue that expectation underlies novelty, surprise, and transformativity, but further work is needed before there is consensus on what kinds of expectation constitute each notion.

Macedo and Cardoso (2001) adopt the definition from Ortony and Partridge (1987) in which surprise is an emotion elicited by the failure of confident expectations, whether those expectations were explicitly computed beforehand or generated in response to an observation. By this construction all forms of expectation can cause surprise, meaning that surprise and novelty have considerable overlap. Wiggins (2006a)

goes further, saying that surprise is always a response to novelty, and thus need not be modelled separately to evaluate creativity. Schmidhuber (2010) takes the opposite approach, stating that all novelty is grounded in unexpectedness, and that creativity can be evaluated by the union of usefulness and improvement in predictability (which would, under our framework, be a kind of *impact-based congruence*). Wiggins (2006b) would consider Schmidhuber's "improvement in predictability" to be a kind of transformation as it is a measure of the degree of change in the creative system's rules about the domain. Maher and Fisher (2012) state that the dividing line between novelty and surprise is temporality – surprise involves expectations about what will be observed next, while novelty involves expectations about what will observed at all. Grace et al. (2014a) expand that notion of surprise to include any conditional expectation, regardless of temporality.

We do not offer a conclusive definition of what constitutes novelty, what constitutes surprise, and what constitutes transformativity, only that each can be thought of as expectation-based. It may well be that – even should we all come to a consensus set of definitions – the three categories are not at all exclusive. We offer some observations on the properties of each as described by our framework:

- Surprise captures some kinds of creativity-relevant expectation that extant models of novelty do not, namely those concerned with trends in the domain and relationships between attributes of artefacts.
- Models of surprise should be defined more specifically than "violation of expectations" if the intent is to avoid overlap with measures of novelty, as novelty can also be expressed as a violation of expectations.
- The unexpectedness of an observation and the degree of change in the system's knowledge as a response to that observation can be measured for any unexpected event, making (P-)transformativity a continuous measure. Models of transformative creativity should specify the kind and degree of change that are necessary to constitute creativity.

Conclusion

We have sought to build theoretical bridges between the notions of novelty, surprise and transformation, reconceptualising all three as forms of expectation. This approach is designed to offer a new perspective on debates about the roles of those disparate notions in evaluating creativity. We have developed a framework for characterising expectations that apply to the evaluation of creativity, and demonstrated that each of novelty evaluation, surprise evaluation, and transformational creativity can be conceived in terms of this framework. Given the wide variety of kinds of expectation that should be considered creativity-relevant we argue that originality alone is not a sufficient accompaniment to value to constitute creativity. This insufficiency is a critical consideration for computational models that can recognise creativity. The expectation-centric approach provides a framing device for future investigations of creativity evaluation. Expectation both serves as a common language by which those seeking to computationally model creativity can compare their disparate work, and provides an avenue by which human judgements of creativity might be understood.

References

Abra, J. 1988. *Assaulting Parnassus: Theoretical views of creativity*. University Press of America Lanham, MD.

Baldi, P., and Itti, L. 2010. Of bits and wows: a bayesian theory of surprise with applications to attention. *Neural Networks* 23(5):649–666.

Berlyne, D. E. 1966. Curiosity and exploration. *Science* 153(3731):25–33.

Berlyne, D. E. 1970. Novelty, complexity, and hedonic value. *Perception & Psychophysics* 8(5):279–286.

Bishop, C. M. 1994. Novelty detection and neural network validation. In *Vision, Image and Signal Processing, IEE Proceedings-*, volume 141, 217–222. IET.

Boden, M. A. 2003. *The creative mind: Myths and mechanisms*. Routledge.

Csikszentmihalyi, M. 1988. *Society, culture, and person: A systems view of creativity*. Cambridge University Press.

Dudek, S. Z. 1993. The morality of 20th-century transgressive art. *Creativity Research Journal* 6(1-2):145–152.

Feyerabend, P. K. 1963. How to be a good empiricist: a plea for tolerance in matters epistemological. In *Philosophy of science: The Delaware seminar*, volume 2, 3–39. New York, Interscience Press.

Fisher, D. H. 1987. Knowledge acquisition via incremental conceptual clustering. *Machine learning* 2(2):139–172.

Florida, R. L. 2012. *The Rise of the Creative Class: Revisited*. Basic books.

Grace, K.; Maher, M.; Fisher, D.; and Brady, K. 2014a. A data-intensive approach to predicting creative designs based on novelty, value and surprise. *International Journal of Design Creativity and Innovation* (to appear).

Grace, K.; Maher, M.; Fisher, D.; and Brady, K. 2014b. Modeling expectation for evaluating surprise in design creativity. In *Proceedings of Design Computing and Cognition*, volume (to appear).

Horvitz, E. J.; Apacible, J.; Sarin, R.; and Liao, L. 2012. Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. *arXiv preprint arXiv:1207.1352*.

Itti, L., and Baldi, P. 2005. A principled approach to detecting surprising events in video. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, 631–637. IEEE.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

Macedo, L., and Cardoso, A. 2001. Modeling forms of surprise in an artificial agent. *Structure* 1(C2):C3.

Maher, M. L., and Fisher, D. H. 2012. Using ai to evaluate creative designs. In *2nd International conference on design creativity (ICDC)*, 17–19.

Maher, M. L.; Brady, K.; and Fisher, D. H. 2013. Computational models of surprise in evaluating creative design. In *Proceedings of the Fourth International Conference on Computational Creativity*, 147.

Maher, M. L. 2010. Evaluating creativity in humans, computers, and collectively intelligent systems. In *Proceedings of the 1st DESIRE Network Conference on Creativity and Innovation in Design*, 22–28. Desire Network.

Newell, A.; Shaw, J.; and Simon, H. A. 1959. *The processes of creative thinking*. Rand Corporation.

O'Quin, K., and Besemer, S. P. 1989. The development, reliability, and validity of the revised creative product semantic scale. *Creativity Research Journal* 2(4):267–278.

Ortony, A., and Partridge, D. 1987. Surprisingness and expectation failure: what's the difference? In *Proceedings of the 10th international joint conference on Artificial intelligence-Volume 1*, 106–108. Morgan Kaufmann Publishers Inc.

Pearce, M. T.; Ruiz, M. H.; Kapasi, S.; Wiggins, G. A.; and Bhattacharya, J. 2010. Unsupervised statistical learning underpins computational, behavioural, and neural manifestations of musical expectation. *NeuroImage* 50(1):302–313.

Peters, M. 1998. Towards artificial forms of intelligence, creativity, and surprise. In *Proceedings of the twentieth annual conference of the cognitive science society*, 836–841. Citeseer.

Runco, M. A. 2010. *Creativity: Theories and themes: Research, development, and practice*. Access Online via Elsevier.

Saunders, R., and Gero, J. S. 2001a. The digital clockwork muse: A computational model of aesthetic evolution. In *Proceedings of the AISB*, volume 1, 12–21.

Saunders, R., and Gero, J. S. 2001b. Artificial creativity: A synthetic approach to the study of creative behaviour. *Computational and Cognitive Models of Creative Design V, Key Centre of Design Computing and Cognition, University of Sydney, Sydney* 113–139.

Saunders, R. 2012. Towards autonomous creative systems: A computational approach. *Cognitive Computation* 4(3):216–225.

Schmidhuber, J. 2010. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *Autonomous Mental Development, IEEE Transactions on* 2(3):230–247.

Schumpeter, J. 1942. Creative destruction. *Capitalism, socialism and democracy*.

Silberschatz, A., and Tuzhilin, A. 1995. On subjective measures of interestingness in knowledge discovery. In *KDD*, volume 95, 275–281.

Sosa, R., and Gero, J. S. 2005. A computational study of creativity in design: the role of society. *AI EDAM* 19(4):229–244.

Strzalecki, A. 2000. Creativity in design: General model and its verification. *Technological Forecasting and Social Change* 64(2):241–260.

Wiggins, G. A. 2006a. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Wiggins, G. A. 2006b. Searching for computational creativity. *New Generation Computing* 24(3):209–222.

Stepping Back to Progress Forwards: Setting Standards for Meta-Evaluation of Computational Creativity

Anna Jordanous
Centre for e-Research, Department of Digital Humanities
King's College London
26-29 Drury Lane, London WC2B 5RL, UK
anna.jordanous@kcl.ac.uk

Abstract

There has been increasing attention paid to the question of how to evaluate the creativity of computational creativity systems. A number of different evaluation methods, strategies and approaches have been proposed recently, causing a shift in focus: which methodology should be used to evaluate creative systems? What are the pros and cons of using each method? In short: how can we evaluate the different creativity evaluation methodologies? To answer this question, five meta-evaluation criteria have been devised from cross-disciplinary research into good evaluative practice. These five criteria are: *correctness; usefulness; faithfulness as a model of creativity; usability of the methodology; generality*. In this paper, the criteria are used to compare and contrast the performance of five various evaluation methods. Together, these meta-evaluation criteria help us explore the advantages and disadvantages of each creativity evaluation methodology, helping us develop the tools we have available to us as computational creativity researchers.

Introduction

Computational creativity evaluation repeatedly appears as a theme in the calls for papers for the ICCC conference series. Such emphasis underlines the growing importance of evaluation to the computational creativity research community.

For transparent and repeatable evaluative practice, it is necessary to state clearly what standards/methods are used for evaluation (Jordanous 2012a). Despite, or perhaps because of, a lack of creativity evaluation being employed in the computational creativity research community until recently (Jordanous 2011), a number of creativity evaluation strategies have been proposed in recent years (Pease, Winterstein, and Colton 2001; Ritchie 2007; Colton et al. 2010; Colton, Charnley, and Pease 2011; Jordanous 2012b). Herein lies a decision for a computational creativity researcher: which evaluation strategy should be adopted to evaluate computational creativity systems? What are the benefits and disadvantages of each?

Such questions have not previously been examined to any detailed extent in computational creativity research. In various other research fields, though, issues around 'evaluating evaluation', or *meta-evaluation*. have been considered in some detail. Meta-evaluation has been considered from philosophical and more practical standpoints. As a burgeoning research community, computational creativity researchers can learn from such considerations, as they apply to our own research efforts.

This paper proposes five standards for meta-evaluation of creativity evaluation methodologies, informed by the wider literature and by evaluative practices outside of the computational creativity field. These standards are offered as factors for assessment and comparison of creativity evaluation methodologies, to help us develop good evaluative practice in computational creativity research.

The five meta-evaluation standards are applied to a case study on creative system evaluation, comparing different evaluation methodologies against each other. Results are reported below. It is proposed that these five standards should help guide us in refining our work on computational creativity evaluation, as we progress in the development of this important area of computational creativity research.

The need to evaluate creativity evaluation

We have an intuitive but tacit understanding of the concept of creativity that we can access introspectively (Kaufman 2009; Jordanous 2012a). For comparative purposes and methodical, transparent evaluation, this intangible understanding is not sufficient to help us identify and learn from our successes and failures in computational creativity research.

To solve the problem of how to evaluate creative systems, various evaluation methodologies or strategies have been offered including the tests offered by Pease, Winterstein, and Colton, Ritchie's empirical criteria, the creative tripod model, the FACE model and the SPECS methodology (Pease, Winterstein, and Colton 2001; Ritchie 2007; Colton et al. 2010; Colton, Charnley, and Pease 2011; Jordanous 2012b, respectively).[1] But which should computational creativity researchers use?

One should note here that we are unlikely to find one single fully-specified, detailed, step-by-step methodology to suit all types of creative system. What we can do is to understand the strengths and weaknesses of different methodologies. Through trial, application of and comparison between different methodologies, refine and develop our eval-

[1] See Jordanous 2012a for full discussion of these methodologies and strategies.

uation strategies within computational creativity so that we can mutually learn from our advances and mistakes; the very essence of what evaluation offers researchers, after all.

How can these methodologies be compared against each other? Reviewing various features of the methodologies and comparing them against each other helps us to learn through comparison. Below, five meta-evaluation standards are identified for comparison and evaluation of creativity evaluation methodologies. These five meta-evaluation standards are drawn from cross-disciplinary reviews of evaluative practice. The meta-evaluation standards are applied in a practical case study, reported below. From this application of the standards, we can appreciate the strengths and weaknesses of each creativity evaluation methodology, guiding us in our evaluative choices when developing computational creativity research. With these meta-evaluation criteria, we can now compare evaluative results obtained through different methods and discuss how useful each of these evaluations are to the computational creativity researcher. Gathering effective evaluative feedback, using solidly developed evaluation methodologies, assists further computational creativity research development and helps identify more clearly the contributions to knowledge made by our research.

Criteria for meta-evaluation of creativity evaluation methodologies

Criteria for evaluation should be clearly stated and justified (Jordanous 2012a). This theme also applies to meta-evaluation criteria for comparing various creativity evaluation methodologies.

Certain areas suggest themselves as meta-evaluation criteria for assessing creativity evaluation methodologies, such as the accuracy and usefulness of the feedback to a researcher, or ease of applicability.

Pease, Winterstein, and Colton (2001) identify two candidate meta-evaluation criteria:

> 'Firstly, to what extent do they reflect human evaluations of creativity, and secondly, how applicable are they?' (Pease, Winterstein, and Colton 2001, p. 9)

More recently, Pease has suggested the set of {generality, usability, faithfulness, value of formative feedback} as candidate criteria (Pease, 2012, personal communications). In relevant literature on evaluation and related literature on proof of hypotheses in scientific method, other contributions could also be used as criteria for measuring the success of computational creativity evaluation methodologies, as outlined below.

Criteria for testing scientific hypotheses and explanatory theories Sloman (1978) outlined seven types of 'interpretative aims of *science*' (Sloman 1978, p. 26, my emphasis added), of which the third aim is the forming of explanatory theories for things we know exist. In the context of this current work, an example of the explanatory theories mentioned in the third aim would be a theory that allows us to explain if or why a computational creativity system is creative. Ten criteria were offered by Sloman (1978) as criteria for comparison of explanatory theories.

> 'a good explanation of a range of possibilities should be definite, general (but not too general), able to explain fine structure, non-circular, rigorous, plausible, economical, rich in heuristic power, and extendable.' (Sloman 1978, p. 53)

Within these criteria there is some significant interdependence and Sloman advises that the criteria are best treated as a set of inter-related criteria rather than distinct yardsticks, with some criteria (such as plausibility, generality and economy) to be used with caution. This may help to explain why Sloman's list of criteria is longer than others mentioned in this Section.

Thagard (1988) defined a 'good theory' as 'true, acceptable, confirmed' (Thagard 1988, p. 48). These criteria were later expressed in the form of 'the criteria of consilience, simplicity of analogy' (Thagard 1988, p. 99) as essential criteria for theory evaluation:

- *Consilience* - how comprehensive the theory is, in terms of how much it explains.
- *Simplicity* - keeping the theory simple so that it does not try to over-explain a phenomenon. Thagard mentions in particular that a theory should not try to 'achieve consilience by means of ad hoc auxiliary hypotheses' (Thagard 1988, p. 99). In other words, the main explanatory power of the theory should map closely to the main part of that theory, without needing extensive correction and supplementation.
- *Analogy* - boosting the 'explanatory value'(Thagard 1988, p. 99) of a theory by enabling it to be applied to other demands. This is especially appropriate where theories can be cross-applied in more established domains where knowledge of facts is more developed.

Guidelines for good practice in research evaluation Suggestions for good practice in performing evaluation in research can be interpreted as criteria that identify such good practice. For example, in his 'Short Course on Evaluation Basics', John W. Evans identifies four 'characteristics of a good evaluation':[2] a good evaluation should be objective, replicable, generalisable and as 'methodologically strong as circumstances will permit'. In considering what constitutes good evaluation practice, the MEERA website ('My Environmental Education Evaluation Resource Assistant')[3] describes 'good evaluation' as being: 'tailored to your program ... crafted to address the specific goals and objectives [of your program'; '[building] on existing evaluation knowledge and resources'; inclusive of as many diverse viewpoints and scenarios as reasonable; replicable; as unbiased and honest as possible; and 'as rigorous as circumstances allow'. From a slightly different perspective on research evaluation, the European Union FP6 Framework Programme describes how FP6-funded projects are evaluated in terms of three criteria:

[2] http://edl.nova.edu/secure/evasupport/evaluationbasics.html, last accessed Feb 2014.

[3] All quotes from the MEERA website are taken from http://meera.snre.umich.edu/plan-an-evaluation/evaluation-what-it-and-why-do-it#good, last accessed Feb 2014.

a project's *rationale* relative to funding guidelines and resources; *implementation* effectiveness, appropriateness and cost-effectiveness; and *achievements* and impact of contributions of objectives and outputs.

Dealing with subjective and/or fuzzy data: Blanke's specificity and exhaustivity In computational creativity evaluation, the frequency of data being returned is low and the correctness of that data is generally subjective and/or fuzzy in definition, rather than being discretely categorisable as either correct or incorrect, or as either present or missing. Blanke (2011) looked at how to evaluate the success of a methodology for measuring aspects like precision and recall, in cases where the results being returned were somewhat difficult to pin down to exact matches due to fuzziness in what could be returned as a correct result. The specific case Blanke considered was in XML retrieval evaluation, where issues such as hierarchical organisation and overlap of elements, and the identification of what was an appropriate part of an XML document to return, caused problems with using precision and recall measures. There was also an issue with relatively low frequencies in what was being returned.

As an evaluation solution, Blanke (2011) proposed *component specificity* and *topical exhaustivity*, following from Kazai and Lalmas (2005). Exhaustivity 'is measured by the size of overlap of query and document component information' (Blanke 2011, p. 178). Specificity 'is determined by counting the rest of the information in the component [of an XML document] that is not about the query' (Blanke 2011, p. 178), such that minimising such information will maximise the specificity value, as more relevant content is returned.

Identifying meta-evaluation criteria

Drawing all the above contributions together, five criteria can be identified for meta-evaluation of computational creativity evaluation methodologies. These are presented here, with relevant points from the comments above being grouped under the most relevant criterion, as far as possible. Some overlap across criteria is acknowledged, for example Thagard's *analogy* criterion can be interpreted as being concerned with both 'usefulness' and 'generality'.

- **Correctness: how accurately and comprehensively the evaluation findings reflect the system's creativity.**
  - MEERA's *honesty of evaluation* criterion.
  - MEERA's *inclusiveness of diverse relevant scenarios* criterion.
  - Evans' *objectiveness* criterion.
  - MEERA's *avoidance of bias in results* criterion.
  - Sloman's *definiteness* criterion.
  - Sloman's *rigorousness* criterion.
  - Sloman's *plausibility* criterion.
  - Thagard's *consilience* criterion.
  - Blanke's *exhaustivity* criterion.
  - Evans' *methodological strength* criterion.
- **Usefulness: how informative the evaluative findings are for understanding and potentially improving the creativity of the system.**
  - Pease's *value of formative feedback* criterion.
  - FP6's *rationale*, *implementation* and *achievements* criteria.
  - Sloman's *heuristic power* criterion.
  - Thagard's *analogy* criterion.
- **Faithfulness as a model of creativity: how faithfully the evaluation methodology captures the *creativity* of a system (as opposed to other aspects of the system).**
  - Pease, Winterstein, and Colton (2001)'s *reflection of human evaluations of creativity* criterion.
  - Pease's *faithfulness* criterion.
  - MEERA's *tailoring of the method to specific goals and objectives* criterion.
  - Blanke's *specificity* criterion.
- **Usability of the methodology: the ease with which the evaluation methodology can be applied in practice, for evaluating the creativity of systems.**
  - Pease, Winterstein, and Colton (2001)'s *applicability* criterion.
  - Pease's *usability* criterion.
  - Evans' *replicability* criterion.
  - MEERA's *replicability* and *rigorousness of a methodology* criteria.
  - Sloman's *non-circularity* criterion.
  - Sloman's *rigorous* and *explicitness* criteria (in how to apply the methodology).
  - Sloman's *economy of theory* criterion.
  - Thagard's *simplicity* criterion.
- **Generality: how generally applicable this methodology is across various types of creative systems.**
  - Pease's *generality* criterion.
  - MEERA's *inclusiveness of diverse relevant scenarios* criterion.
  - Evans' *generalisability* criterion.
  - Sloman's *generality* criterion.
  - Sloman's *extendability* criterion.
  - Thagard's *analogy* criterion.

Applying the criteria: a case study

Now we have identified these five meta-evaluation criteria, we can use them to evaluate the performance of computational creativity evaluation methodologies.

Previously, three different musical improvisation computer systems were evaluated using various computational creativity evaluation methodologies, to compare how creative each system was (Jordanous 2012a; 2012b). The task in this current work is to consider how well the creativity evaluation methodologies performed for this assessment.

For an independent assessment of the relative performance of the evaluation methodologies, external evaluation was sought to consider and perform meta-evaluation

on five key existing evaluative approaches (Ritchie 2007; Colton 2008; Colton, Charnley, and Pease 2011; Jordanous 2012b, surveys of human opinion). The invited external evaluators were the key researchers involved in creating the musical improvisation systems examined in the above-mentioned creativity evaluation case study(Jordanous 2012a): Al Biles (GenJam) and George Lewis (Voyager). Bob Keller was also invited because of his research into and development of a related musical improvisation system, the Impro-Visor system (Gillick, Tang, and Keller 2010).[4] Evaluators were asked to view all the evaluative feedback obtained. They were then asked to give their opinions (as developers of musical improvisation systems) on various aspects of each methodology and on the results obtained.

Below, the methodology used for the meta-evaluation is briefly described, and the obtained meta-evaluations are reported and discussed. Fuller details can be found in Jordanous (Jordanous 2012a).

Methodology for obtaining external evaluation

Each external evaluator was given a feedback sheet reporting the evaluation feedback obtained for their system from each creativity evaluation methodology being investigated: Ritchie's criteria; Colton's creative tripod; survey of human opinion; the FACE model; and SPECS+cc. (N.B. *SPECS+cc* is used here to indicate the use of Jordanous's SPECS methodology with the 14 creativity components (Jordanous 2012a) as the adopted definition of creativity, as recommended (Jordanous 2012b).)

For each methodology, the sheets also included brief comparisons between systems according to the systems' evaluated creativity. An example of these feedback sheets, given in (Jordanous 2012a, Appendices), presents the sheet provided to Al Biles to report the evaluation results for GenJam. A similar set of feedback was prepared and sent to George Lewis as evaluative feedback relating to Voyager. Methodologies were presented under anonymous identifiers in the feedback sheet to avoid any bias from being introduced, as far as possible.

Evaluators were first asked if they had any initial comments on the results. They were then asked to provide full feedback for each methodology in turn, on the five criteria derived above. They looked at all five criteria for the current methodology and then were asked for any final comments on that methodology before moving onto the next methodology. Methodologies were presented to the evaluators in a randomised order, to avoid introducing any ordering bias.

For each criterion, questions and illustrating examples were composed to present the criterion in a context appropriate for computational creativity evaluation. These questions and examples, listed below, were put to external evaluators to gather their feedback on each criterion as meta-evaluation of the various evaluation methodologies.

- **Correctness:**
  - How correct do you think these results are, as a reflection of your system?
  - For example: are the results as accurate, comprehensive, honest, fair, plausible, true, rigorous, exhaustive, replicable and/or as objective as possible?
- **Usefulness:**
  - How useful do you find these evaluation results, as an / the author of the system?
  - For example: do the results provide useful information about your system, give you formative feedback for further development, identify contributions to knowledge made by your system, or give other information which you find helpful?
- **Faithfulness as a model of creativity:**
  - How faithfully do you think this methodology models and evaluates the creativity of your system?
  - For example: do you think the methodology uses a suitable model(s) of creativity for evaluation, does the methodology match how you expect creativity to be evaluated, how specifically does the methodology look at creativity (rather than other evaluative aims)?
- **Usability of the methodology:**
  - How usable and user-friendly do you think this methodology is for evaluating the creativity of computational systems?
  - For example: would you find the methodology straightforward to use if wishing to evaluate the creativity of a computational creativity system (or systems), is the methodology stated explicitly enough to follow, is the method simple, could you replicate the experiments done with this methodology in this evaluation case study?
- **Generality:**
  - How generally do you think this methodology can be applied, for evaluation of the creativity of computational systems?
  - For example: can the methodology accommodate a variety of different systems, be generalisable and extendable enough to be applied to diverse examples of systems, and/or different types of creativity?

For each criterion, evaluators were asked to rate the system's performance on a 5 point Likert scale (all of a format ranging from positive extreme to negative extreme, such as: [Extremely useful, Quite useful, Neutral, Not very useful, Not at all useful]). They could also add any comments they had for each criterion.

Evaluators were asked about the correctness and usefulness of the methodology's results, before learning how the methodology worked. This gave the advantage of being able to hear the evaluators' opinions considering the feedback results in isolation, without any influence from how the results were obtained. Nonetheless, the process by which a product was generated is important to consider alongside that product, for a more rounded and informed evaluation (Rhodes

[4] The author of one evaluated systems (GAmprovising) was not included, due to being the author of one of the evaluation methods being examined (and the researcher conducting this work).

1961). Evaluators were given details on how that methodology worked after evaluating the correctness and usefulness criteria. They were then asked to provide feedback for the final three criteria (faithfulness, usability and generality). The details provided to explain each methodology are reproduced in Jordanous (Jordanous 2012a, Appendices).[5]

Finally, evaluators were asked to rank the evaluation methodologies according to how well they thought the methodologies evaluated the creativity of their system overall. Although the formative feedback is, again, probably more useful in terms of developing the various methodologies, it was interesting to see evaluators' opinions on how the methodologies compared to each other. The rankings, completed by Al Biles and Bob Keller, are reported in Table 1. At this point, evaluators were also given a change to add any final comments, before finishing the study. Al Biles completed a full evaluation of all methodologies and (due to time constraints) George Lewis provided evaluations of two methodologies: Colton's creative tripod and the SPECS+cc methodology. Bob Keller also provided comments on some aspects of all methodologies.

Results and discussion of meta-evaluation

Al Biles summarised the meta-evaluation of the five different methodologies with: *'Five very different approaches, and each bring something to the table.'* In the comparisons between methodologies and the overall rankings listed in Table 1, SPECS+cc was either considered the best methodology overall (ahead of the creative tripod) or the second best (behind Ritchie's criteria) for evaluating a system's creativity. The more useful information, though comes from the more detailed formative feedback and comments rather than a single summative ranking as given in Table 1.

SPECS+cc was evaluated by both Biles and Lewis, with some additional comments from Keller. SPECS+cc generated 'extremely useful' and 'quite correct results', in both of the main evaluators' opinions. One evaluator found SPECS+cc to be an 'extremely faithful' model of creativity, though the other was 'neutral' on this matter. While one evaluator found SPECS+cc 'quite user-friendly', the other questioned how user-friendly the SPECS+cc methodology would be, given the steep learning curve in understanding the components. In terms of generality, evaluators disagreed on how generally SPECS+cc could be applied, further comments illustrated how methods like SPECS+cc were more appropriate for taking into account other system goals, compared to more limited views on creativity such as in the FACE model. Biles and Keller in particular commented on the lack of accommodation of other system goals in the FACE model, though it is to be acknowledged that such accommodation does not form one of the goals of the FACE model and is more of an unintended but useful consequential result in models such as SPECS+cc.

FACE was placed third in the overall rankings by Biles and last by Keller. Biles, the main evaluator for FACE, found the results generated by FACE to be 'completely correct', but gave a neutral opinion (neither positive nor negative) on the usefulness of FACE model feedback, the generality of the FACE model across domains and the faithfulness of the FACE model as a model of creativity. FACE was deemed 'quite user-friendly' due to its simplicity; this opinion was repeated, more strongly, for the other creativity evaluation framework Colton was involved in, the creative tripod. Lewis and Biles both evaluated the tripod; they disagreed as to whether the tripod would be generally applicable across many domains, and also as to how faithfully the tripod modelled creativity. Both evaluators agreed, however, that the feedback from the tripod was 'extremely useful' and either 'completely correct' or 'quite correct'. Biles ranked the creative tripod as the second best creativity evaluation methodology overall, though Keller placed it last.

Ritchie's criteria methodology was fully evaluated by Biles. Biles found the criteria to produce 'quite correct', 'quite useful' feedback that was 'quite faithful to creativity' (despite raising issues with enforced simplifications of the data due to the boolean rather than continuous nature of the feedback). Biles was 'neutral' on the usability of applying the criteria for creativity evaluation and on their generality, questioning how the generic terminology used to solicit ratings of typicality and value could be applied to different domains successfully. Keller considered Ritchie's criteria to be the best methodology overall for creativity evaluation, though Biles gave it a middling ranking.

The opinion survey was ranked overall to be the fourth best methodology out of the five. It received a few negative comments from Biles, the main evaluator for this system, despite Biles noting that 'nothing is simpler than just ... asking whether something is creative or not' and that the survey solicited spontaneous,'unadulterated' opinions rather than restructuring the feedback (though Biles also noted that the tripod feedback was clearer than the survey feedback due to its more structured presentation). Biles was guided in a number of comments by an observation that the opinion survey sacrificed reliability/consistency of results for greater validity in terms of the personal qualitative feedback. He thought that the survey approach could be applied 'quite generally' and was 'quite user-friendly' and 'quite faithful' to what it means to be creative. The success of this methodology would depend on the type of person participating, and whether they were clear on what 'creative' means. Given that the GenJam system has been publicly presented many times before, though, Biles felt he learned nothing new from the feedback from the survey, unlike the other methodologies. He was 'neutral' on the correctness of the methodology, confirming observations made in Jordanous 2012a that human opinion cannot be relied on as a 'ground truth' to measure evaluations against, due to varying viewpoints.

[5] It is worth noting that methodologies may well perform differently against the five criteria when applied to different systems (a meta-application of the generality criterion?) The evaluators cannot be expected to give rigorous feedback on the potential of the methodologies in evaluating *any* possible type of system, and we should refrain from drawing too-broad conclusions from their feedback. Nonetheless, with careful consideration of the evaluators' feedback, we gain valuable insights on the methodologies.

Table 1: Judges were asked to rank the methodologies according to how well overall they thought the methodologies evaluated the systems' creativity:

Position	Al Biles	Bob Keller
1st (best)	SPECS+cc	Ritchie's criteria
2nd	Creative Tripod	SPECS+cc
3rd	Ritchie's criteria	FACE
4th	Opinion survey	Opinion survey
5th (worst)	FACE	Creative Tripod

Comparing and contrasting methodologies

Five meta-evaluation criteria have now been identified for meta-evaluation of creativity evaluation methodologies and have been used for evaluation by external evaluators, as reported above. Next, the criteria were applied for further analysis of all the methodologies investigated earlier in this paper, using the full findings from the Jordanous (2012a) case study evaluating the creativity of musical improvisation systems. Such considerations on the methodologies allow us to compare if, and how, a particular evaluation methodology marks a development of our evaluation 'toolkit' as computational creativity researchers. Here, the considerations are focused towards evaluating how well the SPECS+cc methodology (Jordanous 2012a) performed, to gain feedback as to how to improve SPECS+cc and what its strengths were in comparison to other methods. The considerations below also complement the evaluative case study findings by accounting for more detailed information and observations that may not have been detected by the external evaluators, but which should still be considered.

Correctness Showing that human opinion cannot necessarily be relied on as a ground truth, even on a large scale, some participants in opinion surveys admitted that they were likely to be evaluating the systems based on how highly they rated a system's performance overall rather than specifically how creative they thought it was, which would affect the overall correctness of the results of evaluations from the human opinion survey.

SPECS+cc performed better than Ritchie's criteria for correctness. Although Ritchie's 18 criteria have a comprehensive coverage of observations over the products of the system, criteria evaluation is based solely on the products of the creative system, not accounting for the system's process, or observations on the system or how it interacted with its environment. Colton's tripod model was found to be reasonably accurate in terms of identifying and evaluating important aspects in the case study, but it has disregarded aspects such as social interaction, communication and intention, which have been shown to be very important in understanding how musical improvisation creativity is manifested (Jordanous and Keller 2014).

It should be noted that 'correctness' does not imply that the results from evaluation match common human consensus as a 'ground truth', or 'right answer'; Jordanous (Jordanous 2012a) demonstrated that these are not reliable goals in creativity evaluation. Instead, correctness is concerned with how appropriate the feedback is and how accurately and realistically the feedback describes the system.

Usefulness The methodologies differed in the amount of feedback generated through evaluation. A fairly large volume of qualitative and quantitative feedback was returned through the application of SPECS+cc. This is unlike Ritchie's criteria which only returned a set of 18 Boolean values, one for each criterion, with some interpretation effort needed to understand how each criterion influences creativity within the system.[6] Colton's creative tripod generated feedback for 3 components, rather than 14 components, so was shorter than SPECS+cc. The human opinion surveys generated similar quantities of feedback to SPECS+cc, from more people but a shallower level of detail.

The human opinions surveys returned less detailed feedback than SPECS+cc, which generated a large amount of detailed formative feedback. The opinion surveys' feedback also often concentrated on aspects of the systems other than its creativity, according to participant feedback (Jordanous 2012a).

Ritchie's criteria returned a set of 18 boolean values rather than any formative feedback, in a fairly opaque form given the formal abstraction of the criteria specification; if there were no output examples, Ritchie's criteria would not generate any feedback at all, even based on other observations about the system. Colton's creative tripod returned information at the same level of detail as SPECS+cc per component/tripod quality, but less information overall, as several useful components of SPECS+cc were overlooked because they did not map onto the set of tripod aspects.

Faithfulness as a model of creativity Participant feedback for the human opinion surveys acknowledged that evaluations may have related more to the quality of the system, not its creativity, with several participants requesting a definition of creativity to refer to when evaluating how creative the systems were (Jordanous 2012b). The SPECS methodology requires researchers to base their evaluations on a researched and informed understanding of creativity that takes into account both domain-specific and domain-independent aspects of creativity. In this way it is the only methodology that directly accounts for specific informed requirements for creativity in a particular domain. Human opinion surveys would acknowledge this but only tacitly, without these requirements necessarily being identifiable or explainable. Although the parameters and weights in Ritchie's criteria could be customised to reflect differing requirements for creative domains, in practice no researchers have attempted this

[6] One reviewer of this paper pointed out that Ritchie (2007) also briefly considered how his criteria could be adapted to return measurements of each criterion in the range [0,1], rather than Boolean values, although Ritchie's main presentation of the criteria is as statements which generate Boolean values. This alternative usage gives slightly more information, but the issues of interpreting these criteria's contribution to overall creativity still remain.

when applying Ritchie's criteria, probably due to the formal and abstracted presentation of the criteria. In Colton's creative tripod, all three tripod qualities are treated equally in previous examples (including those in Colton (2008)) regardless of their contribution in a specific creative domain and no further qualities can be introduced into the tripod framework.

Usability of the methodology Less information needed to be collected for Colton's creative tripod than for the other methodologies, taking less time to collect. Coupled with the informal nature of performing creativity evaluation with the tripod framework, Colton's creative tripod emerged as the most easy-to-use of the methodologies evaluated. Data collection for the other methodologies was of a similar magnitude, although data analysis for Ritchie's criteria was slightly more involved and more specialist than the other methodologies, requiring a specific understanding of the criteria.

Feedback reflected on the volume of data generated by using the components as a base model of creativity, as recommended for SPECS. If SPECS is applied without using the Jordanous (2012b) components as the basis for the adopted definition of creativity, then SPECS becomes more involved and more demanding in terms of researcher effort, negatively affecting its usability. Hence the recommendation in Jordanous (2012b) for using the components within SPECS (i.e. SPECS+cc) becomes further strengthened.

One issue is with who/what performs evaluation, and what effect that has on how usable the evaluation methodology. Using external evaluators increases the time demands of the experiment in the human opinion surveys, as this requires studies to be carried out and introduces extra work to be done such as planning experiments for participants or applying for ethical clearance for conducting experiments with people. While the use of external evaluators is not a formal requirement for the SPECS+cc methodology - indeed evaluation can be performed using quantitative tests rather than subjective judgements if deemed most appropriate - the accompanying commentary to SPECS+cc strongly encourages researchers to use independent evaluation methods in order to capture more independent and unbiased results (Jordanous 2012b). In the application of SPECS+cc that is being reviewed here, external judges were consulted to give feedback on the creative systems being evaluated. Hence SPECS+cc in this case is subject to similar criticisms, in terms of ease of use, as when conducting opinion surveys. These extra demands are not necessarily encountered when performing evaluation as recommended using Colton's tripod, Ritchie's criteria, or FACE evaluation, where no specific demands or recommendations are made for evaluation to be performed independently of the project team behind the creative software. It is important to acknowledge, though, that should independent evaluation be sacrificed in order to make an evaluation methodology more useful, there is a worrying knock-on effect, in terms of potential biases being introduced if evaluation is not being performed by independent evaluators.

Generality SPECS+cc, Colton's tripod and to some extent, Ritchie's criteria and the human opinion surveys, could all be applied to different types of system, providing that the system produces the appropriate information relevant to the individual methodologies.[7] Ritchie's criteria cannot be applied to systems that produce no tangible outputs, making this approach less generally applicable across creative systems. There is also some question of whether opinion surveys could be carried out for evaluating all types of creativity, particularly where creativity is not manifested outwardly in production of output, affecting the generality of opinion surveys.

Overall comparisons Considering all the observations made in this paper from the perspective of the five meta-evaluation criteria presented in this paper, SPECS+cc performed well in comparison with the other evaluation methodologies on its faithfulness in modelling creativity. SPECS+cc also performed better than Ritchie's criteria for usefulness and correctness and produced larger quantities of useful feedback than Colton's creative tripod (because less information was collected for Colton's creative tripod). A consequence of the information collection meant that Colton's creative tripod was the easiest to use of the methodologies evaluated.

Somewhat counterintuitively, all the methodologies were more likely to generate correct results compared to the surveys of human opinion. A number of participants in the opinion surveys reported that they evaluated systems based on factors other than creativity, due to difficulties in evaluating creativity of the Case Study systems without a definition of creativity to refer to. There is also some question of whether human opinion surveys could be carried out for evaluating all types of creativity (particularly where creativity is not manifested outwardly in copious production of output); this affects the general applicability of using opinion surveys. Reliance on the existence of output examples also affects the usability and generalisability of Ritchie's criteria.

Conclusions

Several evaluation methods were applied to three musical improvisation systems. Human opinion was consulted to try and capture a 'ground truth' for creativity evaluation (Zhu, Xu, and Khot 2009). Four key existing methodologies for computational creativity were also applied (Ritchie 2007; Colton 2008; Colton, Charnley, and Pease 2011; Jordanous 2012b, Ritchie's criteria, the Creative Tripod, the FACE model and the SPECS+cc methodology, respectively). Results were compared; it was noted that few 'right answers' or 'ground truths' for creativity were found.

For the purposes of progressing in research, learning from advances and improving what has been done, how well did each evaluation methodology perform? To assist in answering this question, external evaluation was solicited from the authors of the evaluated musical improvisation systems and one other researcher with interests in creative musical improvisation systems.

[7] This is illustrated further in Case Study 2 in Jordanous 2012a.

Five criteria were identified from relevant literature sources for meta-evaluation of important aspects of the evaluation methodologies:

- Correctness
- Usefulness
- Faithfulness as a model of creativity
- Usability of the methodology
- Generality

The methodologies were compared based on the external evaluators' feedback concerning the evaluations performed on their system and the comparative feedback generated by each methodology considered so far. Further comments could be made using the meta-evaluation criteria, based on detailed study of the methodologies themselves.

These results are too small in number to be a comprehensive evaluation but they do help to give us some feedback on the compared methodologies. The results showed that SPECS+cc and Ritchie's empirical criteria compared favourably to the other methodologies overall. SPECS+cc performed well on most of the five meta-evaluation criteria, though the volume of data produced by SPECS+cc raised questions on SPECS+cc's usability compared to more succinct presentations. Colton's creative tripod was the easiest to use although there were some concerns about the generality of the tripod across creative domains and its faithfulness as a general model of creativity. Ritchie's criteria were considered accurate but there were usability issues with the abstract nature of the criteria and accompanying function definitions. The FACE model was considered quite user friendly but perhaps limited in how it could incorporate aspects of creativity that were important to the system domain but outside of the face model. Each of the evaluation methodologies proved to be an improvement (in at least some ways) over the approach of simply asking people's opinions on how creative the systems were.

The development of creativity evaluation methods is clearly a key current area of interest in the computational creativity research community, as partly illustrated by the prominent inclusion of requests for papers on evaluation, in the call for papers for ICCC 2014. The five meta-evaluation criteria offered in this paper are taken from a cross-disciplinary review of good practice in evaluation of areas relevant to computational creativity research. These five criteria help us to contrast different evaluation methodologies against each other

Acknowledgments

Thanks to Alison Pease, Steve Torrance and Nick Collins for helpful comments during the formulation of these ideas. Also thanks to Al Biles, Bob Keller and George E. Lewis for willingly offering their time and helpful comments as evaluators for this work, and to the three anonymous reviewers for their useful remarks on the original version of this paper.

References

Blanke, T. 2011. *Using Situation Theory to evaluate XML retrieval*. Dissertations in Database and Information Systems. Heidelberg, Germany: IOS Press.

Colton, S.; Gow, J.; Torres, P.; and Cairns, P. 2010. Experiments in objet trouvé browsing. In *Proceedings of the International Conference on Computational Creativity*.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational Creativity Theory: The FACE and IDEA descriptive models. In *Proceedings of the 2nd International Conference on Computational Creativity*, 90–95.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *Proceedings of AAAI Symposium on Creative Systems*, 14–20.

Gillick, J.; Tang, K.; and Keller, R. M. 2010. Machine learning of jazz grammars. *Computer Music Journal* 34(3):56–66.

Jordanous, A., and Keller, B. 2014. What makes musical improvisation creative? *Journal of Interdisciplinary Music Studies* Forthcoming.

Jordanous, A. 2011. Evaluating evaluation: Assessing progress in computational creativity research. In *Proceedings of the Second International Conference on Computational Creativity (ICCC-11)*.

Jordanous, A. 2012a. *Evaluating Computational Creativity: A Standardised Procedure for Evaluating Creative Systems and its Application*. Ph.D. Dissertation, University of Sussex, Brighton, UK.

Jordanous, A. 2012b. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.

Kaufman, J. C. 2009. *Creativity 101*. The Psych 101 series. New York: Springer.

Kazai, G., and Lalmas, M. 2005. Notes on what to measure in INEX. In *INEX 2005 Workshop on Element Retrieval Methodology*.

Pease, A.; Winterstein, D.; and Colton, S. 2001. Evaluating machine creativity. In *Proceedings of Workshop Program of ICCBR-Creative Systems: Approaches to Creativity in AI and Cognitive Science*, 129–137.

Rhodes, M. 1961. An analysis of creativity. *Phi Delta Kappan* 42(7):305–310.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.

Sloman, A. 1978. *The computer revolution in philosophy*. Hassocks, Sussex: Harvester Press.

Thagard, P. 1988. *Computational Philosophy of Science*. Cambridge, MA: MIT Press.

Zhu, X.; Xu, Z.; and Khot, T. 2009. How creative is your writing? a linguistic creativity measure from computer science and cognitive psychology perspectives. In *Proceedings of NAACL HLT Workshop on Computational Approaches to Linguistic Creativity (ACL)*, 87–93.

Assessing Progress in Building Autonomously Creative Systems

Simon Colton*, Alison Pease†, Joseph Corneli*, Michael Cook* and Teresa Llano*
*Computational Creativity Group, Department of Computing, Goldsmiths, University of London, UK
†School of Computing, University of Dundee, UK
ccg.doc.gold.ac.uk

Abstract

Determining conclusively whether a new version of software creatively exceeds a previous version or a third party system is difficult, yet very important for scientific approaches in Computational Creativity research. We argue that software product and process need to be assessed simultaneously in assessing progress, and we introduce a diagrammatic formalism which exposes various timelines of creative acts in the construction and execution of successive versions of artefact-generating software. The formalism enables estimations of progress or regress from system to system by comparing their diagrams and assessing changes in quality, quantity and variety of creative acts undertaken; audience perception of behaviours; and the quality of artefacts produced. We present a case study in the building of evolutionary art systems, and we use the formalism to highlight various issues in measuring progress in the building of creative systems.

Introduction

Creativity, we believe, relates to a perception that others have of certain behaviours exhibited by some person or system, rather than an inherent property of people or software: in this sense it is a secondary quality. Moreover, we believe that, just as the endless debates about "is it art?" fuel innovation in the arts, the endless debates about "is it creative?" are a force for good: they drive forward creative practices and Computational Creativity research. A longer discussion of this philosophical position is given in (Colton et al. 2014), and an exposition of creativity as being *essentially contested* (Gallie 1956) is given in (Jordanous 2012).

In such a context of energetic and subjective debate about creativity, it has been difficult to derive systematic approaches to assessing progress in the building of software for creative purposes. One main issue has been the cross-purposes of the creativity project(s) for which software is developed. A useful analogy with the notions of weak and strong AI has arisen recently in Computational Creativity research. Focusing on software which generates artefacts such as poems, paintings or games, we can say that *weak Computational Creativity* objectives emphasise the production of increasingly higher valued artefacts, whereas *strong Computational Creativity* objectives emphasise increasing the perception of creativity people have of the system. This is similar to the distinction put forward in (al-Rifaie and Bishop 2012). In many projects, there are both strong and weak objectives, and often they are not complementary. For instance, increasing autonomy in software may lead simultaneously to higher perception of creativity and lower value artefacts being produced. This is described as the *latent heat* problem in (Colton and Wiggins 2012), and is analogous to U-shaped learning, where to get better, we first have to get worse.

The objectives for a project usually influence the assessment methods employed. In particular, to assess progress with respect to weak objectives, it makes sense to evaluate the quality of the artefacts produced. In contrast, for strong objectives, it makes more sense to assess what software actually does and how and why people perceive it as creative or not. To this end, in (Colton, Pease, and Charnley 2011) we introduced the FACE descriptive model to formalise descriptions of the creative acts undertaken by software, and the IDEA model to formalise the impact those creative acts might have on people. Subsequent attempts to use these models to describe particular systems have highlighted another major issue: the assignment of programmer/software ownership of creative acts. Along with other issues in applying it to describe systems, we have found the FACE model to be inadequate for fully capturing the interplay of creative acts between programmer and program in this respect.

We describe here the next stage of our formalism for capturing notions of progress in building creative systems. We first provide a potted history of how progress has been evaluated in Computational Creativity research, and lay out some intuitive notions of progress. Given our philosophical and practical standpoints, we place less emphasis on asking whether artefacts are 'better' than previously. We also avoid direct questions about 'creativity' in computational systems. Instead, we integrate (i) aspects of the FACE and IDEA models (ii) objective measures of quality, quantity and variety of creative acts and (iii) audience perceptions of software behaviour and quality of output. We present a two-stage method for estimating whether obvious or potential progress or regress has occurred when building a new system. This involves diagrammatically capturing various timelines in the building and execution of a system, then comparing diagrams. We use the method to describe progress of an evolutionary art system, leading to a general discussion about how the approach could be used in practice. We conclude by describing future directions for this formalism.

Background in Assessing Creative Progress

The assessment of progress in building creative systems has been a bespoke and multi-faceted endeavour, driven by various, often opposing objectives ranging from understanding human creativity to practical generation of artefacts to the raising of philosophical questions. The majority of practical researchers who engineer and test software joined the Computational Creativity field with objectives in the weak sense of getting software to produce quality artefacts. Hence the first way in which progress was assessed was Boolean: if software reliably produces artefacts of a particular type, then this is progress over software which was unreliable or unable to produce artefacts of the required form.

In such a context, Turing-style discrimination tests indicated a particularly strong milestone: if certain artefacts – usually hand-selected – looked/sounded so like human-authored counterparts that observers couldn't tell the difference, progress had certainly been made. This approach was pioneered by (Pearce and Wiggins 2001) who were one of the first to emphasise the importance and role of evaluation in Computational Creativity, and to propose a concrete way of applying Popperian falsificationism. However, despite them urging caution at depending on the discrimination test to evaluate creativity, direct comparison of human produced and computer generated artefacts has frequently been used to assess progress. We further criticised such Turing-style tests in Computational Creativity, for, among other reasons, encouraging naïvety in software and the generation of pastiches (Pease and Colton 2012). Moreover, we question whether this methodology, while beneficial for short-term scientific progress, is actually detrimental to the longer-term goal of embedding creative software in society (Colton et al. 2014). The work of (Ritchie 2007) was an important step away from simplistic discrimination tests, establishing an approach to assessing the value of artefacts according to their *novelty*, *typicality*, and *quality* within a genre. A number of practitioners have used this approach to compare and contrast their systems, e.g., (Pereira et al. 2005).

As the field matured, attention moved from *mere generation* to programs able to assess, critique and select from their output. Often searching large spaces, software was required to find the best artefacts using mathematically derived or machine-learned aesthetic/utilitarian calculations (Wiggins 2006). If a later version of software – with more sophisticated internal assessment techniques – was able to produce higher yields of higher quality artefacts when assessed externally, then clear progress had been made.

Audience perceptions of software became a focus, as the field further matured. Jordanous used methods from linguistics to determine how people are using the word 'creativity', and which other concepts are associated with it, and then used crowdsourcing techniques to evaluate a creative system in terms of the associated concepts (Jordanous 2012). As a complement to Jordanous's work in which she tried to *capture* society's perception of creativity, researchers began investigating ways to *influence* people's perception of creativity in software. Software assessing its own work made it *appear* more intelligent, and *seem* more creative. This led to the engineering of software that *framed* its processes and outputs by producing titles, commentaries and other material. (Charnley, Pease, and Colton 2012) propose that this may increase perception of creativity, and audiences would possibly appreciate the artefacts produced more. Studying audience perceptions of creativity in software opened many research avenues, but raised an important problem: that the original product-based assessment methods no longer capture all intuitions of what constitutes progress in the field.

From a strong perspective, some researchers, including ourselves, are not content to accept the underlying assumption of product-based evaluation methods: if better artefacts are produced, the software must have been improved, hence people will project higher perceptions of creativity onto the software and progress will have been made. As mentioned previously, the main problem here is that increasing autonomy – which must happen if strong objectives are to be met – can decrease artefact value. Conversely, when the objectives of a project are weak, it is perfectly natural to decrease software autonomy to produce artefacts of presentation quality, especially when a concert/exhibition is looming, but this is unlikely to increase any perceptions of creativity.

Concentrating on understanding perceptions of software creativity by the general public, we introduced the *creativity tripod* in (Colton 2008b) as three types of behaviours which were necessary (but not necessarily sufficient) for software to avoid being labelled as 'uncreative'. We proposed that people are influenced by their understanding of what software does when assessing its output. We argue that it is easy to ascribe uncreativity to software which is not simultaneously seen as *skillful*, *appreciative* and *imaginative*.

Focusing on assessment of progress by peers, we introduced the FACE and IDEA descriptive models in (Colton, Pease, and Charnley 2011) and (Pease and Colton 2011). The FACE model categorises generative acts by software into those at (g)round level, during which base objects are produced, and (p)rocess level, during which methods for generating base objects are produced. These levels are subdivided by the types of objects/processes they produce: F_g denotes a generative act producing some framing information, A_g denotes an act producing an aesthetic measure, C_g denotes an act producing a concept and E_g denotes an act producing an example of a concept. Generative acts producing new processes are defined accordingly as F_p, A_p, C_p and E_p. Tuples of generative acts are compiled as *creative acts*, and various calculations and recommendations are suggested in the model with which to compare creative systems. We developed the IDEA model so that creative acts and any impact they might have could be properly separated. We defined various stages of software development and used an ideal audience notion, where people are able to quantify changes in well-being and the cognitive work required to appreciate a creative act and the resulting artefact/process.

We have arrived at a very observer-centric situation in the assessment of progress towards creative systems, in which progress can only be measured using feedback from independent observers about both the quality of artefacts produced and their perceptions of creativity in the software. Unfortunately, the majority of researchers develop software using only themselves as an evaluator, because observer-

based models are too time-consuming to use on a day-to-day progress. These informal in-house evaluation techniques generally do not capture the global aims of the research project, or of the field (e.g. producing culturally important artefacts and/or convincing people that software is acting in a creative fashion). In many cases, systems are presented as feats of engineering, with little or no evaluation at all (Jordanous 2012). We argue that assessing *progress* is inherently a process-based problem. We focus here on modeling diachronic change across multiple levels.

A Formal Assessment of Progress

We combine the most useful aspects of the IDEA and FACE models, an enhanced creativity tripod, and aspects of assessing artefact value into a diagrammatic formalism for evaluating progress in the building of creative systems. We focus on the creative acts that software performs, the artefacts it produces and the way in which audiences perceive it and consume its output. We simplify by assuming a development model where a single person or team develops the software, with various major points where the program is sufficiently different for comparisons with previous versions. We aim for the new formalism to be used on a daily basis without audience evaluations, to determine short term progress, but for it also to enable fuller audience-level evaluations at the major development points. We also aim for the formalism to help determine progress in projects where there are both weak and strong objectives. We found that the original FACE model didn't enable us to properly express the process of building and executing generative software. Hence another consideration for our new model is that it can capture various timelines both in the development and the running of software in such a way that it is obvious where the programmer contributed creatively and where the software did likewise.

With the above aims in mind, we envisage a scenario where we are comparing two versions of creative software $v1$ and $v2$. At the highest level, we split the assessment method into a two stage process as follows:

---

1. Diagrams are drawn for both $v1$ and $v2$ which capture the interplay of programmer and program behaviours as timelines during both the development phase and the runtime execution of both versions of the software.

2. The diagrams for $v1$ and $v2$ are compared by an audience to determine if the second system represents progress over the first in terms of process. Similarly, the output from $v1$ and $v2$ is compared, to see if progress has been made.

Stage 1: Diagrammatic Capture of Timelines

Taking a realistic but abstracted view of generative software development and deployment, we identify four types of timeline. Firstly, generative programs are developed in **system epochs**, with new versions being regularly signed off. Secondly, each process a program undertakes will have been implemented during a **development period** where creative acts by programmer and program have interplayed.

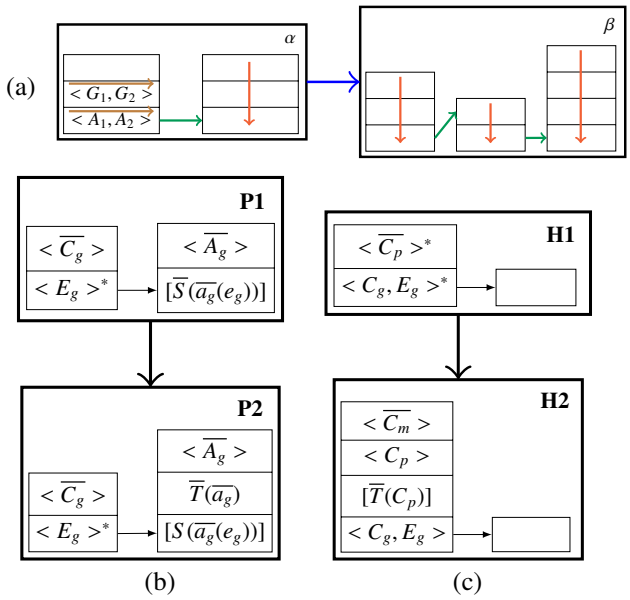


Figure 1: *(a) Key showing four types of timelines (b) Progression of a poetry system (c) Progression of the HR system.*

Thirdly, at run-time, data will be passed from process to process in series of creative and administrative **subprocesses** performed by software and programmer. Finally, each subprocess will comprise a sequence of generative or administrative **acts**. We capture these timelines diagrammatically, highlighted with coloured arrows in Figure 1(a). The blue arrow from box α to β represents a change in epoch at system level. The red arrows overlapping a *process stack* represent causal development periods. The green arrows represent data being passed from one subprocess to another at run-time. The brown arrows represent a series of generative/administrative acts which occur within a subprocess.

Inside each subprocess box is either a < creative act > from the FACE model (i.e., a sequence of generative acts), or an [administrative act] which doesn't introduce any new concept, example, aesthetic or framing information/method. Administrative acts were not originally described in the FACE model, but we needed them to describe certain progressions of software. For our purposes here, we use only T to describe a translation administrative act often involving programming, and S to describe when an aesthetic measure is used to select the best from a set of artefacts. To add precision, we indicate the output from which generative act the administrative routine is applied, and to which examples a ground aesthetic is applied. To enable this, we employ the FACE model usage of lower-case letters to denote the output from the corresponding upper-case generative acts. We extend the FACE notion of (g)round and (p)rocess level generative acts with (m)eta level acts during which process generation methods are invented. As in the original description of the FACE model, we use bar notation to indicate that a particular act was undertaken by the programmer. We use a superscripted asterisk (*) to point out repetition.

As a simple example diagram, Figure 1(b) shows the progression from poetry generator version P1 to P2. In the

first version, there are two process stacks, hence the system works in two stages. In the first, the software produces some example poems, and in the second the user chooses one of the poems (to print out, say). The first stack represents two timesteps in development, namely that (a) the programmer had a creative act $< \overline{C_g} >$ whereby he/she came up with a concept in the form of some code to generate poems, and (b) the programmer ran the software to produce poems in creative acts of the form $< E_g >^*$. The second stack represents the user coming up with an idea for an aesthetic, e.g., much rhyming, in creative act $< \overline{A_g} >$, and then applying that aesthetic $\overline{a_g}$ him/herself to the examples, e_g, produced by the software, in the *selection* administrative act $[\overline{S}(\overline{a_g}(e_g))]$, which maps the aesthetic $\overline{a_g} : \{e_g\} \rightarrow [0, 1]$ over the generated examples, and picks the best one. In the $P2$ version of the software, the programmer undertakes *translation* act $[\overline{T}(\overline{a_g})]$, writing code that allows the program to apply the rhyming aesthetic itself, which it does at the bottom of the second stack in box $P2$.

Figure 1(c) shows a progression in the HR automated theory formation system (Colton 2002) which took the software to a meta-level, as described in (Colton 2001). HR operates by applying production rules which invent concepts that categorise and describe input data. Each production rule was invented by the programmer during creative acts of the type $< \overline{C_p} >$, then at run-time, HR uses the production rules to invent concepts and examples of them in $< C_g, E_g >^*$ acts. In the meta-HR version, during the $< \overline{C_m} >$ creative act, the programmer had the idea of getting HR to form theories about theories, and in doing so, generate concept-invention processes (production rules) in acts of the form $< C_p >$. The programmer took meta-HR's output and translated $[\overline{T}(C_p)]$ it into an implemented production rule that HR could use, which it does at the bottom of the stack in box H2.

Stage 2: Comparing Diagrams and Output

In both simple cases of Figure 1, it is clear that progress has been made in the strong sense, but not clear in the weak sense, as the output could easily be degraded by the more sophisticated processing of the systems. The diagrams help us to capture the creative interplay between software and programmer at design time and run time. However, given that the ultimate aim of both strong and weak projects is to impress audiences with process and product, any assessment of progress must be done in a context of audience evaluation. However, as mentioned previously, audience evaluation is too expensive to help assess progress on a day to day basis. Hence, it seems sensible for the programmer to step in and act as a proxy for a perceived audience: we advocate the programmer putting themselves in the position of the type of person they would expect to form their audience, and answer questions about the products and processes accordingly.

Examining the transition from one diagram to another should provide some shortcuts to estimate audience reactions, especially when there are strong project objectives. In particular, as with the original FACE model, the diagrams make it obvious where creative or administrative responsibility has been handed over to software, namely where an act which used to be barred has become unbarred, i.e., the same type of generative act still occurs, but it is now performed by software rather than programmer. This happened when the $\overline{S}$ became an S in Figure 1(b) and when the $\overline{C_p}$ became a C_p in Figure 1(c). At the very least in these cases, an unbiased observer would be expected to project more autonomy onto the software, and so progress in the strong sense has likely happened. In addition, the diagrams make it obvious when software is doing more processing in the sense of having more stacks, bigger stacks or larger tuples of acts in the stack entries. Moreover, the diagrams make it clear that more varied or higher-level creative acts are being performed by the software – again, this was one of the benefits of the original FACE model. Both of these have the potential to convince audience members that software is being more sophisticated with respect to various behaviours described below, and hence can be a shorthand for progress.

When dealing with actual external evaluation, where people don't know what software does, we suggest that the diagrams above (and verbalisations/simplifications of them) can be used to describe to audiences what the software and what the programmer have done in a project. In this way, using also their judgements about the artefacts produced, people can make fully informed decisions in evaluation studies. As a general philosophical standpoint, we suggest not asking people if they believe software is behaving creatively, but rather concentrating on whether they perceive the software as acting *uncreatively*. Our argument for this is that the concept of creativity is essentially contested (Gallie 1956), hence, no matter how sophisticated our software gets, we should not expect consensus on such matters. However, we have found that people agree much more on notions of uncreativity: if a program doesn't exhibit behaviours onto which certain words like *intentionality* can be projected, then it is very easy to condemn it as being uncreative.

Hence, we advocate not asking a set of questions from which we can conclude that an audience member thinks that software is creative, but rather asking questions from which we can determine whether they think that software is acting uncreatively. It may seem like rather a negative admission, but we believe that the best way to get people to accept software as being creative is for them to eventually realise that there is no good reason to call it uncreative. Even then, people would be perfectly at liberty to say that while software is not uncreative, it is not creative either: creativity and uncreativity do not appear to be exact opposites. With this in mind, we have boiled down audience evaluation of behaviour to asking people whether they would project certain words onto software in reaction to understanding what it did in the context of a particular project. We then tentatively conclude that they believe the software is uncreative if they don't project onto it some or all of these words, as originally intended in the creativity tripod proposition (Colton 2008b).

In the five years since the introduction of the creativity tripod, we have slowly added additional behaviours which we have found to be important in the perception of creativity in software. That is, for people to take seriously software as being not uncreative, we believe it needs to exhibit behaviours onto which people can meaningfully project (at least) these

Product change	Process change	Weak	Strong
Up	Up	OP	OP
Up	Down	PP	PR
Up	Same	OP	PP
Down	Up	PR	PP
Down	Down	OR	OR
Down	Same	OR	PR
Same	Up	PP	OP
Same	Down	PR	OR
Same	Same	PP	PP

Table 1: Guidelines for using change in evaluation of product and process in gauging (O)bvious or (P)otential (P)rogress or (R)egress, in both weak and strong agendas.

eight words: skill, appreciation, imagination, learning, intentionality, accountability, innovation, subjectivity and reflection. We have found that assessing the level of projection of these words onto the behaviours of software can help us to gauge people's opinions about (the lack of) important higher-level aspects of software behaviour, such as autonomy, adaptability and self-awareness.

The method we suggest for estimating progress from version $v1$ of a creative system to version $v2$ is to: (a) show audience members the diagrams for $v1$ and $v2$ as above, and explain the acts undertaken by the software, then (b) show audience members the output from $v1$ and $v2$, and (c) ask each person to compare the pair of product and process for $v1$ with that of $v2$. A statistical analysis could then be used to see whether the audience as a whole evaluates the output as being better, worse or the same, and whether they think that the processing is better, worse or the same in terms of the software seeming less uncreative. This takes into account the phenomenon described in (Colton 2008b) whereby the process can influence value judgements for artefacts.

To use this analysis to estimate progress, it's important to first prioritise objectives for the project locally in terms of strong and weak agendas. Then, taking the audience evaluation of change in output and in process, we suggest using the guidelines in Table 1. Here, we have stipulated that certain evaluation pairs indicate obvious progression (OP) or obvious regression (OR). For instance, in the weak sense, when the evaluation of output goes up and the evaluation of process increases or stays the same, it seems clear to indicate obvious progress. Other cases are not so clear-cut, for instance when evaluation of artefacts goes up, but evaluation of process goes down. In this case, we suggest that this is potential progress (PP) in a weak agenda, and potential regress (PR) in a strong agenda. In such cases, we give our judgements for whether it is likely, after more development, that $v2$ will be viewed retrospectively as a progressive success or a step backwards. Note that we have tended to be optimistic, e.g., when evaluation of output and process stay the same, we say that this is potential progress in both weak and strong agendas. Note also that this table is meant to be used flexibly, possibly in a context of more fine grained analysis. For instance, the focus of a subproject might be to increase audience perception of intentionality, and if this increases while audience perception of the value of the process as a whole reduces, it should still be seen as progress.

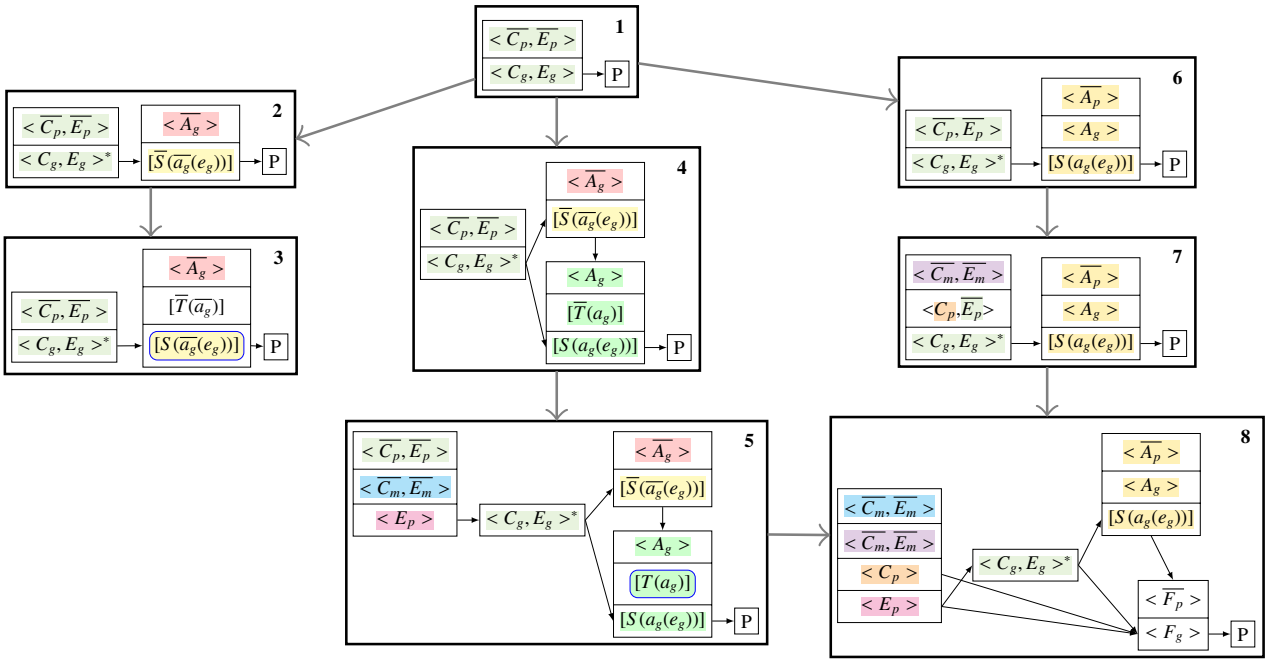
A Case Study in Evolutionary Art

Evolutionary art – where software is evolved which can generate abstract art – has been much studied within Computational Creativity circles (Romero and Machado 2007). Based on actual projects which we reference, we hypothesise here the various timelines of progress that could lead from a system with barely any autonomy to one with nearly full autonomy. Figure 2 uses our diagrammatic approach to capture three major lines of development, with the final (hypothetical) system in box 8 representing finality, in the strong sense that the software can do very little more creatively in generating abstract art. Since features from earlier system epochs are often present in later ones, we have colour-coded individual creative acts as they are introduced, so the reader can follow their usage through the systems. If an element repeats with a slight variation (such as the removal of a bar), this is highlighted. The figure includes a key, which describes the most important creative and administrative acts in the systems. Elements in the key are indexed with a dot notation: *system.process-stack.subprocess* (by number, from left to right, and top to bottom, respectively). System diagrams have repetitive elements, so that the timelines leading to its construction and what it does at run-time can be read in a stand-alone fashion.

Following the first line of development, system 1 of Figure 2 represents an entry point for many evolutionary art systems: the programmer invents ($\overline{C_p}$) (or borrows) the concept formation process of crossing over sets of mathematical functions to produce offspring sets. He/she also has an idea ($\overline{E_p}$) for a *wrapper* routine which can use such a set of functions to produce images. He/she then uses the program to generate (C_g) a set of functions and employ the wrapper to produce (E_g) an image which is sent to the (P)rinter. The crossover and subsequent image generation is repeated multiple times in system 2, and then the programmer – who has invented ($\overline{A_g}$) their own aesthetic – chooses a single image to print. In system 3, as in the poetry example above, the programmer translates their aesthetic into code so the program can select images. This is a development similar to that for the NEvAr system (Machado and Cardoso 2002).

Following the second line of development, in system 4, the programmer selects multiple images using his/her own aesthetic preferences, and these become the positives for a machine learning exercise as in (Li et al. 2013). This enables the automatic invention (A_g) of an aesthetic function, which the programmer translates by hand $\overline{T}(a_g)$ from the machine learning system into the software, as in (Colton 2012), so the program can employ the aesthetic without user intervention. In system 5, more automation is added, with the programmer implementing their idea ($\overline{C_m}$) of getting the software to search for wrappers, then implementing this ($\overline{E_m}$), so that the software can invent (E_p) new example generation processes for the system.

Following the final line of development, in system 6, we return to aesthetic generation. Here the programmer has the idea ($\overline{A_p}$) of getting software to mathematically invent fitness functions, as we did in (Colton 2008a) for scene generation, using the HR system (Colton 2002) together with The Paint-



ID	Event	Explanation
1.1.1	$\overline{C_p}$	The programmer invents the idea of crossing over two sets of mathematical functions to produce a new set of mathematical functions.
1.1.1	$\overline{E_p}$	The programmer implements a wrapper method that takes a set of mathematical functions and applies them to each (x, y) co-ordinate in an image to produce an RGB colour.
1.1.2	C_g	The software generates a new set of functions by crossing over two pairs of functions.
1.1.2	E_g	The software applies these functions to the (x, y) co-ordinates of an image, to produce a piece of abstract art.
2.2.1	$\overline{A_g}$	The programmer had in mind a particular aesthetic (symmetry) for the images.
2.2.2	$\overline{S}(\overline{a_g}(e_g))$	The programmer uses his/her aesthetic to select a preferred image for printing.
3.2.2	$\overline{T}(\overline{a_g})$	The programmer took their aesthetic and turned it into code that can calculate a value for images.
3.2.3	$S(\overline{a_g}(e_g))$	The software applies the aesthetic to select one of a set of images produced by crossover and the wrapper.
4.3.1	A_g	The software uses machine learning techniques to approximage the programmer's aesthetic.
4.3.2	$\overline{T}(a_g)$	The programmer hand-translates the machine learned aesthetic into code.
4.3.3	$S(a_g(e_g))$	The software applies the new aesthetic to choosing the best image from those produced.
5.1.2	$\overline{C_m}$	The programmer has the idea of getting the software to search through a space of wrapper routines.
5.1.2	$\overline{E_m}$	The programmer implements this idea.
5.1.3	E_p	The software invents a new wrapper.
5.4.2	$T(a_g)$	The software translates the machine-learned aesthetic itself into code.
6.2.1	$\overline{A_p}$	The programmer has the idea of getting the software to invent a mathematical fitness function.
6.2.2	A_g	The software invents a novel aesthetic function.
6.2.3	$S(a_g(e_g))$	The software selects the best artefact according to its aesthetic function.
7.1.1	$\overline{C_m}$	The programmer has the idea of getting the software to invent and utilise novel combination techniques for sets of functions, generalising crossover.
7.1.1	$\overline{E_m}$	The programmer implements this idea so that the software can invent new combination techniques.
7.1.2	C_p	The software invents a novel combination technique.
8.4.1	$\overline{F_p}$	The programmer has the idea of getting the software to produce a commentary on its process and artwork by describing its invention of a new aesthetic, combination method and wrapper.
8.4.2	F_g	The software produces a commentary about its process and product.

Figure 2: *The progression of an evolutionary art program through eight system epochs.*

ing Fool (Colton 2012b). In system 7, the programmer realises ($\overline{C_m}$) that crossover is just one way to combine sets of functions, and gives ($\overline{E_m}$) the software the ability to search a space of combination methods (C_p). The software does this, and uses the existing wrapper to turn the functions into images. System 8 is the end of the line for the development of the software, as it brings together all the innovations of previous systems. The software invents aesthetic functions, innovates with new concept formation methods that combine mathematical functions, and generates new wrappers which turn the functions into images. Finally, the programmer has the idea ($\overline{F_p}$) of getting the software to write commentaries, as in (Colton, Goodwin, and Veale 2012), about its processing and its results, which it does in generative act F_g.

Tracking how the system diagrams change can be used to estimate how audiences might evaluate the change in processing of the software, in terms of the extended creativity tripod described above. Intuitively, each system represents progress from the one preceding it, justified as follows:

1 → 2: $< C_g, E_g > \rightarrow < C_g, E_g >^*$

Simple repetition means that the software has more *skill*, and the introduction of independent user selection shouldn't change perceptions about *autonomy*.

2 → 3: $\overline{S} \rightarrow S$

By reducing user intervention in choosing images, the software should appear to have more *skill* and *autonomy*.

1 → 4: Introduction of A_g and $S(a_g(e_g))$ acts

Machine learning enables the generation of novel aesthetics (albeit derived from human choices), which should increase perception of *innovation*, *appreciation* and *learning*, involving more varied creative acts.

4 → 5: Introduction of an E_p act, $\overline{T} \rightarrow T$

Wrapper generation increases variety of creative acts, and may increase perception of *skill* and *imagination*.

1 → 6: Introduction of A_g and $S(a_g(e_g))$ acts

The software has more variety of creative acts, and the invention and deployment of its own aesthetic – this time, without any programmer intervention – should increase perception of *intentionality* in the software.

6 → 7: Introduction of a C_p act

Changes in the evolutionary processes should increase perceptions of *innovation* and *autonomy*.

5, 7 → 8: Introduction of an F_g act

Framing its work should increase perceptions of *accountability* and *reflection*.

With all strands brought together, the programmer does nothing at run-time and can contribute little more at design time. The software exhibits behaviours onto which we can meaningfully project words like skill, appreciation, innovation, intentionality, reflection, accountability and learning, which should raise impressions of autonomy, and make it difficult to project uncreativity onto the software.

Discussion

Capturing what programmers and software do creatively over long periods and during complicated program executions is difficult and open to variability. The systems in the above case study could easily have been interpreted and presented differently. In essence, we have provided some tools for presenting software development in terms of creative acts, and suggested a mechanism for turning audience perceptions into estimates of progress. We advise flexible application in both cases. In particular, the difference between potential progress and potential regress is quite subtle. Both mean that it is too early to determine whether progress or regress has been made, and the programmer should proceed with caution: the former suggesting cautious optimism and the latter, cautious pessimism. Practically speaking, the programmer may want to review longer term goals, archive previous versions, and/or clarify research directions.

Our approach is currently more tailored to capturing progress in software behaviour than its output. We would understand some resistance to the approach, particularly from researchers with agendas for Computational Creativity in the weak sense. For example, if product evaluations remain the same, yet processing evaluations go up, this is presumably because the software is performing more sophisticated routines. From a weak perspective, the simpler version of the software clearly has advantages, as it produces the same results in a more understandable way. In certain application domains, for instance mathematical discovery, where aesthetics like truth are of paramount importance, a simpler method for finding a result is usually preferred. While reducing complexity of processing normally requires considerable invention or intervention, unless such invention is done by the software itself, the resulting simplicity would tend to increase perceptions of uncreativity in software, regardless (or, indeed, because of) how easy it is to understand what it has done.

Our approach is also more tailored towards capturing progress from version to version of the same software than to comparing different programs. However, we have used the formalism to compare systems in the same application domains, such as mathematical discovery systems AM (Lenat 1976) and HR (Colton 2002), and various poetry and art generators. The comparative approach works somewhat here, because it was possible to compare diagrams meaningfully to suggest where one system would likely be perceived as an improvement over the other. However, full application of the approach may be difficult as the context for evaluating artefacts (and the processes producing them) can change greatly with small changes in artefact composition. For instance, we recently attempted to compare one-line "What if...?" ideas produced textually by three systems. We found that it was not possible to conceive a fair approach involving an audience to determine which system's artefacts or processes were the best. Fields like Machine Learning have largely homogenised the testing of their systems in a problem-solving paradigm. Given the tacit requirements for software to surprise us through its output and processing, and to innovate on many levels, it seems unlikely that such standardisation could apply in Computational Creativity research.

Related Work

Diagrammatic approaches to software modelling have been extensively studied in the last two decades. The best known example is the Unified Modelling Language (UML), managed by the Object Management Group (OMG), a standard that is widely used to visualize the design of systems (`www.omg.org/spec`). The main objectives of modelling with UML are to represent the architecture of a system, including use cases, deployment, information flow diagrams, etc., and to model system behaviour and data flow via activity diagrams, state machines, sequence diagrams, etc.

Progress at the process level can be modelled with UML by diagramming the steps used to complete a task within the system. However, UML is not typically applied to model progress at the level of system epochs, although two UML diagrams can of course be compared on the basis of the functionality they describe. Some diagrams created using the UML model, such as use case diagrams, enable designers to specify the agents that participate in the development of a system: people, external processes, other systems and the system itself can all be modelled as agents. However, there is no formal notation to distinguish between the different agents, rather, they are simply assigned a label which is meaningful for the system designer. The OMG have also developed other graphical notations specialised for other aspects of systems modelling. For instance, the Business Process Model and Notation (BPMN) is used to model business processes by extending the original activity diagrams of UML. The specific objective of BPMN is to provide a high-level overview of business systems, rather than detailed information about how the system works.

UML diagrams have also been used in the context of formal methods. In particular, the UML-B language (Said, Butler and Snook 2009) enables the modelling of Event-B specifications as UML-like diagrams. Event-B is a formalism based on set theory for the modelling and verification of systems (Abrial 2010). One of the main aspects of Event-B is the use of refinement to handle the complexity of systems at different levels of abstraction. UML-B can be used to diagrammatically model a system at increasing levels of refinement, and system consistency can then be verified through mathematical proof. However, UML-B considers one system at a time, so it is not possible to use this formalism to model creative change as system development progresses.

Using the Event-B formalism, it is possible to model aspects of the environment, such as external systems that affect the behaviour of the modelled system. The aim is to ensure that the designed system will work in harmony with its operating environment. However, there is no clear way to delimit the aspects of the model that are related to the environment and those that are part of the final system. Again, the environment is simply identified by the designer assigning meaningful names to the state representing it. Other related approaches include Z-notation (Spivey 1992), the Vienna Development Method (Jones 1990) and the B-method (Abrial 1996). The objective of these approaches is to verify properties of systems. Progress would be meaningful at the modelling level, i.e., by building models that offer increasing detail (and assurance) about how a given system works.

Petri nets provide a graphical notation used primarily to model systems with concurrency (Girault and Valk 2003). With petri nets, progress at the process level is modelled in the form of state transitions, and data is represented by abstract tokens, with no data values assigned. An extension, called coloured petri nets (Jensen and Kristensen 2009), allow data values to be assigned to tokens. Neither type of petri net is used for modelling changes through versions of a system. Petri nets are an event-based modelling language and representations of agents (such as the programmer or the system) are not included in the formalism.

Conclusions and Future Work

We have presented a new diagrammatic formalism for assessing progress in building creative systems. Our aims were to enable more precise understanding of progress in Computational Creativity in general, and in mapping the progress of particular systems. In doing so, we aimed to bring closer together public/peer appreciation of progress, strong/weak agendas, and day-to-day/milestone progress assessments. The new approach involves producing diagrams of systems that depict creative acts in timelines, which are compared in a context of audience evaluation of process and product. When applied, the formalism captures some intuitive notions, including: quality of artefacts; quantity, level and variety of creative acts performed; and audience perception of software behaviour. To enable better understanding of process, and more informed audience judgements about (un)creativity, the diagrams explicitly separate creative acts coming from the programmer and the program. Even in the absence of audience participation, the diagrams themselves can be used in combination with straightforward assumptions about audience reactions to system design features to perform low-cost estimates of progress in a strong agenda.

We motivated the approach throughout with various philosophical standpoints, as per (Colton et al. 2014), supported by a critical review of the ways in which progress in building creative systems has been measured historically. To highlight the potential for the formalism, we presented a case study where the progress through eight versions of evolutionary art software was mapped and justified.

Our audience evaluation model is far from complete. We plan to employ the criteria specified in (Ritchie 2007), for more fine-grained evaluations of the quality, novelty and typicality of artefacts. We will also import audience reflection evaluation schemes from the IDEA descriptive model, e.g., change in well-being, cognitive effort and emotional responses such as surprise and amusement. We have so far used the diagrammatic approach to fully depict timelines in the building of generative software producing mathematics, visual art, poetry and video games, including dozens of system diagrams (omitted for space reasons). This has worked well, but there are still some subtle improvements required to capture better the functioning of the software at run-time.

(Gabriel and Goldman 2000) describe system development environments with many contributing programmers, and multiple interacting, self-programming, and self-updating distributed systems (Gabriel and Goldman 2006). It would be straightforward to modify our formalism to deal

with multiple agents, for example by turning bars into superscripts. However, this does complicate the notion of progress: if system μ chooses to hand off creative control to system ν, this would amount to changing a superscript – but it's not immediately clear that this should count as progress in the same way that removing bars does. If the agents are considered to be full partners in the creative process, μ and ν may well have their own perspectives on what counts as progress, and this needs to be formalized.

Broadly speaking, we expect that the distinction between strong and weak agendas will eventually disappear: in order to produce higher quality artefacts, more sophisticated systems involving behaviours perceived as creative will be required, and audiences will expect to project notions of creativity onto software to fully appreciate its output. In such a context, assessing processes and products simultaneously will be important, and we hope versions of this diagrammatic approach will enable this. In (Colton, Goodwin, and Veale 2012), we used the FACE model as a driving force for poetry generation software, rather than as a descriptive tool. We hope that system developers will similarly begin to think about their software in the above diagrammatic terms, in order to suggest interesting new avenues for implementation.

Acknowledgements

This work has been supported by EPSRC grants EP/J004049 and EP/L00206X, and through EC funding for the project COINVENT 611553 by FP7, the ICT theme, and the Future Emerging Technologies FET programme. We would like to thank the anonymous reviewers for their helpful comments.

References

Abrial, J.-R. 1996 *The B-Book – Assigning Programs to Meanings*. Cambridge University Press.

Abrial, J.-R. 2010. *Modeling in Event-B – System and Software Engineering*. Cambridge University Press.

al-Rifaie, M and Bishop, M. 2012. Weak vs. strong Computational Creativity, computing, philosophy and the question of bio-machine hybrids. In *Proceedings of the AISB Symposium on Computing and Philosophy*.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in Computational Creativity. In *Proceedings of the International Conference on Computational Creativity*.

Colton, S., and Wiggins, G. 2012. Computational Creativity: The final frontier? In *Proceedings of the European Conference on AI*.

Colton, S.; Cook, M.; Hepworth, R.; and Pease, A. 2014. On acid drops and teardrops: Observer issues in Computational Creativity. In *Proceedings of the AISB Symposium on AI and Philosophy*.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-FACE poetry generation. In *Proceedings of the International Conference on Computational Creativity*.

Colton, S.; Pease, A.; and Charnley, J. 2011. Computational Creativity Theory: The FACE and IDEA descriptive models. In *Proceedings of the Int. Conference on Computational Creativity*.

Colton, S. 2001. Experiments in meta-theory formation. In *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Science*.

Colton, S. 2002. *Automated Theory Formation in Pure Mathematics*. Springer.

Colton, S. 2008a. Automatic invention of fitness functions with application to scene generation. In *Proceedings of EvoMusArt*.

Colton, S. 2008b. Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI Spring Symposium on Creative Intelligent Systems*.

Colton, S. 2012. Evolving a library of artistic scene descriptors. In *Proceedings of EvoMusArt*.

Colton, S. 2012b. The Painting Fool: Stories from building an automated painter. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*. Springer.

Gabriel, R. P. and Goldman, R. 2000. Mob software: The erotic life of code. In *Proceedings of the ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications*.

Gabriel, R. P. and Goldman R. 2006. Conscientious software. In *ACM SIGPLAN Notices*, 41.

Gallie, W. 1956. Essentially contested concepts. *Proceedings of the Aristotelian Society* 56.

Girault, C. and Valk, R. 2003. *Petri nets for systems engineerings – a guide to modeling, verification, and applications*. Springer.

Jensen, K. and Kristensen, L. M. 2009. *Coloured Petri Nets - Modelling and Validation of Concurrent Systems*. Springer.

Jones, C. B. 1990. *Systematic software development using VDM*. Prentice Hall.

Jordanous, A. 2012. A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. Cognitive Computation 4(3).

Jordanous, A. 2012. *Evaluating Computational Creativity: A Standardised Procedure for Evaluating Creative Systems and its Application*. Ph.D. Dissertation, University of Sussex.

Lenat, D. 1976. *AM: An Artificial Intelligence approach to discovery in mathematics*. Ph.D. Dissertation, Stanford University.

Li, Y.; Hu, C.; Minku, L.; and Zuo, H. 2013. Learning aesthetic judgements in evolutionary art systems. *GPEM* 14(3).

Machado, P., and Cardoso, A. 2002. All the truth about NEvAr. *Applied Intelligence* 16(2).

Pearce, M. T. and Wiggins, G. A. 2001. Towards a Framework for the Evaluation of Machine Composition. In *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Science*.

Pease, A., and Colton, S. 2011. Computational Creativity Theory: Inspirations behind the FACE and IDEA models. In *Proceedings of the International Conference on Computational Creativity*.

Pease, A., and Colton, S. 2012. On impact and evaluation in Computational Creativity: A discussion of the Turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and Philosophy*.

Pereira, F.; Gervás, P.; and Cardoso, A. 2005. Experiments with assessment of creative systems: An application of Ritchie's criteria. In *Proceedings of the IJCAI Computational Creativity Workshop*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17.

Romero, J., and Machado, P., eds. 2007. *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer.

Wiggins, G. A. 2006. Searching for computational creativity. *New Generation Computing* 24(3).

Said, M. Y., Butler, M. J., and Snook, C. F. 2009. Language and tool support for class and state machine refinement in UML-B. In *Proceedings of the 2nd World Congress on Formal Methods*.

Spivey, M. 1992. *The Z notation: A reference manual*. Prentice Hall.

Can a Computationally Creative System Create Itself? Creative Artefacts and Creative Processes

Diarmuid P. O'Donoghue, James Power, Sian O'Briain, Feng Dong[*]*, Aidan Mooney, Donny Hurley, Yalemisew Abgaz, Charles Markham,*

Department of Computer Science, NUI Maynooth, Co. Kildare, Ireland.

[*] *Department of Computer Science and Technology, University of Bedfordshire, Luton, UK.*

Abstract

This paper begins by briefly looking at two of the dominant perspectives on computational creativity; focusing on the creative *artefacts* and the creative *processes* respectively. We briefly describe two projects; one focused on (artistic) creative artefacts the other on a (scientific) creative process, to highlight some similarities and differences in approach. We then look at a 2-dimensional model of Learning Objectives that uses independent axes of *knowledge* and (cognitive) *processes*. This educational framework is then used to cast artefact and process perspectives into a common framework, opening up new possibilities for discussing and comparing creativity between them. Finally, arising from our model of creative processes, we propose a new and broad 4-level hierarchy of computational creativity, which asserts that the highest level of computational creativity involves processes whose creativity is comparable to that of the originating process itself.

Introduction

Creativity is frequently seen through the "search space" metaphor (Boden, 1992; O'Donoghue and Crean, 2002; Wiggins, 2006; O'Donoghue *et al*, 2006; Ritchie, 2012; Veale, 2012; Pease *et al*, 2013). The space of possible *products* is represented as physical space, where each location represents a different product. Other search processes have been through this space previously, so a creative search process attempts to focus on regions of this space that have not yet been explored. The space of all search products carries different, often unpredictable values (including novelty). Boden (1992) identified three levels of creativity with *improbable* creativity exploring regions of this search space that are unlikely to have been visited previously. *Exploratory* creativity deliberately attempts to explore the boundaries of that search space. *Transformational* creativity attempts to identify and explore new search spaces, to identify products that did not exist in the original search space.

Viewing computational creativity through this search space metaphor, we can see that many artistic forms of creativity are adequately described. Artistic styles of creativity can be seen to explore the space of possible creative *artefacts* from one of the traditional creative domains like art, music, creative writing *etc.* (as used in Carson *et al*, 2005). Highly creative individuals transform accepted search spaces to create new possibilities – such as impressionism or cubism.

Creative artefacts and creative processes are generally discussed quite separately, with creative products/artefacts attracting the most attention. One criticism often levelled at the discipline of computational creativity, is that it is overly focused on creative products – paying too little attention to the process (Stojanov and Indurkhya, 2012; O'Donoghue and Keane, 2012). Analogy, metaphor are often seen as the dominant approaches to processes centred creativity, though evolutionary computing approaches are also popular. These creative processes appear to be generally associated with creativity within scientific or engineering types of disciplines. Thus, the starting point for this paper concerns the two distinct perspectives on computational creativity, focusing on *artistic products* and *scientific processes*. Later in this paper we shall use an educational assessment framework to cast both perspectives into a common framework, in order to bring resolution to these apparently conflicting perspectives.

It should be noted that even the basic distinction between artistic creativity and scientific creativity is not universally accepted. The noted 18th century mathematician (and poet) W. R. Hamilton regarded mathematics *"as an aesthetic creation, akin to poetry, with its own mysteries and moments of profound revelation"* (from Hankins, 1980). Mathematicians have also compared the aesthetic beauty of various equations, with Euler's identity ($e^{i\pi} + 1 = 0$) ranked the most beautiful equation in mathematics (Wells, 1990). Conversely, the process of analogical reasoning is generally seen as a driving force of scientific creativity (Brown, 2003), but at least one study has shown that analogical reasoning appears to play a part in some contemporary artistic creativity (Okada *et al*, 2009). Despite these overlaps, we shall proceed with the two basic categories of creative products and creative processes for the purposes of this paper.

Creative Products and Creative Processes

We briefly compare and contrast creative products (or artefacts) and creative processes using two projects that serve to highlight some commonalities and help identify some differences. The first is *ImageBlender* that creates new images using complex transformations of two given input images. The second *RegExEvolver* represents simple pro-

cesses (a finite automaton) as regular expressions, creating new regular expressions from that expression.

Another criticism often levelled at computationally creative systems is that *"Most of them are given, in advance, a detailed (hardcoded) description of the domain"* (Stojanov and Indurkhya, 2012). The two models presented in this paper make minimal assumptions about their relevant problem domains. ImageBlender is based on the assumption that the inspiring set contains images - regardless of what those images depict. RegExEvolver assumes only that the input is a valid regular expression – again with no additional limits. Additionally, both models take a very small inspiring set of two and just one items respectively.

Both systems use the search and evaluate strategy of evolutionary computation to explore the space of possible outputs. Both adopt a multi-objective selection strategy (Luke, 2013) to promote the emergence of high quality outputs. Multi-objective evaluation uses several independent objective functions to evaluate individuals in the population. Evolution then proceeds under the guidance of a Pareto-optimal selection strategy.

Finally, both projects use *interesting-ness* as one of the objective functions to guide evolution towards the creation of solutions. In both cases interestingness is estimated by the Kolmogorov complexity of the created output. This use of Kolmogorov complexity is slightly different to that discussed by McGregor (2007). Other metrics are used to ensure that the results have some measurable *novelty* compared to the given inspiring set – by measuring the dissimilarity between an evolved output and the given input(s).

These two metrics of interestingness and novelty are used as simple, general purpose estimates of the quality and novelty (Ritchie, 2001) that are sought by creative systems. We shall now see if these minimal assumptions can prove useful for computational creativity – in the absence of more detailed information on the problem domain.

Creative Artefacts from *ImageBlender* *ImageBlender* creates new images by combining two given input images. Well known techniques exist for combining two images using techniques like; super-positioning those images; selecting and combining sub-regions of the images using image manipulators like rotation, translation, scale, reflection *etc*. Many such techniques can be considered as collage generation that selectively combine parts of (two or more) given images.

However, ImageBlender does not operate directly upon the images but explores the space of possible images produced by combining transformed representations of those images. This process might be considered transformational in that it explores a space of possible images that has not been explicitly explored before (as far the authors can ascertain). ImageBlender currently focuses on the *Fast Fourier Transform (FFT)* of those images, creating a new image by combining portions of the *phase* and *frequency* information from those images. ImageBlender explores the space of possible images produced by various combinations of FFT's and then using the inverse transform (FFT^{-1}) to produce the resulting image. No restrictions are placed on the input images – other than those inherent to the FFT transform. Thus images may be black and white, greyscale, or colour; representing geometric figures, paintings, photographs *etc*. or any combination of these.

ImageBlender uses evolutionary computation to produce creative images, guided by a Pareto-optimal selection strategy. Among the metrics used are a number of estimates of the Kolmogorov complexity of the output image – ensuring there is some appropriate level of interestingness associated with the output images. Other metrics favour new images that are different from both input images.

Interestingly, some of these measures also have a role in assessing the beauty of images. Forsythe *et al* (2010) found that visual complexity can be adequately assessed using GIF compression and that the fractal dimension of an image often appears to be an adequate predictor of people's judgements of beauty.

Figure 1 shows two input images formed from black and white pixels only; a "checkerboard" of alternating black and white pixels (top left) and a black circle on a white background (top right of Figure 1). The grey appearance of the first image is caused by the low resolution reproduction of alternating black and white pixels. The final image was formed by combining the phase information from one image with the frequency information from the other, forming the third (bottom) image in Figure 1. Surprisingly, the output image has a far higher Kolmogorov complexity than either input image, suggesting a more interesting product. We argue that this output is creative in that it has the properties most frequently associated with creativity, it is: novel, interesting, unexpected and (arguably) has some aesthetic if geometric beauty. Appexdix 1 contains a few more sample images created by ImageBlender.

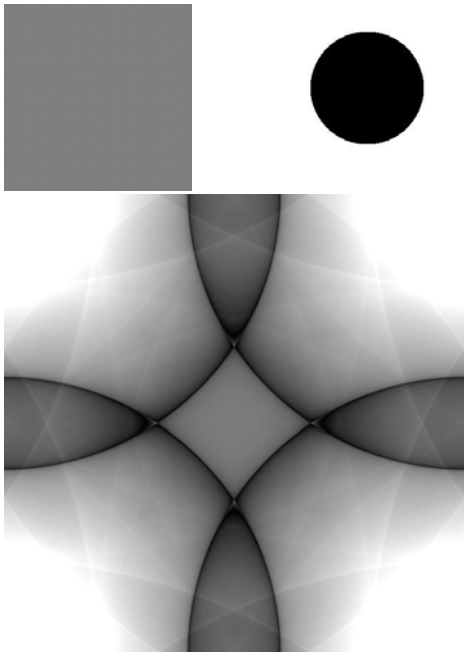


Figure 1: The two input images (above) and the new image (below) formed by blending the FFT of these images.

Creative Processes with RegExEvolver Computational creativity has addressed process centred creativity under three main categories: traditional GOFAI (Good Old Fashioned Artificial Intelligence) search processes, evolutionary search and analogy/metaphor/blending (Veale and O'Donoghue, 2000) approaches. However, instead of focusing on specific processes we look instead at general Turing Machine models of computational processes.

In this section we consider the case of creating outputs that are themselves processes. Creating a process rather than an "artefact" shouldn't in principle be that much of a change since computational processes are easily represented as strings of characters, parse trees or other structures. Such representations can allow "traditional" creativity e search to explore the space of possible artefacts/processes. In fact, evolutionary programming, genetic programming and grammatical evolution regularly output new programs in some executable programming language, though their focus in not normally on creative outputs. This situation where the creative output is itself a process also underpins the later section (below) that integrates creative processes and products through a theory of Educational Assessment.

A number of previous project have looked at creating outputs that are themselves processes. Procedural content generation (Togelius *et al*, 2011) is an emerging area devoted to the creation of game content for playable computer games. Cook *et al* (2013) discuss the Mechanic Miner system that generates the game mechanics for platform games using evolutionary computation. However Mechanic Miner and other procedural content generators are very focused on the domain of platform games and not on general purpose software development.

The Arís model (Pitu *et al*, 2013) creates formal specifications (in Spec#) for a given implementation (in C#) using analogical reasoning. Due to the creative and arguably unreliable nature of analogical reasoning, Arís uses a theorem prover to validate the inferences it automatically accepts. But unverified specifications may also spur the workaday *little-c creativity* (Gardner, 1993) of human specification writers. Finally, we note that Arís is also (potentially) capable of operating in the reverse direction, creating new source code (a process) for a given specification.

Many practitioners of computational creativity use the concept of *inspiring sets* to describe both the creative domain and (a sample of) the artefacts that have already been generated within that domain. In this section we briefly look at the creation of simple computational processes, as represented by *Regular Expressions (RegEx).* Each regular expression defines a language, and any regular expression can be converted to a *Finite State Machine (FSM)* that recognises strings from this language. The RegExEvolver project uses just one regular expression for its inspiring set and attempts to create new and potentially useful expressions from it.

As a simple example, a regular expression for the registration numbers of Irish vehicles before 2013 would be:

```
[0-9]{2}[A-Z]{1,2}[0-9]{1,5}
```

After this date, a new system was introduced conforming to the following regular expression:

```
[0-9]{2}[1-2]{1}[A-Z]{1,2}[0-9]{1,5}
```

As a second example we consider the rules for valid passwords used in a computer system. Valid passwords may be specified by a regular expression, with different "strengths" associated with different expressions. A weak expression might accept any combination of letters and numbers, but a stronger expression might require at least one of each of: a lower case letter, an upper case letter and a digit. RegExEvolver could also be used to create a new password specification given a pre-existing expression.

The process is similar to that used in fuzz-testing (Godfried *et al*, 2012), a software engineering technique used to find bugs in a program. One approach to ('black-box') fuzz testing involves analysing existing test inputs and then generating different, new inputs that may expose previously unknown vulnerabilities. A more sophisticated ('white-box') approach involves analysing the program's source code in order to generate test inputs that cause unexpected combinations of the program's flow of control. Common to both approaches is the goal of creating new combinations that had not been previously envisaged by the testers.

RegExEvolver uses evolutionary computation techniques to guide formation of the new RegEx under the guidance of a Pareto-optimal selection technique. The objective functions focus on the original and evolved expressions and also assess the languages that are generated by these expressions. To this end RegExEvolver uses the Xeger tool to generate random strings for any given RegEx. This is achieved by employing standard algorithms to convert the RegEx to an equivalent FSM and then choosing random transitions through this machine. Although repetition (denoted by the Kleene Star '*') in a RegEx can theoretically generate a string of infinite length, this is not an issue in practice as it would require the same transition to be chosen every time.

In addition to evaluating the generated strings (products) we also evaluate the processes themselves. The generated RegEx is compared to the original (input) RegEx by calculating the intersection of their corresponding FSM using the `dk.brics.automaton` package. In this way, evaluation of the new process (RegEx) itself ensures it overlaps the input expression, while also ensuring it contains some *novelty* compared to the input expression. However, in the absence of a problem domain, we do not evaluate the *usefulness* quality of the generated expressions.

RegExEvolver is focused on generating *novel* and potentially *useful* "processes" at level 3 of the Chomsky hierarchy. However, it is easy to see that other computationally creative processes could generate creative processes at any level from the Chomsky Hierarchy. It has been shown that the set of regular languages corresponding to regular expressions (or produced by a regular grammar) at level 3 are a subset of the set of context free languages at level 2, which in turn are a subset of the set of content sensitive languages at level 1, and that these in turn are a subset of

the set of recursively enumerable languages at level 0 (Chomsky, 1959).

Evaluating Creativity

Both ImageBlender and RegExEvolver create new outputs without the benefit of any specific context or the constraints and values that frequently arise from such contexts. Thus, evaluating their outputs can be considered all the more difficult. While this might be seen as a weakness, we see it as positive support for the generality of our approach. That is, some creativity is possible without making detailed assumptions about the target domain – without committing to some low level detail that will later limit the breadth or *flexibility* (Guilford, 1950) of our creative system.

Defeasible Creativity Newell, Shaw and Simon (1963) highlighted that one criterion for creativity is that a given answer should cause us to reject an answer that we had previously accepted. From this perspective computational creativity should place its highest value on creativity that contradicts some existing belief, leading to the "shock and amazement" often associated with H-Creativity.

Evaluation plays a central role in computational creativity. We identify two distinct types of evaluation: Subjective evaluation and Objective evaluation. Subjective evaluation is carried out by a computationally creative process to ensure the quality and novelty of the output. However that real value of a creative output can only ever be truly determined by an independent group of evaluations. A true determination of the qualities of novelty and/or quality can only ever be made by an independent adjudicator.

Objective evaluation relies heavily on consensus reality and thus on some target population of evaluators – either the general public or some target group of critics. To this end, a comprehensive model of computational creativity must incorporate a model of the beliefs of that target group. Thus a creative system must either implicitly or explicitly, incorporate a model of the beliefs of that target group of evaluators. Thus, a *Theory of Mind* (ToM) is a fundamental issue in computational creativity – be that either an explicit theory or one implicitly instantiated in the model and its use of data (such as the inspiring set). Any ToM will suffer inaccuracies and other problems, especially when it is used within the context of creative reasoning. Thus, we conclude that a defining characteristic of computational creativity is that the output can only be truly evaluated and assessed by an independent adjudicator.

In effect, the objective metrics used in the two projects described above implicitly incorporate a simple ToM in terms of the *interestingness* value estimated by multi-valued pareto-optimal values, including the Kolmogorov complexity of the created products.

Integrating Creative Products and Creative Processes

Creative *products* and creative *processes* appear to bring different perspectives to computational creativity. Often, it appears that these perspectives are almost irreconcilable in terms of their values and objectives. We now explore one means of resolving the apparent differences between the product and process perspectives of computational creativity. The integration we explore is at the cognitive level, but it also bares relevance to other levels of creativity; from the neurological to the sociological.

In this section we review some work on education, as this is another discipline that values the creativity of its outputs – promoting the creativity of students produced by educational systems. Bloom's (1956) taxonomy of Learning Objectives (top of Figure 2) tried to get away from simple rote learning and promote higher forms of learning such as evaluating and analysing. The taxonomy was primarily aimed at informing education and assessment activities. The taxonomy was aimed at supporting objective assessment of educational activities and thus focuses on measurable and quantifiable properties.

While rote learning was seen as the lowest form of education attainment, synthesis and evaluation were seen as the highest achievements in the original (1956) taxonomy. "Creation" was only included in this original taxonomy as part of the "Synthesis" category and surprisingly, Synthesis was seen as a lower level of attainment than "Evaluation".

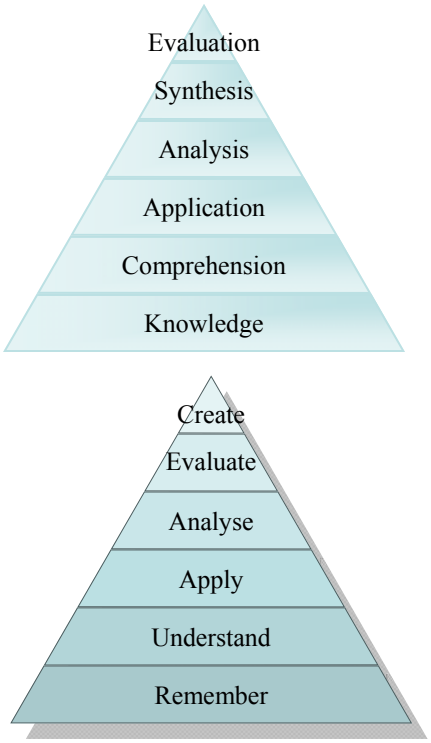


Figure 2: Bloom's Revised Taxonomy (below) places greater emphasis on the role of creativity in educational attainment

Bloom's Revised Taxonomy A subsequent revision of this taxonomy (Anderson and Krathwohl, 2001) (bottom of Figure 2) introduced a number of changes as moving from noun based to a verb based form and other changes. One of the most significant changes involved the introduction of "Create" as the highest level of educational attainment and a "demotion" of Evaluation below the Create level.

Learning Objectives Matrix

As noted by Krathwohl (2002) the unidimensional hierarchy of Bloom's Revised Taxonomy incorporated both noun/knowledge and verb/process and thus was essentially dual in nature. Anderson and Krathwohl (2001) overcame this problem by separating the noun (knowledge) dimension from the verb (process) dimension.

This resulted in a two dimensional matrix, with one axis called The Knowledge Dimension representing the noun related information. The other axis is called The Process Dimension and this represents verb related information. Towards the origin of this array are found some of the simplest forms of educational attainment, involving rote learning and the listing of facts. Furthest from the origin then, are the highest forms of educational attainment - notably including "create".

At this point we should acknowledge that Krathwohl's original diagram was a simple 2D matrix. However, in Figure 3 we depict a representation due to Rex Heer's model[1] that uses the third dimension to highlight the difference between the simpler and higher forms of educational attainment. Thus, the simpler forms of learning are depicted with the least height, while the highest forms of learning are depicted by greater heights. Heer's model and this paper make the assumption by using the third dimension that the Cognitive Process and Knowledge Dimensions are represented to the same scale. However, relative heights are merely suggestive of the levels of educational attainment.

Learning Objectives are typically stated in the form "*The learner will be able to do X with Y*" where X is a verb representing the relevant cognitive process and Y is a noun representing the corresponding knowledge. Of course, both the X and Y are sourced from the two axes of Figure 3. For example, "*The learner will be able to remember the law of supply and demand*" where X is "*remember*" and Y is "*the law of supply and demand*". The nouns and verbs on the two axes, along with the verbs contained in each vertex of the matrix, provide a terminology and reference points to describe and discuss different creative systems.

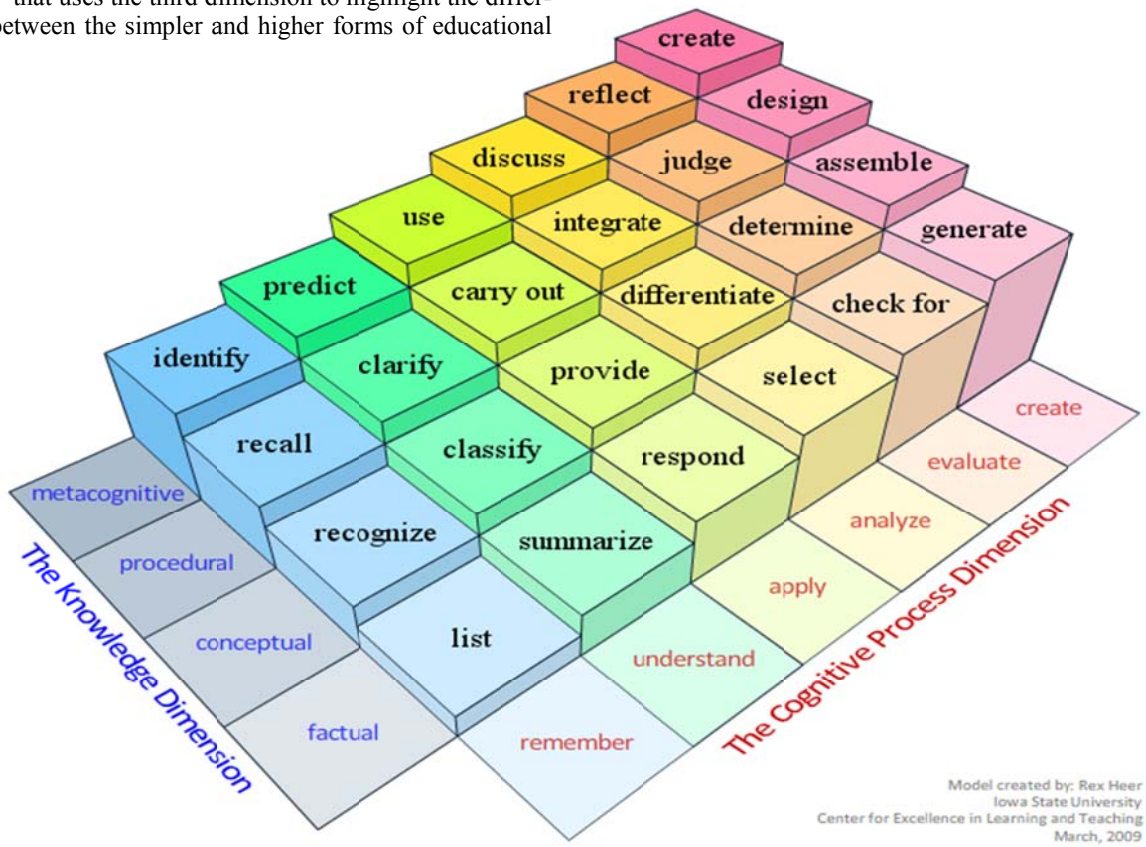


Figure 3: A 3D representation[1] of Krathwohl's 2D Matrix of Educational Assessment. This is used to view the artefact and process perspectives of computational creativity within a common framework.

[1] This image has been reproduced from: A Model of Learning Objectives–based on *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives* by Rex Heer, Center for Excellence in Learning and Teaching (CELT), Iowa State University.

We propose an adaptation of this taxonomy for the purposes of informing work on computational creativity. Adapting the typical statement of Learning Objectives to the domain of computational creativity, we suggest that we read this as *"A computationally creative system should be able to do X with Y",* where X and Y are identified from the diagram in Figure 3.

Of course, we acknowledge that adopting this matrix is contingent upon accepting some similarity between an *artefact* and the *knowledge* that it embodies. We feel that allowing this comparison may provide a new and useful perspective on computational creativity.

The Knowledge Dimension Firstly we look at the Knowledge Dimension of Figure 3. This we liken to the artefact perspective of computational creativity, as both are concerned with the production of new ideas in the form of knowledge or artefacts that represent that knowledge.

Factual: Knowledge of the basic elements of the discipline, essential facts, terminology and details. Factual knowledge details the basic elements required to function in some discipline – music, art, maths etc.

Conceptual: knowledge of classifications, categories and generalisations; knowledge of theories, models, and structures. Knowledge about how factual elements can be related and combined to form low level structures; this might include ontological and other knowledge (*warm* colours, *emotive* words).

Procedural: knowledge of genre-specific skills, algorithms and techniques, knowledge of criteria for determining when to use appropriate procedures, details how to do something; skills, algorithms, techniques and method, including their use.

Metacognitive: strategic knowledge, knowledge about the cognitive tasks including appropriate contextual and conditional knowledge, self-knowledge, and awareness of one's own cognition (or the systems own cognition).

The Cognitive Process Dimension depicted in Figure 3 highlights different levels of cognitive processes. While simple cognitive process are identified (like remember and understand), our concern is with the create level. Figure 3 depicts "create" as the highest level of cognitive process. However, it is interesting to note that *creative* and *evaluate* are seen as distinct regions on the cognitive dimension, given their joint roles in many creative systems.

We shall examine how the creative process interactions with (or relies upon) the previous four levels of knowledge: *factual, conceptual, procedural* and *metacognitive*.

Cognitive Processes and "Create"

Before we look at the "create" level of Cognitive Processes Dimension, we note that the adjacent level of process is "evaluate". This would appear to highlight the close relationship between creation and evaluation. For example, at the metacognitive level of evaluation we see the "reflect" verb – with reflection often being seen as a precursor to creativity. However, this paper is focused on the differing levels of the "create" cognitive process.

Generate: Create Factual Outputs

While we may not frequently think of producing new facts as a creative challenge, we can see creativity as sometimes being involved - even when there is a known technique to help generate these facts. Let us consider the domain of prime numbers, whole natural numbers divisible only by themselves and 1. Prime numbers play an important role in cryptography and other domains. A non-creative process may simply list the known prime numbers. However, looking at the creative dimension we can see that "generating" a new prime number might be considered a creative task. Let us restrict the set of numbers even further to the set of Mersenne primes – that is, a prime number that is also a Mersenne number of the form ($M_n = 2^n - 1$). While this equation looks like it can "generate" arbitrary prime numbers, in fact most Mersenne numbers are not prime. The "Great Internet Mersenne Prime Search" project is devoted to discovering ever larger Mersenne prime numbers. Among the reasons for considering this to be a creative task is the enormity of the space of numbers and Mersenne numbers and the enormity of verifying that a given candidate is actually prime.

Assemble: Create Conceptual Outputs Creating new concepts might be achieved by combining previously existing concepts, by appropriately assembling a new construct using the lower factual level of knowledge. This could involve finding or creating new similarities between existing knowledge. Here the creation process is already known or relatively straightforward, with the focus being on the concepts and their creation. That is the "assembly" process is already known and is used to create the new knowledge.

Many creative systems appear to produce artefacts that introduce new concepts and facts, using systems that do not change while that artefact is being created. Even powerful systems like analogical reasoning and evolutionary computation typically create new concepts in an "assembly" like manner.

Design: Create Procedural Outputs The next level of creativity aims to design new procedures that might operate on existing or new facts. This level of creativity introduces additional flexibility and creative power, in that the range of possible outputs and artefacts is greatly increased upon the lower level.

Analogical reasoning, evolutionary computation and other approaches might be seen as involving metacognitive creation were they to reflect upon their own processes – and use this reflection to guide further progress (while evolutionary strategies take their progress into account through strategies like adaptive mutation and others, reactions do not (usually) take the form of metacognitive or reflective modifications to the creative process).

Create: Create Meta-Cognitive Outputs These often involve self-knowledge and reflection on that knowledge. The authors are not aware of any computational models addressing this level of computational creativity. Metacognitive and reflective processes may well encompass a Theory of Mind (ToM) as mentioned earlier. However, metacognitive aspects are generally not made explicit in most creative systems.

Levels of Computational Creativity

In this section we build on this joint perspective of knowledge/artefacts and processes. We begin by re-visiting computational creativity, but bearing in mind that creativity is also valued among thinking, processing students.

Creating Outputs that themselves Create Artefacts One significant feature of the generated RegEx is that it has a dynamic productive quality. The created product is itself, capable of generating products. In this case the created regular expression is at the lowest level of the Chomsky hierarchy, however a similar approach can in principle be adopted to generate automata at any level from the Chomsky hierarchy.

Interestingly, from a creativity perspective it is relatively straightforward to generate an output process that is at a more complex level that the input expression. That is an FSA can be easily transformed into a pushdown automaton by introducing an additional rule from a higher level automaton or by introducing higher level rules that overlap with the pre-existing grammar.

While there has been some discussion on the Turing Test and its potential use and adaptation for computational creativity (Boden, 2010; Pease *et al*, 2012), there have been surprisingly few references to Turing Machines in the various discussions on computational creativity.

What limits can we see on the artefacts that are produced by a computationally creative process? Similarly, what limits can we see in the creative processes generated by a creative system? Let us consider a creative system that outputs new and interesting Turing Machines. Earlier in this paper we saw a creative system that created a very simple Turing Machine (a regular expression). Is it possible to generate a creative Turing machine whose output could be (or at least include) a creative Turing Machine?

Turing Machine TM1 can be considered creative only if it generates an output string that was not produced by other machines in its inspiring set. Or alternatively, it produced the same output but did so using a different grammar. That is, either the language or the grammar must be different in some novel and useful way.

We now look at four levels of computationally creative system that arise from our focus on creative processes.

1. Direct Computational Creativity (DCC): In direct computational creativity the outputs (artefacts or processes) display the *novelty* and *quality* attributes associated with creativity. This category includes the majority of work in computational creativity where the (direct) output of the computational process is seen as creative. The directly created output might be an image, a poem, a piece of music, a recipe, or it might be a computational process such as a regular expression or an evolved program.

In terms of the search space metaphor, *direct* computational creativity searches through the space of novel and useful outputs.

2. Direct Self-Sustaining Creativity (DSC): In direct self-sustaining creativity, the outputs are added to the inspiring set and serve to drive subsequent creative episodes. Supporting this type of creativity involves two distinct factors. Firstly, the process must be capable of generating multiple creative artefacts and secondly the quality of the creative outputs must be adequately judged before inclusion in the inspiring set.

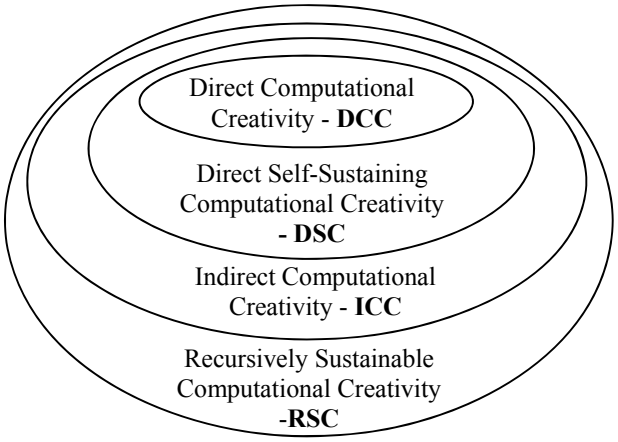


Figure 4: Levels and Limits of Computational Creativity

3. Indirect Computational Creativity (ICC): Indirect computational creativity outputs a creative process - and that creative process is *itself* creative. That is, ICC outputs processes and those creative processes can be considered as computationally creative systems. We see this as a form of *indirect* computational creativity, where we attribute creativity to the created process (as well as its creator).

We do not see these created processes as simple variants on some successful template – outputting a family of closely related creative models. But instead, the ICC should also itself display an ability to produce processes with the attributes of novelty and quality.

4. Recursively Sustainable Creativity (RSC): This is a further restriction on ICC, where RCC learns from its own outputs to maintain its own creativity. This would appear to be a very challenging level of computational creativity, creating highly creative processes. RCS represents the most significant challenge for computational creativity arising from this discussion. It would appear that techniques like evolutionary and genetic programming are best suited to producing such creative models.

Conclusion

The search space metaphor pervades most work on computational creativity but appears to have led towards a divide, between a focus on creative artefacts and less of a focus on the creative *processes*. Two projects are briefly described to highlight some differences between artefact centred and process centred computational creativity. ImageBlender creates new images by combining two input images in complex mathematical transformation of those images. RegExEvolver takes just one regular expression as its input and creates new expressions that differ from their expressions, either in terms of the language it produces or in terms of the expression itself.

Kolmogorov complexity and other general purpose compression algorithms appear to offer very useful and widely applicable mechanisms for assessing the *quality* of output artefacts. In particular they offer a means of assessing the interestingness of creative outputs. In recent work it has been shown that interestingness as estimated by the fractal dimension has been closely correlated with judgements of artistic quality (Forsythe *et al*, 2010).

To help clarify the apparent friction between artefact and process centred creativity we turned to educational assessment – as this is another discipline that values creativity among its outputs. We suggest that the 2-dimensional model of Learning Objectives by Anderson and Krathwohl (2002) can offer guidance in comparing creative artefacts and processes. Among its advantages are its 2D matrix, elucidating different levels of attainment achieved along the "Cognitive Process Dimension" and the "Knowledge Dimension". We argue that these two dimensions can be seen as loosely analogous to the "Creative Process" and the "Creative Artefact" perspectives that are common to computational creativity. Four increasing levels of creative process were identified, described using the verbs; *generate, assemble, design* and *create*. Each of these four levels impacts on increasing levels of the knowledge (or artefact) dimension.

Finally, our focus on computationally creative processes allowed us to identify a four-level hierarchy of computational processes. We suggest that the majority of work on computational creativity is at the level of "Direct Computational Creativity" and arguably some work approaches the level of "Direct Self-Sustaining Computational Creativity". However, we also define two higher levels, the first being "Indirect Computational Creativity" that outputs processes that themselves are creative. The final level we call "Recursively Sustainable Computational Creativity" and only this highest level is capable of outputting creative processes that are akin in their creative potential to the originating process.

Acknowledgements

Some of the research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement 611383. We would like to thank John McDonald, Tom Naughton, Ronan Reilly and Stephen Brown for their contributions to the *ImageBlender* project and we would like to thank Amy Wall for her assistance with *RegExEvolver*.

References

Anderson, L.W.; Krathwohl, D.R.; Airasian, P.W.; Cruikshank, K.A.; Mayer, R.E.; Pintrich, P.R.; Raths, J.; and Wittrock, M.C. (eds) (2000). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives,*

Anderson, L. W. and Krathwohl, D. R., *et al* (Eds.) (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon, Boston, MA.

Bloom, B. S.; Engelhart, M. D.; Furst, E. J.; Hill, W. H.; and Krathwohl, D. R. (1956). *Taxonomy of educational objectives: the classification of educational goals; Handbook I: Cognitive Domain,* New York, Longmans, Green.

Boden, M.A. (1992). *The Creative Mind,* Abacus.

Boden, M.A. (2010). The Turing Test and Artistic Creativity, *Kybernetes*, 39 (3), pp 409-413.

Carson, S.H.; Peterson, J.B.; and Higgins, D.M. (2005) Reliability, validity, and factor structure of the creative achievement questionnaire, *Creativity Research Journal*, 17, pp 37–50.

Chomsky, N., (1959). On certain formal properties of grammars, *Information and Control*, 2(2), pp 137-167.

Cook, M.; Colton S.; Raad, A.; and Gow J (2013). Mechanic Miner: Reflection-Driven Game Mechanic Discovery and Level Design, *LNCS Vol.* 7835, pp 284-293.

Forsythe, A.; Nadal, M.; Sheehy, N.; Cela-Conde, C.J.; and Sawey, M. (2010). Predicting beauty: Fractal dimension and visualcomplexity in art, *British Journal of Psychology,* 102(1), pp 49-70.

Gardner, H. (1993). Creating Minds, Basic Books, NY.

Godefroid, P.; Levin, M.Y.; and Molnar, D. (2012). SAGE: Whitebox Fuzzing for Security Testing, *Communications of the ACM*, 55(3), pp 40-44.

Guilford, J.P. (1950). Creativity, *American Psychologist*, 5(9), pp 444-454.

Hankins, T. (1980). *Sir William Rowan Hamilton,* Johns Hopkins University Press.

Krathwohl, D.R.(2002). A Revision of Bloom's Taxonomy: An Overview, *Theory Into Practice*, 41(4), pp 212-218.

Luke, S. (2013). *Essentials of Metaheuristics*, 2nd Edn. Lulu, http://cs.gmu.edu/~sean/book/metaheuristics/

McGregor, S. (2007). Algorithmic Information Theory and Novelty Generation, *Proc. 4th Intl. Joint Workshop on Computational Creativity (IJWCC),* London, June.

Okada, T.; Yokochi, S.; Ishibashi, K.; and Ueda, K. (2009). Analogical modification in the creation of contemporary art, *Cognitive Systems Research*, 10, pp 189–203.

O'Donoghue, D. and Crean, B. (2002) Searching for Serendipitous Analogies, *ECAI - Workshop on Creative Systems*, Lyon, France.

O'Donoghue, D.P.; Bohan, A.; and Keane, M.T. (2006). Seeing things: Inventive reasoning with geometric analogies and topographic maps, *New Generation Computing,* 24(3), pp 267-288.

O'Donoghue, D.P. and Keane, M.T. (2012). A Creative Analogy Machine: Results and Challenges, *International Conference on Computational Creativity (ICCC),* UCD, Dublin, Ireland, pp 17-24.

Pease, A.; Colton, S.; Ramezani, R.; Charnley, J.; and Reed, K. (2013). A Discussion on Serendipity in Creative Systems, *International Conference on Computational Creativity (ICCC),* Sydney, Australia, pp 64-71.

Pitu, M.; Grijincu, D.; Li, P.; Saleem, A.; Monahan, R.; and O'Donoghue, D.P. (2013). Arís: Analogical Reasoning for reuse of Implementation & Specification, *AI for Formal Methods (AI4FM) Workshop,* Rennes France, 22 July.

Ritchie, G. (2012). A Closer look at Creativity as Search, *4th International Conference on Computational Creativity (ICCC),* UCD, Dublin, Ireland, pp 41-48.

Newell, A.; Shaw, J.G.; and Simon, H.A. (1963). The Process of Creative Thinking, in *Contemporary Approaches to Creative Thinking,* pp 63-119. New York: Atherton.

Stojanov, G. and Indurkhya, B. (2012). Perceptual Similarity and Analogy in Creativity and Cognitive Development, *1st International Workshop on Similarity and Analogy-based Methods in AI (SAMAI),* at ECAI, France.

Togelius, J.; Yannakakis, G.N.; Stanley, K.O.; and Cameron Browne C. (2011). Search-based Procedural Content Generation: A Taxonomy and Survey, *IEEE Transactions on Computational Intelligence and AI in Games* 3(3), pp 1-15.

Veale, T.; and O'Donoghue, D. (2000). Computation and Blending, *Cognitive Linguistics,* 11(3/4), pp 253-282.

Veale, T. (2012). *Exploding the Creativity Myth*, London: Bloomsbury Academic.

Wells, D. (1990). Are these the Most Beautiful? *The Mathematical Intelligencer,* 12(3), pp 37-41.

Wiggins, G.A. (2006). Searching for Computational Creativity, *New Generation Computing*, 24(3), pp 209–222.

Appendix

This appendix contains a small sample of the images created by ImageBlender.

Automatic Detection of Irony and Humour in Twitter

Francesco Barbieri
Pompeu Fabra University
Barcelona, Spain
francesco.barbieri@upf.edu

Horacio Saggion
Pompeu Fabra University
Barcelona, Spain
horacio.saggion@upf.edu

Abstract

Irony and humour are just two of many forms of figurative language. Approaches to identify in vast volumes of data such as the internet humorous or ironic statements is important not only from a theoretical view point but also for their potential applicability in social networks or human-computer interactive systems. In this study we investigate the automatic detection of irony and humour in social networks such as Twitter casting it as a classification problem. We propose a rich set of features for text interpretation and representation to train classification procedures. In cross-domain classification experiments our model achieves and improves state-of-the-art performance.

Introduction

Irony and humour are just two examples of figurative language (Reyes, Rosso, and Veale 2013). Approaches to identify in vast volumes of data such as the internet humorous or ironic statements are important not only from a theoretical view point but also for their potential applicability in social network analysis and human-computer interactive systems. Systems able to select humorous/ironic statements on a given topic to present to a user are important in human-machine communication. It is also important for a system being able to recognise when users are being ironic/humorous to appropriate deal with their requests. Irony has also relevance in the field of sentiment analysis and opinion mining (Pang and Lee 2008) since it can be used to express a negative statement in an apparently positive way. However, irony detection appears as a difficult problem since ironic statements are used to express the contrary of what is being said (Quintilien and Butler 1953), therefore being a tough nut to crack by current systems. Reyes et al. (2013) approach the problem as one of classification training machine learning algorithms to sepatate ironic from non-ironic statements. Humour has been studied for a number of years in computational linguistics in terms of both humour generation (Stock and Strapparava 2006; Ritchie and Masthoff 2011) and interpretation (Mihalcea and Pulman 2007; Taylor and Mazlack 2005). In particular it has also been approached as classification by Mihalcea and Strapparava (2005) creating a specially designed corpus of one-liners (i.e., one sentence jokes) as the positive class and headlines and other short statements as a negative class.

Following these lines of research, we first try to detect these topics separately; then, since they are both figurative language, and they may have some correlation, we also try to detect them at the same time (we use the union of them as positive example). This last experiment is interesting as it will give us hints for figurative language detection, hence it will help us exploring new aspects of creativity in language (Veale and Hao 2010b). This experiment can be seen as a small step toward the design of a machine capable to evaluate creativity, and with further work also capable to generate creative utterances.

Our dataset is composed of text retrieved from the microblogging service Twitter[1].

For the experiments to be presented in this paper we use a dataset created for the study of irony detection which allows us to compare our findings with recent state-of-the-art approaches (Reyes, Rosso, and Veale 2013). The dataset also contains humorous tweets therefore being appropriate for our purpose.

The contributions of this paper are as follows:

- the evaluation of our irony detection model (Barbieri and Saggion 2014) to humour classification;
- a comparison of our model with the state-of-the-art; and
- a novel set of experiments to demonstrate cross-domain adaptation.

The paper will show that our model achieves and improve state-of-the-art performance, and that it can be applied to different domains.

Related Work

Verbal irony has been defined in several ways over the years but there is no consensual agreement on its definition. The standard definition is considered "saying the opposite of what you mean" (Quintilien and Butler 1953) where the opposition of literal and intended meanings is very clear. Grice (1975) believes that irony is a rhetorical figure that violates the maxim of quality: "Do not say what you believe to be false". Irony is also defined (Giora 1995) as any form of negation with no negation markers (as most

[1] https://twitter.com/

of the ironic utterances are affirmative, and ironic speakers use indirect negation). Wilson and Sperber (2002) defined it as echoic utterance that shows a negative aspect of someone's else opinion. Finally irony has been defined as form of pretence by Utsumi (2000) and Veale and Hao (2010b). Veale states that "ironic speakers usually craft their utterances in spite of what has just happened, not because of it. The pretence alludes to, or echoes, an expectation that has been violated". Past computational approaches to irony detection are scarce. Carvalho et. al (2009) created an automatic system for detecting irony relying on emoticons and special punctuation. They focused on detection of ironic style in newspaper articles. Veale and Hao (2010a) proposed an algorithm for separating ironic from non-ironic similes, detecting common terms used in this ironic comparison. Reyes et. al (2013) have recently proposed a model to detect irony in Twitter, which is based on four groups of features: signatures, unexpectedness, style, and emotional scenarios. Their classification results support the idea that textual features can capture patterns used by people to convey irony. Among the proposed features, *skip-grams* (part of the style group) which captures word sequences that contain (or skip over) arbitrary gaps, seems to be the best one. Computational approaches to humour generation include among others the JAPE system (Ritchie 2003) and the STANDUP riddle generator program (Ritchie and Masthoff 2011) which are largely based on the use of a dictionary for humorous effect. It has been argued that humorous discourse depend on the fact that they can have multiple interpretations, that is they are ambiguous. These characteristics are explored in approaches to humour detection. Mihalcea and Strappavara (2005) study classification of a restricted type of humorous discourse: *one-liners*, which have the purpose of producing humorous effect in very few words. They created a dataset semi-automatically by retrieving itemized sentences from web sites whose URLs contain words such as "oneliner", "humour", "joke", etc. Non-humorous data was created using Reuters titles, Proverbs, and sentences extracted from the British National Corpus. They use two types of models to separate humorous from non-humorous texts. On the one hand a specially designed set of features is created to model *Alliteration*, *Antonymy*, and *Slang* of a sexual oriented nature. On the other hand they tried a word-based text classification algorithm. Non surprisingly the word-based classifier is much more effective than the specially designed features. In (Mihalcea and Pulman 2007) additional features to model violated expectations, human oriented activities, and polarity are introduced. Veale (2013) also created a dataset of humorous similes by querying the web with specific similes patterns.

Data and Text Processing

The dataset used for the experiments reported in this paper has been prepared by Reyes et al. (2013). It is a corpus of 40.000 tweets equally divided into four different topics: *Irony*, *Education*, *Humour*, and *Politics*. The tweets were automatically selected by looking at Twitter hashtags (#irony, #education, #humour, and #politics) added by users in order to link their contribution to a particular subject and community. The hashtags are removed from the tweets for the experiments. According to Reyes et. al (2013), these hashtags were selected for three main reasons: (i) to avoid manual selection of tweets, (ii) to allow irony analysis beyond literary uses, and because (iii) irony hashtag may reflect a tacit belief about what constitutes irony and humour.

Another corpora is employed in our approach to measure the frequency of word usage. We adopted the Second Release of the American National Corpus Frequency Data[2] (Ide and Suderman 2004), which provides the number of occurrences of a word in the written and spoken ANC. From now on, we will mean with "frequency of a term" the absolute frequency the term has in the ANC.

In order to process the tweets we used the Gate plugin *Twitie* (Bontcheva et al. 2013), an open-source information extraction pipeline for Microblog Text. We used it as tokeniser and part-of-speech tagger. We also adopted Rita WordNet API (Howe 2009) and Java API for WordNet Searching (Spell 2009) to perform operations on WordNet synsets (Miller 1995).

Methodology

We approach the detection of irony and humour as a classification problem applying supervised machine learning methods to the Twitter corpus previously introduced. When choosing the classifiers we had avoided those requiring features to be independent (e.g. Naive Bayes) as some of our features are not. Since we approach the problem as a binary decision (deciding if a tweet is ironic or not) we picked two tree-based classifiers: Random Forest and Decision tree (the latter allows us to compare our findings directly to Reyes et. al (2013)). We use the implementations available in the Weka toolkit (Witten and Frank 2005).

To represent each tweet we use seven groups of features. Some of them are designed to detect imbalance and unexpectedness, others to detect common patterns in the structure of the tweets (like type of punctuation, length, emoticons). Below is an overview of the group of features in our model:

- Frequency *(gap between rare and common words)*
- Written-Spoken *(written-spoken style uses)*
- Intensity *(intensity of adverbs and adjectives)*
- Structure *(length, punctuation, emoticons, links)*
- Sentiments *(gap between positive and negative terms)*
- Synonyms *(common vs. rare synonyms use)*
- Ambiguity *(measure of possible ambiguities)*

In the following sections we describe the theoretical motivations behind the features and how them have been implemented.

[2]The American National Corpus (http://www.anc.org/) is, as we read in the web site, a massive electronic collection of American English words (15 million)

Frequency

Unexpectedness and Incongruity can be a signals of irony and humour (Lucariello 2007; Venour 2013). In order to study these aspects we explore the frequency imbalance between words, i.e. register inconsistencies between terms of the same tweet. The intuition is that the use of many words commonly used in English (i.e. high frequency in ANC) and only a few terms rarely used in English (i.e. low frequency in ANC) in the same sentence creates imbalance that may cause unexpectedness, since within a single tweet only one kind of register is expected. We are able to explore this aspect using the ANC Frequency Data corpus.

Three features belong to this group: **frequency mean**, **rarest word**, **frequency gap**. The first one is the arithmetic average of all the frequencies of the words in a tweet, and it is used to detect the *frequency style* of a tweet. The second one, **rarest word**, is the frequency value of the rarest word, designed to capture the word that may create imbalance.

Written-Spoken

Twitter is composed of written text, but an informal spoken English style is often used. We designed this set of features to explore unexpectedness and incongruity created by using spoken style words in a mainly written style tweet or vice versa (formal words usually adopted in written text employed in a spoken style context). We can analyse this aspect with ANC written and spoken, as we can see using this corpora whether a word is more often used in written or spoken English. There are three features in this group: **written mean**, **spoken mean**, **written spoken gap**. The first and second ones are the means of the frequency values, respectively, in written and spoken ANC corpora of all the words in the tweet. The third one, **written spoken gap**, is the absolute value of the difference between the first two, designed to see if ironic writers use both styles (creating imbalance) or only one of them. A low difference between written and spoken styles means that both styles are used.

Structure

With this group of features we want to study the structure of the tweet: if it is long or short (length), if it contains long or short words (mean of word length), and also what kind of punctuation is used (exclamation marks, emoticons, etc.). This is a powerful feature, as ironic and humorous tweets in our corpora present specific structures: for example ironic tweets are longer (mean length of an ironic tweet is 94.7 characters against 82.0467, 86.5776, 86.5307 of the other topics), and humorous tweets use more emoticons than the other domains (mean number of emoticons in a humorous tweet is 0.012 and in the other corpora is only 0.003, 0.001, 0.002). The Structure group includes several features that we describe below.

The **length** feature consists of the number of characters that compose the tweet, **n. words** is the number of words, and **words length mean** is the mean of the words length. Moreover, we use the number of verbs, nouns, adjectives and adverbs as features, naming them **n. verbs**, **n. nouns**, **n. adjectives** and **n. adverbs**. With these last four features we also computed the ratio of each part of speech to the number of words in the tweet; we called them **verb ratio**, **noun ratio**, **adjective ratio**, and **adverb ratio**. All these features have the purpose of capturing the style of the writer.

Inspired by Davidov et al. (2010) and Carvalho (2009) we designed features related to punctuation. These features are: number of **commas**, **full stops**, **ellipsis**, **exclamation** and **quotation** marks that a tweet contain.

We also added the feature **laughs** which is the number of *hahah*, *lol*, *rofl*, and *lmao*.

Additionally, there are the *emoticon* feature, that is the number of *:)*, *:D*, of *:(*, and *;)* in a tweet. This feature works well in the Humour corpus as it contains four times more emoticons than the other corpora. The ironic corpus is the one with the least emoticons (there are only 360 emoticons in the Irony corpus, while in Humour, Education, and Politics tweets they are 2065, 492, 397 respectively). In the light of these statistics we can argue that ironic authors avoid emoticons and leave words to be the central thing: the audience has to understand the irony without explicit signs, like emoticons. Humour seems, on the other hand, more explicit.

Finally we added a simple but powerful feature, *web-links*. It simply say if a tweet include or not an internet link. This feature result good for Humour and excellent for Irony, where internet links are not used frequently.

Intensity

We also study the intensity of adjectives and adverbs. We adopted the intensity scores of Potts (2011) who uses naturally occurring metadata (star ratings on service and product reviews) to construct adjectives and adverbs scales. An example of adjective scale (and relative scores in brackets) could be the following: horrible (-1.9) $\rightarrow$ bad (-1.1) $\rightarrow$ good (0.2) $\rightarrow$ nice (0.3) $\rightarrow$ great (0.8).

With these scores we evaluate four features for adjective intensity and four for adverb intensity (implemented in the same way): **adj (adv) tot**, **adj (adv) mean**, **adj (adv) max**, and **adj (adv) gap**. The sum of the AdjScale scores of all the adjectives in the tweet is called **adj tot**. **adj mean** is **adj tot** divided by the number of adjectives in the tweet. The maximum AdjScale score within a single tweet is **adj max**. Finally, **adj gap** is the difference between **adj max** and **adj mean**, designed to see "how much" the most intense adjective is out of context.

Synonyms

As previously said, irony convey two messages to the audience at the same time (Veale 2004). It follows that the choice of a term (rather than one of its synonyms) is very important in order to send the second, not obvious, message. The choice of the synonym is an important feature for humour as well, and it seems that authors of humours tweets prefer using common terms.

For each word of a tweet we get its synonyms with WordNet (Miller 1995), then we calculate their ANC frequencies and sort them into a decreasing ranked list (the actual word is part of this ranking as well). We use these rankings to define the four features which belong to this group. The first

one is **syno lower** which is the number of synonyms of the word w_i with frequency lower than the frequency of w_i. It is defined as in Equation 1:

$$sl_{w_i} = |syn_{i,k} \ : \ f(syn_{i,k}) < f(w_i)| \quad (1)$$

where $syn_{i,k}$ is the synonym of w_i with rank k, and $f(x)$ the ANC frequency of x. Then we also defined **syno lower mean** as mean of sl_{w_i} (i.e. the arithmetic average of sl_{w_i} over all the words of a tweet).

We also designed two more features: **syno lower gap** and **syno greater gap**, but to define them we need two more parameters. The first one is *word lowest syno* that is the maximum sl_{w_i} in a tweet. It is formally defined as:

$$wls_t = \max_{w_i}\{|syn_{i,k} \ : \ f(syn_{i,k}) < f(w_i)|\} \quad (2)$$

The second one is *word greatest syno* defined as:

$$wgs_t = \max_{w_i}\{|syn_{i,k} \ : \ f(syn_{i,k}) > f(w_i)|\} \quad (3)$$

We are now able to describe **syno lower gap** which detects the imbalance that creates a common synonym in a context of rare synonyms. It is the difference between *word lowest syno* and **syno lower mean**. Finally, we detect the gap of very rare synonyms in a context of common ones with **syno greater gap**. It is the difference between *word greatest syno* and *syno greater mean*, where *syno greater mean* is the following:

$$sgm_t = \frac{|syn_{i,k} \ : \ f(syn_{i,k}) > f(w_i)|}{n.\ words\ of\ t} \quad (4)$$

Ambiguity

Another interesting aspect of irony and humour is ambiguity. We noticed that ironic tweets includes the greatest arithmetic average of the number of WordNet synsets, and humour the least; this indicates that ironic tweets presents words with more meanings, an humorous tweets words with less meaning. In the case of irony, our assumption is that if a word has many meanings the possibility of "saying something else" with this word is higher than in a term that has only a few meanings, then higher possibility of sending more then one message (literal and intended) at the same time.

There are three features that aim to capture these aspects: **synset mean**, **max synset**, and **synset gap**. The first one is the mean of the number of synsets of each word of the tweet, to see if words with many meanings are often used in the tweet. The second one is the greatest number of synsets that a single word has; we consider this word the one with the highest possibility of being used ironically (as multiple meanings are available to say different things). In addition, we calculate **synset gap** as the difference between the number of synsets of this word (**max synset**) and the average number of synsets (**synset mean**), assuming that if this gap is high the author may have used that inconsistent word intentionally.

Sentiments

We analyse also the sentiments of irony and humour by using the SentiWordNet sentiment lexicon (Esuli and Sebastiani 2006) that assigns to each synset of WordNet sentiment scores of positivity and negativity.

There are six features in the Sentiments group. The first one is named **positive sum** and it is the sum of all the positive scores in a tweet, the second one is **negative sum**, defined as sum of all the negative scores. The arithmetic average of the previous ones is another feature, named **positive negative mean**, designed to reveal the sentiment that better describe the whole tweet. Moreover, there is **positive-negative gap** that is the difference between the first two features, as we wanted also to detect the positive/negative imbalance within the same tweet.

The imbalance may be created using only one single very positive (or negative) word in the tweet, and the previous features will not be able to detect it, thus we needed to add two more. For this purpose the model includes **positive single gap** defined as the difference between most positive word and the mean of all the sentiment scores of all the words of the tweet and **negative single gap** defined in the same way, but with the most negative one.

Experiments and Results

The experiments described in this section aim at verifying: (i) the discriminative power of our model, (i) the portability of the model across domains, and (iii) its state-of-the-art status. In order to carry out experimentation and to be able to compare our approach to that of (Reyes, Rosso, and Veale 2013) we use several datasets derived from the corpus used in the paper.

Irony Detection

Our first experiment addresses the problem of irony detection comparing the performance of our model with that of Reyes et al. (Reyes, Rosso, and Veale 2013). In order to replicate their experimental setting, three balanced datasets were created from the corpus: (i) *Irony vs Humour*, (ii) *Irony vs Education*, and (iii) *Irony vs Politics*. Each dataset is composed of 10,000 examples of irony and 10,000 examples of a different topic. A 10-fold cross-validation experiment was run in each dataset and precision, recall, and f-measure computed. The results of the experiments are presented in Table 1.

Cross-domain Irony and Humour Detection

Our second experiment addresses cross-domain adaptation, which has not been addressed in previous work. We designed three balanced *training* sets composed of 7500 positive tweets (irony or humour) and 7500 of each negative topic that remain available (Education/Humour/Politics when the positive is Irony and Education/Irony/Politics when the positive is Humour) and three balanced *test* sets composed of 2500 positive and 2500 of each negative topic (Education/Humour/Politics when the positive is Irony and Education/Irony/Politics when the positive is Humour). We carried out all the Train/Test possible combinations to verify

	Education			Humour			Politics		
Model	P	R	F1	P	R	F1	P	R	F1
Reyes et. al	.76	.66	.70	.78	.74	.76	.75	.71	.73
Our model	.87	.87	**.87**	.88	.88	**.88**	.87	.87	**.87**

Table 1: Precision, Recall, and F-Measure over the three corpora Education, Humour, and Politics. Both our and Reyes et al. results are shown; the classifier used is Decision Tree for both models.

	Training Set								
	Education			**Humour**			**Politics**		
Test set	P	R	F1	P	R	F1	P	R	F1
Education	.87/.89	.87/.89	.87/.89	.86/.86	.86/.85	.86/.85	.86/.87	.86/.87	.86/.87
Humour	.78/.79	.77/.74	.77/.74	.88/.89	.88/.89	.88/.89	.78/.79	.77/.74	.76/.74
Politics	.82/.83	.82/.83	.82/.82	.83/.83	.82/.82	.82/.82	.88/.89	.88/.89	.88/.89

Table 2: Results of Experiment 2 when positive topic is Irony and negative topics are Education, Humour and Politics. The table includes Precision, Recall and F-Measure for each Training/Testing topic combination written in the form "Decision Tree / Random Forest" as we used these two algorithms as classifiers.

how the model works when the domain is changed (one such instance is to train in the Irony/Politics dataset and evaluate it in the Irony/Education dataset). The results of the experiments are presented in Tables 2 and 3.

Figurative Language Filtering

Our third experiment consists on treating irony and humour as a single class representing figurative language; here we want to verify whether our model can separate "figurative" from "non-figurative" language. We designed one balanced Training set composed of 15000 positive tweets (7500 of Irony and 7500 of Humour) and 15000 negative examples (7500 of Education and 7500 of Politics). Then a balanced Test set composed of 5000 positive tweets (2500 of Irony and 2500 of Humour) and 5000 negative examples (2500 of Education and 2500 of Politics). Table 4 presents results of this experiment comparing two classification algorithms: Decision Tree and Random Forest.

Feature Analysis

Finally and in order to have a clear understanding about the contribution of each features of our model, we also studied the behaviour of information gain in each dataset. We compute information gain experiments over the three Training sets of our "cross-domain" experiments. Information gain results are directly correlated to the classification results as we are using tree based classifiers and features with high information gain will be at the top of the tree i.e. important discriminators. Figure 1 shows the information gain when the positive topic is Irony, Figure 2 when the positive topic is Humour. In Table 5 (a) and (b) are shown the Pearson Correlation between information gain of each feature over different topics when training Irony and Humour. The correlation has been calculated to determine whether the system uses similar features for different negative topics (if the correlation is low we are likely to have cross-domain problems). The correlation can tell us how well correlated two topics are.

Discussion

Looking at the figures obtained in our irony detection experiments, it appears that our model is more balanced in terms of precision and recall and that our overall f-measure improves over previous work having the additional advantage of the features being easy to compute.

Now turning to the cross-domain experiments we observe that our model performs reasonably well across-domains. That is to say except when we try to identify humorous tweets having trained with irony. This is in fact an interesting result which may indicate that not all features of our model are appropriate for humorous discourse, requiring the design of additional features for this type of figurative language.

With respect to the figurative language filtering experiments, results seem promising. Our experiments can not be compared with previous approaches directly because of differences in datasets but we point out that in humour classification (Mihalcea and Strapparava 2005) using specially designed "humour" characteristics accuracy results are around 76%.

Finally, our feature analysis experiments (Figures 1 and 2), we observe that features for structure, frequency, and synonymy are discriminators of irony. Although there is great variability across domains which is also shown in the correlation Table 5. Where humour is concerned, we see that features of structure, synonymy, frequency and intensity also are good discriminators again with great variability across domains. Features belonging to ambiguity and sentiment have little discriminative power. Regarding figurative versus not figurative experiment the best features are **syno lower**, **rarest val**, **word length** and **adj/adv max**. In comparison to education and politics, humour and irony include longer (**word length**) and more common words (**syno lower**, **rarest val**). Moreover, intensity of adjectives and adverbs (**adj/adv max**) is important characteristic as humour and irony include more intense terms.

	Training Set								
	Education			**Irony**			**Politics**		
Test set	P	R	F1	P	R	F1	P	R	F1
Education	.78/.81	.78/.81	.78/.81	.55/.57	.53/.53	.46/.43	.72/.77	.71/.75	.71/.75
Irony	.72/.64	.71/.61	.71/.58	.88/.89	.88/.88	.88/.88	.60/.67	.69/.63	.69/.61
Politics	.73/.77	.73/.76	.73/.76	.60/.61	.56/.55	.51/.48	.80/.84	.80/.84	.80/.84

Table 3: Results of Experiment 2 when positive topic is Humour and negative topics are Education, Irony and Politics. The table includes Precision, Recall and F-Measure for each Training/Testing topic combination written in the form "Decision Tree / Random Forest" as we used these two algorithms for the classifications.

P	R	F1
.80/.83	.80/.83	.80/.83

Table 4: Figurative language filtering results. Precision, Recall, and F-measure numbers correspond to two algorithms: Decision Tree/Random Forest.

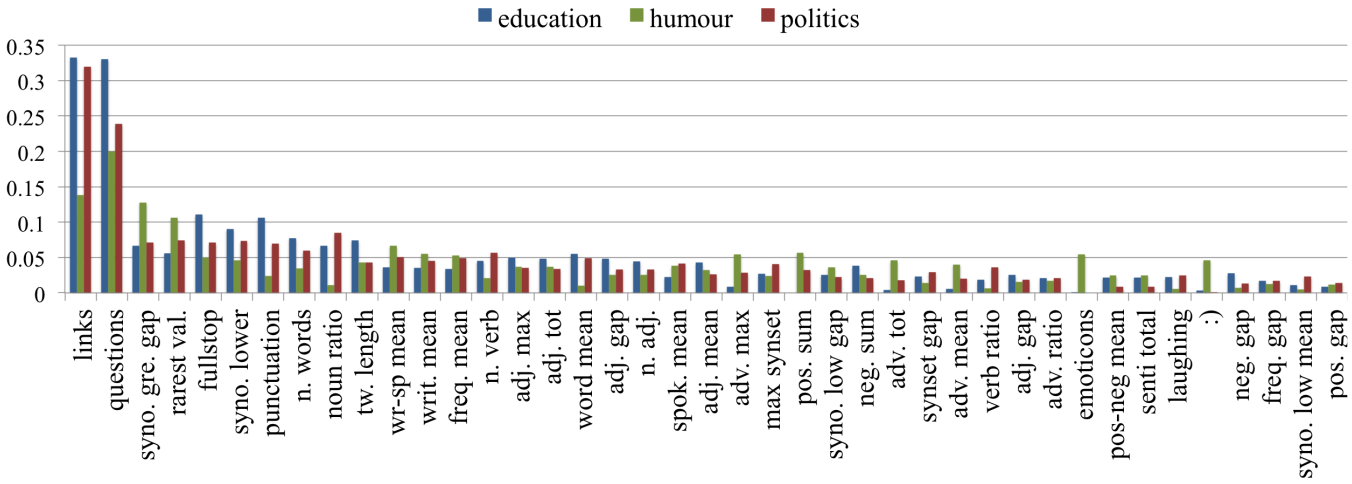


Figure 1: Information gain of each feature of the model. Irony corpus is compared to Education, Humour, and Politics corpora. High values of information gain help to better discriminate ironic from non-ironic tweets.

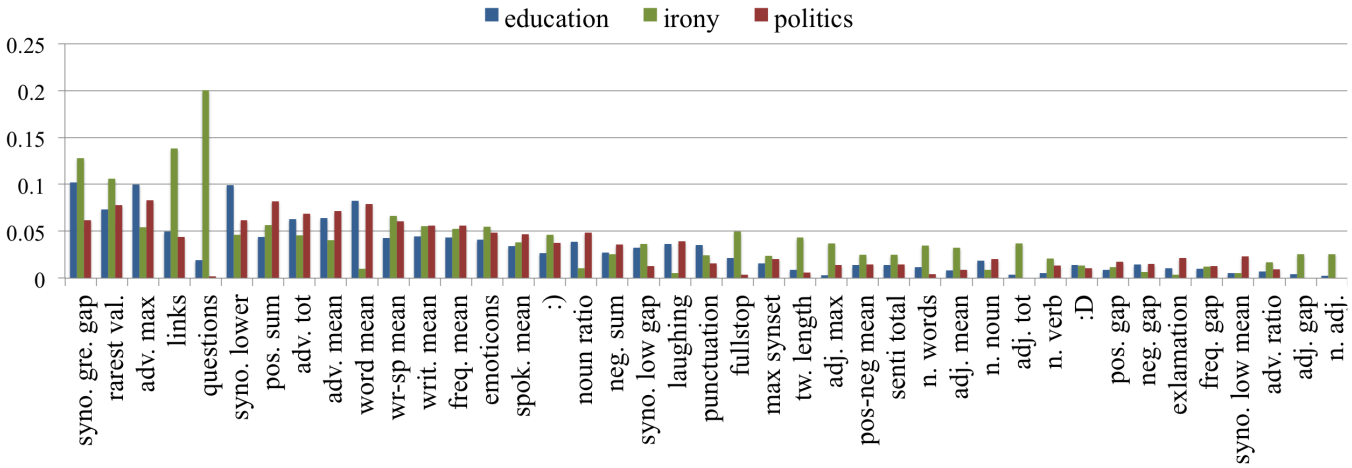


Figure 2: Information gain of each feature of the model. Humour corpus is compared to Education, Irony, and Politics corpora. High values of information gain help to better discriminate humorous from non-humorous tweets.

(a)	Education	Humour	Politics
Education	1	0.76	0.96
Humour	-	1	0.76
Politics	-	-	1

(b)	Education	Irony	Politics
Education	1	0.48	0.89
Irony	-	1	0.36
Politics	-	-	1

Table 5: Pearson Correlation between information gain of each feature over different topics when training on Irony (a) or Humour (b)

Conclusion and Future Work

In this article we have proposed a novel linguistically motivated set of features to detect irony and humour in the social network Twitter. The features take into account frequency, written/spoken differences, sentiments, ambiguity, intensity, synonymy and structure. We have designed many of them to be able to model "unexpectedness" and "incongruity", a key characteristic of both genres.

We have performed controlled experiments with an available corpus used in previous work which allow us to carried out experimentation in different scenarios. First, we carried out experiments to verify the performance of our set of features compared with previous work obtaining promising results. Second, we have carried out cross-domain experiments to show that the model can be used across domains. This experiment also shows that additional features are needed because irony and humour have their own particular characteristics. Third, we have performed an experiment to try to classify figurative language obtaining initial reasonable results. There is however much space for improvements. The ambiguity aspect is still weak in this research, and it needs to be improved. Also experiments adopting different topics may be useful in order to explore the system behaviour in a more realistic situation. We plan to model additional features to better distinguish between the two forms of figurative language.

Acknowledgments

We are grateful to three anonymous reviewers for their comments and suggestions that help improve our paper. The research described in this paper is partially funded by fellowship RYC-2009-04291 from Programa Ramón y Cajal 2009 and project number TIN2012-38584-C06-03 (SKATER-UPF-TALN) from Ministerio de Economía y Competitividad, Secretaría de Estado de Investigación, Desarrollo e Innovación, Spain. We also acknowledge partial support from the EU project Dr. Inventor (FP7-ICT-2013.8.1 project number 611383).

References

Barbieri, F., and Saggion, H. 2014. Modelling Irony in Twitter. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, 56–64. Gothenburg, Sweden: Association for Computational Linguistics.

Bontcheva, K.; Derczynski, L.; Funk, A.; Greenwood, M. A.; Maynard, D.; and Aswani, N. 2013. Twitie: An open-source information extraction pipeline for microblog text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. Association for Computational Linguistics.

Carvalho, P.; Sarmento, L.; Silva, M. J.; and de Oliveira, E. 2009. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, 53–56. ACM.

Davidov, D.; Tsur, O.; and Rappoport, A. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, 107–116. Association for Computational Linguistics.

Esuli, A., and Sebastiani, F. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of Language Resources and Evaluation Conference*, volume 6, 417–422.

Giora, R. 1995. On irony and negation. *Discourse processes* 19(2):239–264.

Grice, H. P. 1975. Logic and conversation. *1975* 41–58.

Howe, D. C. 2009. Rita wordnet. java based api to access wordnet.

Ide, N., and Suderman, K. 2004. The American National Corpus First Release. In *Proceedings of the Language Resources and Evaluation Conference*.

Lucariello, J. 2007. Situational irony: A concept of events gone away. *Irony in language and thought* 467–498.

Mihalcea, R., and Pulman, S. G. 2007. Characterizing humour: An exploration of features in humorous texts. In *CICLing*, 337–347.

Mihalcea, R., and Strapparava, C. 2005. Making computers laugh: Investigations in automatic humor recognition. In *HLT/EMNLP*.

Miller, G. A. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Pang, B., and Lee, L. 2008. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* 2(1-2):1–135.

Potts, C. 2011. Developing adjective scales from user-supplied textual metadata. *NSF Workshop on Restructuring Adjectives in WordNet. Arlington,VA*.

Quintilien, and Butler, H. E. 1953. *The Institutio Oratoria of Quintilian. With an English Translation by HE Butler*. W. Heinemann.

Reyes, A.; Rosso, P.; and Veale, T. 2013. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation* 1–30.

Ritchie, G., and Masthoff, J. 2011. The STANDUP 2 interactive riddle builder. In Ventura, D.; Gervás, P.; Harrell, D. F.; Maher, M. L.; Pease, A.; and Wiggins, G., eds., *Proceedings of the Second International Conference on Computational Creativity*, 159.

Ritchie, G. 2003. The jape riddle generator: technical specification. Technical report, University of Edingurgh.

Spell, B. 2009. Java api for wordnet searching (jaws).

Stock, O., and Strapparava, C. 2006. Laughing with hahacronym, a computational humor system. In *AAAI*, 1675–1678.

Taylor, J., and Mazlack, L. 2005. Toward computational recognition of humorous intent. In *Cognitive Science Conference*.

Utsumi, A. 2000. Verbal irony as implicit display of ironic environment: Distinguishing ironic utterances from nonirony. *Journal of Pragmatics* 32(12):1777–1806.

Veale, T., and Hao, Y. 2010a. Detecting ironic intent in creative comparisons. In *ECAI*, volume 215, 765–770.

Veale, T., and Hao, Y. 2010b. An ironic fist in a velvet glove: Creative mis-representation in the construction of ironic similes. *Minds and Machines* 20(4):635–650.

Veale, T. 2004. The challenge of creative information retrieval. In *Computational Linguistics and Intelligent Text Processing*. Springer. 457–467.

Veale, T. 2013. Humorous similes. *Humor* 26(1):3–22.

Venour, C. 2013. *A computational model of lexical incongruity in humorous text*. Ph.D. Dissertation, University of Aberdeen.

Wilson, D., and Sperber, D. 2002. Relevance theory. *Handbook of pragmatics*.

Witten, I. H., and Frank, E. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Knowledge Discovery of Artistic Influences: A Metric Learning Approach

Babak Saleh, Kanako Abe, Ahmed Elgammal
Computer Science Department
Rutgers University
New Brunswick, NJ USA
{babaks,kanakoabe,elgammal}@rutgers.edu

Abstract

We approach the challenging problem of discovering influences between painters based on their fine-art paintings. In this work, we focus on comparing paintings of two painters in terms of visual similarity. This comparison is fully automatic and based on computer vision approaches and machine learning. We investigated different visual features and similarity measurements based on two different metric learning algorithm to find the most appropriate ones that follow artistic motifs. We evaluated our approach by comparing its result with ground truth annotation for a large collection of fine-art paintings.

Introduction

How do artists describe their paintings? They talk about their works using several different concepts. The elements of art are the basic ways in which artists talk about their works. Some of the elements of art include space, texture, form, shape, color, tone and line (Fichner-Rathus). Each work of art can, in the most general sense, be described using these seven concepts. Another important descriptive set is the principles of art. These include movement, unity, harmony, variety, balance, contrast, proportion, and pattern. Other topics may include subject matter, brush stroke, meaning, and historical context. As seen, there are many descriptive attributes in which works of art can be talked about.

One important task for art historians is to find influences and connections between artists. By doing so, the conversation of art continues and new intuitions about art can be made. An artist might be inspired by one painting, a body of work, or even an entire genre of art is this influence. Which paintings influence each other? Which artists influence each other? Art historians are able to find which artists influence each other by examining the same descriptive attributes of art which were mentioned above. Similarities are noted and inferences are suggested.

It must be mentioned that determining influence is always a subjective decision. We will not know if an artist was ever truly inspired by a work unless he or she has said so. However, for the sake of finding connections and progressing through movements of art, a general consensus is agreed upon if the argument is convincing enough. Figure 1 represents a commonly cited comparison for studying influence.

Figure 1: An example of an often cited comparison in the context of influence. Diego Velázquez's Portrait of Pope Innocent X (left) and Francis Bacon's Study After Velázquez's Portrait of Pope Innocent X (right). Similar composition, pose, and subject matter but a different view of the work.

Is influence a task that a computer can measure? In the last decade there have been impressive advances in developing computer vision algorithms for different object recognition-related problems including: instance recognition, categorization, scene recognition, pose estimation, etc. When we look into an image we not only recognize object categories, and scene category, we can also infer various cultural and historical aspects. For example, when we look at a fine-art paining, an expert or even an average person can infer information about the genre of that paining (e.g. Baroque vs. Impressionism) or even can guess the artist who painted it. This is an impressive ability of human perception for analyzing fine-art paintings, which we approach to it in this paper as well.

Besides the scientific merit of the problem from the perception point of view, there are various application motivations. With the increasing volumes of digitized art databases on the internet comes the daunting task of organization and retrieval of paintings. There are millions of paintings present on the internet. It will be of great significance if we can infer new information about an unknown painting using already existing database of paintings and as a broader view can in-

Figure 2: Gustav Klimt's *Hope* (Top Left) and nine most similar images across different styles based on LMNN metric. Top row from left to right: "Countess of Chinchon" by Goya; "Wing of a Roller" by Durer; "Nude with a Mirror" by Mira; "Jeremiah lamenting the destruction of Jerusalem" by Rembrandt. Lower row, from left to right: "Head of a Young Woman" by Leonardo Da Vinci; "Portrait of a condottiere" by Bellini; "Portrait of a Lady with an Ostrich Feather Fan" by Rembrandt; "Time of the Old Women" by Goya and "La Schiavona" by Titian.

fer high-level information like influences between painters. Although there have been some research on automated classification of paintings (Arora and Elgammal 2012; Cabral et al. 2011; Carneiro 2011; Li et al. 2012; Graham 2010). However, there is very little research done on measuring and determining influence between artists ,e.g. (Li et al. 2012). Measuring influence is a very difficult task because of the broad criteria for what influence between artists can mean. As mentioned earlier, there are many different ways in which paintings can be described. Some of these descriptions can be translated to a computer. Some research includes brushwork analysis (Li et al. 2012) and color analysis to determine a painting style. For the purpose of this paper, we do not focus on a specific element of art or principle of art but instead we focus on finding new comparisons by experimenting with different similarity measures.

Although the meaning of a painting is unique to each artist and is completely subjective, it can somewhat be measured by the symbols and objects in the painting. Symbols are visual words that often express something about the meaning of a work as well. For example, the works of Renaissance artists such as Giovanni Bellini and Jan Van-Eyck use religious symbols such as a cross, wings, and animals to tell stories in the Bible.

One important factor of finding influence is therefore having a good measure of similarity. Paintings do not necessarily have to look alike but if they do or have reoccurring objects (high-level semantics), then they will be considered similar. However similarity in fine-art paintings is not limited to the co-occurrence of objects. Two abstract paintings look quite similar even though there is no object in any of them. This clarifies the importance of low-level features for painting representation as well. These low-level features are able to model artistic motifs (e.g. texture, decomposition and negative space). If influence is found by looking at similar characteristics of paintings, the importance of finding a good similarity measure becomes prominent. Time is also a necessary factor in determining influence. An artist cannot influence another artist in the past. Therefore the linearity of paintings cuts down the possibilities of influence.

By including a computer's intuition about which artists and paintings may have similarities, it not only finds new knowledge about which paintings are connected in a mathematical criteria but also keeps the conversation going for artists. It challenges people to consider possible connections in the timeline of art history that may have never been seen before. We are not asserting truths but instead suggesting a possible path towards a difficult task of measuring influence.

The main contribution of this paper is working on the interesting task of determining influence between artist as a knowledge discovery problem. Toward this goal we propose two approaches to represent paintings. On one hand high-level visual features that correspond to objects and concepts in the real world have been used. On the other hand we extracted low-level visual features that are meaningless to human, but they are powerful for discrimination of paintings using computer vision algorithms. After image representation we need to define similarity between pairs of artist based on their artworks. This results in finding similarity at the level of images. Since the first representation is mean-

Figure 3: Gustav Klimt's *Hope* (Top Left) and nine most similar images across different styles based on Boost metric. Top row from left to right: "Princesse de Broglie" by Ingres; "Portrait, Evening (Madame Camus)" by Degas; "The birth of Venus-Detail of Face" by Botticelli; "Danae and the Shower of Gold" by Titian. Lower row from left to right: "The Burial of Count Orgasz" by El Greco; "Diana Callist" by Titian; "The Starry Night" by Van Gogh; "Baronesss Betty de Rothschild" by Ingres and "St Jerome in the Wilderness" by Durer.

ingful by its nature (a set of objects and concepts in the images) we do not need to learn a semantically meaningful way of comparison. However for the case of low-level representation we need to have a metric that covers the absence of semantic in this type of image representation. For the latter case we investigated a set of complex metrics that need to be learned specifically for the task of influence determination.

Because of the limited size of the available influence ground-truth data and the lack of negative examples in it, it is not useful for comparing different metrics. Instead, we resort to a highly correlated task, which is classifying painting style. The assumption is that metrics that are good for style classification (which is a supervised learning problem), would also be good for determining influences (which is an unsupervised problem). Therefore, we use painting style label to learn the metrics. Then we evaluate the learned metrics for the task of influence discovery by verifying the output using well-known influences.

Related Works

Most of the work done in the area of computer vision and paintings analysis utilizes low-level features such as color, shades, texture and edges for the task of style classification. Lombardi (Lombardi 2005) presented a comprehensive study of the performance of such features for paintings classification. Sablatnig et al. (R. Sablatnig and Zolda 1998) uses brush-strokes patterns to define structural signature to identify the artist style. Khan et al. (Fahad Shahbaz Khan 2010) use a Bag of Words(BoW) approach with low-level features of color and shades to identify the painter among eight different artists. In (Sablatnig, Kammerer, and Zolda 1998) and (I. Widjaja and Wu. 2003) also similar experiments with low-level features were conducted.

Carneiro et al. (Carneiro et al. 2012) recently published the dataset "PRINTART" on paintings along with primarily experiments on image retrieval and painting style classification. They define artistic image understanding as a process that receives an artistic image and outputs a set of global, local and pose annotations. The global annotations consist of a set of artistic keywords describing the contents of the image. Local annotations comprise a set of bounding boxes that localize certain visual classes, and pose annotations consist of a set of body parts that indicate the pose of humans and animals in the image. Another process involved in the artistic image understanding is the retrieval of images given a query containing an artistic keyword. In. (Carneiro et al. 2012) an improved inverted label propagation method has been proposed that produces the best results, both in the automatic (global, local and pose) annotation and retrieval problems.

Graham et. al. (Graham 2010) pose the question of finding the way we perceive two artwork similar to each other. Toward this goal, they acquired strong supervision of human experts to label similar paintings. They apply multidimensional scaling methods to paired similar paintings from either Landscape or portrait/still life and showed that similarity between paintings can be interpreted as basic image statistics. In the experiments they show that for landscape paintings, basic grey image statistics is the most important factor for two artwork to be similar. For the case of still life/portrait most important element of similarity is seman-

tic variable, for example representation of people.

Extracting visual features for paintings is very challenging that should be treated differently from feature representation of natural images. This difference is due to, first unlike regular images(e.g. personal photographs), paintings have been created by involving abstract ideas. Secondly the effect of digitization on the computational analysis of paintings is investigated in great depth by Polatkan et. al (Gungor Polatkan 2009).

Cabral et al (Cabral et al. 2011) approach the problem of ordering paintings and estimating their time period. They formulate this problem as embedding paintings into a one dimensional manifold. They applied unsupervised embedding using Laplacian Eignemaps (Belkin and Niyogi 2002). To do so they only need visual features and defined a convex optimization to map paintings to a manifold.

Influence Framewrok

Consider a set of artists, denoted by $A = \{a^l, l = 1 \cdots N_a\}$, where N_a is the number of artists,. For each artist, a^l, we have a set of images of paintings, denoted by $P^l = \{p_i^l, i = 1, \cdots, N^l\}$, where N^l is the number of paintings for the l-th artist. For clarity of the presentation, we reserve the superscript for the artist index and the subscript for the painting index. We denote by $N = \sum_l N_l$ the total number of paintings. Therefore, each image $p_i^l \in R^D$ is a D dimensional feature vector that is the outcome of the Classemes classifiers, which defines the feature space.

To represent the temporal information, for each artist we have a ground truth time period where he/she has performed their work, denoted by $t^l = [t^l_{start}, t^l_{end}]$ for the l-th artist, where t^l_{start} and t^l_{end} are the start and end year of that time period respectively. We do not consider the date of a given painting since for some paintings the exact time is unknown.

Painting Similarity:

To encode similarity/dissimilarity between paintings, we consider two different category of approaches. On one hand we applied simple distance metrics (note that distance is dissimilarity measure) on top of high-level visual features(we used Classemes features) as they are understandable by human. On the other hand we applied complex metrics on low-level visual features that are powerful for machine learning, however they don not make sense to human. Details on the features used will be explained in experiment section.

Predefined Similarity Measurement

Euclidean distance: The distance $d_E(p_i^l, p_j^k)$ is defined to be the Euclidean distance between the Classemes feature vectors of paintings p_i^l and p_j^k. Since Classemes features are high-level semantic features, the Euclidean distance in the feature space is expected to measure dissimilarity in the subject matter between paintings. Painting similarity based on the Classemes features showed some interesting cases, several of which have not been studied before by art historians as a potential comparison.

Metric Learning Approaches:

Despite the simplicity, Euclidean distance is not taking into account expert supervision for comparing two paintings together. We approach measuring similarity between two paintings by enforcing expert knowledge about fine art paintings. The purpose of Metric Learning is to find some pair-wise real valued function $d_M(x, x')$ which is non-negative, symmetric, obeys the triangle inequality and returns zero if and only if x and x' are the same point. Training such a function in a general form can be seen as the following optimization problem:

$$\min_M l(M, D) + \lambda R(M) \tag{1}$$

This optimization has two sides, first it minimizes the amount of loss by using metric M over data samples D while trying to adjust the model by the regularization term $R(M)$. The first term shows the accuracy of the trained metric and second one estimates its capability over new data and avoids overfitting. Based on the enforced constraints, the resulted metric can be linear or non-linear, also based on the amount of used labels training can be supervised or unsupervised.

For consistency over the metric learning algorithms, we need to fix the notation first. We learn the matrix M that will be used in Generalized Mahalanobis Distance: $d_M(x, x') = \sqrt{(x - x')' M (x - x')}$, where M by definition is a semi-positive definite matrix.

Dimension reduction methods can be seen as learning the metric when M is a low rank matrix. There has been some research on "Unsupervised Dimension Reduction" for fine-art paintings. We will show how the supervised metric learning algorithms beat the unsupervised approaches for different tasks. More importantly, there are significantly important information in the ground-truth annotation associated with paintings that we use to learn a more reliable metric in a supervised fashion for both the linear and non-linear case.

Considering the nature of our data that has high variations due to the complex visual features of paintings and labels associated with paintings, we consider the following approaches that differ based on the form of M or amount of regularization.

Large Margin Nearest Neighbors (Weinberger and Saul 2009) LMNN is a widely used approach for learning a Mahalanobis distance due to its global optimum solution and its superior performance in practice. The learning of this metric involves a set of constrains, all of which are defined locally. This means that LMNN enforce the k nearest neighbor of any training instance should belong to the same class(these instances are called "target neighbors"). This should be done while all the instances of other classes, ,referred as "Impostors", should keep a way from this point. For finding the target neighbors, Euclidean distance has been applied to each pair of samples, resulting in the following formulation:

$$\min_M (1 - \mu) \sum_{(x_i, x_j) \in T} d^2_M(x_i, x_j) + \mu \sum_{i,j,k} \eta_{i,j,k}$$

$$s.t. : d^2_M(x_i, x_k) - d^2_M(x_i, x_j) \geq 1 - \eta_{i,j,k} \forall (x_i, x_j, x_k) \in I.$$

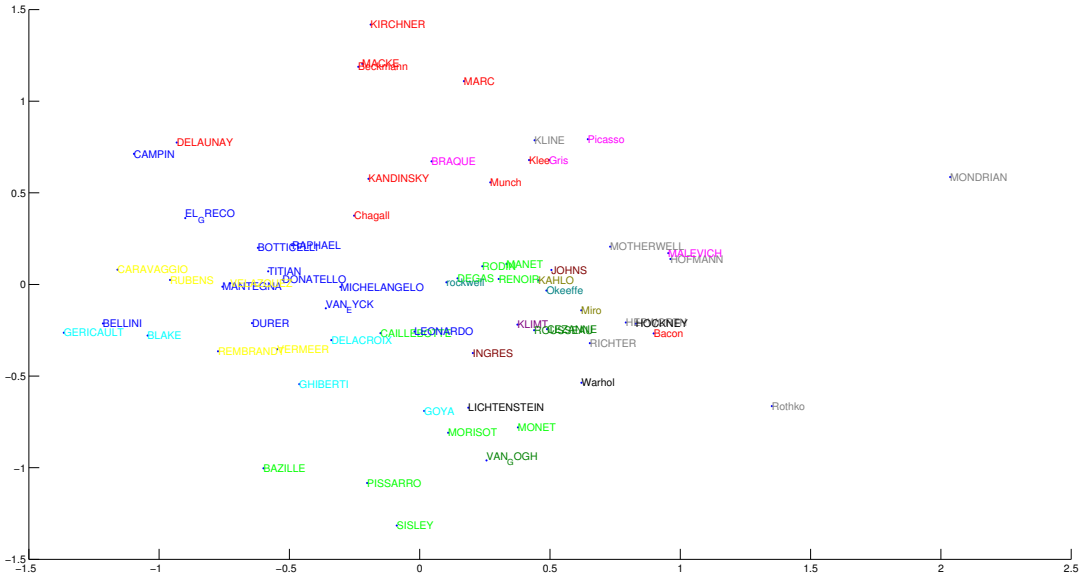


Figure 4: Map of Artists based on LMNN metric between paintings. Color coding indicates artists of the same style.

Where T stands for the set of *Target* neighbors and I represents *Impostors*. Since these constrains are locally defined, this optimization leads to a convex formulation and a global solution. This metric learning approach is related to Support Vector Machines in principle, which theoretically engages its usage along with Support Vector Machines for different tasks including style classification. Due to its popularity, different variations of this method have been expanded, including a non linear version called gb-LMNN (Weinberger and Saul 2009) which we will use in our experiments as well.

Boost Metric (Shen et al. 2012) This approach is based on the fact that a Semi-Positive Definite matrix can be decomposed into a linear combination of trace-one rank-one matrices. Shen et al (Shen et al. 2012) use this fact and instead of learning M, find a set of weaker metrics that can be combined and give the final metric. They treat each of these matrices as a *Weak Learner*, which is used in the literature of Boosting methods. The resulting algorithm is applying the idea of AdaBoost to Mahalanobis distance, which is quiet efficient in practical usages. This method is particularly of our interest, since we can learn an individual metric for each style of painting and finally merge these metric to get the final one. Theoretically the final metric can perform well to find similarities inside each painting style as well.

We considered the aforementioned types of metrics(Boost metric and LMNN) for measuring similarity between paintings. On one hand it is been stated (Weinberger and Saul 2009) that "Large Margin Nearest Neighbors" outperforms other metrics for the task of classification. This is rooted in the fact that this metric imposes the largest margin between different classes. Considering this property of LMNN, we expect it to outperform other methods for the task of painting's style classification. On the other hand, as it is mentioned in the introduction, artists compare paintings based on a list of criteria. Assuming we can model each criteria via a *Weak Learner*, we can combine these metrics using Boost metric learning. We argue that searching for similar paintings based on this metric would be more realistic and intuitive.

Artist Similarity:

Once painting similarity is encoded, using any of aforementioned methods, we can design a suitable similarity measure between artists. There are two challenges to achieve this task. First, how to define a measure of similarity between two artists, given their sets of paintings. We need to define a proper set distance $D(P^l, P^k)$ to encode the distance between the work of the l-th and k-th artists. This relates to how to define influence between artists to start with, where there is no clear definition. Should we declare an influence if one paining of artist k has strong similarity to a painting of artist l ? or if a number of paintings have similarity ? and what that "number" should be ?

Mathematically speaking, for a given painting $p_i^l \in P^l$ we can find its closest painting in P^k using a point-set distance as

$$d(p_i^l, P^k) = \min_j d(p_i^l, p_j^k).$$

We can find one painting in by artist l that is very similar to a painting by artist k, that can be considered an influence. This dictates defining an asymmetric distance measure in the

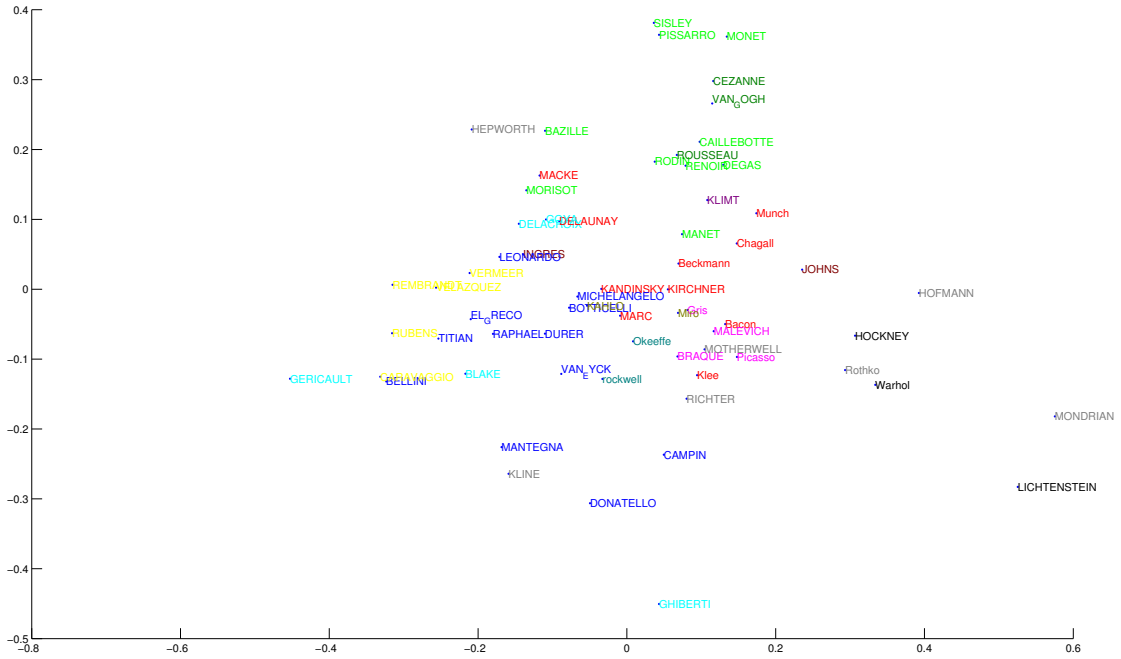


Figure 5: Map of Artists based on Boost metric between paintings. Color coding indicates artists of the same style.

form of

$$D_{min}(P^l, P^k) = \min_i d(p_i^l, P^k).$$

We denote this measure by *minimum link influence*.

On the other hand, we can consider a central tendency in measuring influence, where we can measure the average or median of painting distances between P^l and P^k, we denote this measure *central-link influence*.

Alternatively, we can think of Hausdorff distance (Dubuisson and Jain 1994), which measures the distance between two sets as the supremum of the point-set distances, defined as

$$D_H(P^l, P^k) = \max(\max_i d(p_i^l, P^k), \max_j d(p_j^k, P^l)).$$

We denote this measure *maximum-link influence*. Hausdorff distance is widely used in matching spatial points, which unlike a minimum distance, captures the configuration of all the points. While the intuition of Hausdorff distance is clear from a geometrical point of view, it is not clear what it means in the context of artist influence, where each point represent a painting. In this context, Hausdorff distance measures the maximum distance between any painting and its closest painting in the other set.

The discussion above highlights the challenge in defining the similarity between artists, where each of the suggested distance is in fact meaningful, and captures some aspects of similarity, and hence influence. In this paper, we do not take a position in favor of any of these measures, instead we propose to use a measure that can vary through the whole spectrum of distances between two sets of paintings. We define asymmetric distance between artist l and artist k as the q-percentile Hausdorff distance, as

$$D_{q\%}(P^l, P^k) = \max_i^{q\%} d(p_i^l, P^k). \quad (2)$$

Varying the percentile q allows us to evaluate different settings ranging from a minimum distance, D_{min}, to a central tendency, to a maximum distance D_H.

Experimental Evaluation

Evaluation Methodology:

We used dataset of fine-art paintings (Abe, Saleh, and Elgammal 2013) for our experiments. This collection contains color images from 1710 paintings of 66 artist created during the time period of 1400-1935. This dataset covers all genres and thirteen styles of paintings(e.g. classic, abstract).

This dataset has some known influences between artists within the collection from multiple resources such as *The Art Story Foundation* and *The Metropolitan Museum of Art*. For example, there is a general consensus among art historians that Paul Cézanne's use of fragmented spaces had a large impact on Pablo Picasso's work. In total, there are 76 pairs of one-directional artist influences, where a pair (a^i, a^j) indicates that artist i is influenced by artist j. Generally, it is a sparse list that contains only the influences which are consensual among many. Some artists do not have any influences in our collection while others may have up to five.

We use this list as ground-truth for measuring the accuracy of our experiments. There is an agreement that influence happens mostly when two paintings belong to the same style (e.g. both are classic). Inspired by this fact we used the annotation of paintings to put paintings from same style close to each other, when we learn a metric for similarity measurement between paintings.

Learning the Painting Similarity Measure

We experimented with the Classemes features (Torresani, Szummer, and Fitzgibbon 2010), which represents the high level information in terms of presence/absence of objects in the image. We also extracted GIST descriptors (Oliva and Torralba 2001) and Histogram of Oriented Gradients (HOG) (Dalal and Triggs 2005), since they are the main ingredients in the Classemes features. For the task of measuring the similarity between paintings, we followed two approaches: First, we investigated the result of applying a predefined metric (Euclidean) on extracted visual features. Second, for low-level visual features(HOG and GIST), we learned a new set of metrics to put similar images from same style close to each other. These metrics are learned in the way that we expect to see paintings from same style be the most similar pairs of paintings. However it is interesting to look at most similar pairs of paintings when their style is different. Toward this goal we computed the distance between all the possible pairs of paintings based on learned Boost metric and LMNN metric. Some of the most similar pairs across different styles(with the smallest distances) are depicted in figure 9(for LMNN metric) and figure 8 for Boosting metric approach.

We also evaluated these metrics for the task of painting retrieval. Figure 2 shows the top nine closest matches for the *Hope* by Klimt when we used LMNN metric to learn the measure of similarity between paintings. Figure 3 represents results of the same task when we used Boost metric approach instead of LMNN. Although the retrieved results are from different styles, but they show different aspects of similarities, in color, texture, composition, subject matter, etc.

Painting Style Classification

To verify the performance of these learned metrics for measuring similarity, we compared their accuracy for the task of style classification of paintings. We train a set of one-vs-all classifiers using Support Vector Machines(SVM) after applying different similarity measurements. Each classifier corresponds to one painting style and in total we trained 13 classifiers using LIBSVM package (Chang and Lin 2011). Performance of these classifiers are reported in table 1 in terms of average and the standard deviation of the accuracy. We compared our implementations with the method of (Arora and Elgammal 2012) as the baseline. Both variations of LMNN method (linear and non-linear)that are trained on low-level visual features outperform the baseline. However the trained classifier based on measure of similarity of Boosting metric performs slightly worse than the baseline.

Table 1: Style Classification Accuracy

Method	LMNN	gb-LMNN	Boost Metric	Baseline
Accuracy mean(%)	69.75	68.16	64.71	65.4
std	4.13	3.52	3.06	4.8

Influence Discovery Validation

As mentioned earlier, based on similarity between paintings, we measure how close are works of an artist to another and build an influenced-by-graph by considering the temporal information. The constructed influenced-by graph is used to retrieve the top-k potential influences for each artist. If a retrieved influence concur with an influence ground-truth pair that is considered a hit. The hits are used to compute the recall, which is defined as the ratio between the correct influence detected and the total known influences in the ground truth. The recall is used for the sake of comparing the different settings relatively. Since detected influences can be correct although not in our ground truth, so no meaning to compute the precision.

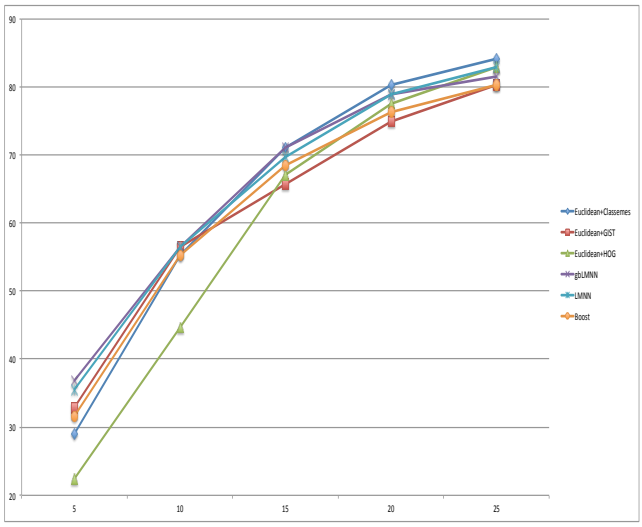


Figure 6: Recall curves of top-k(x-axis values) influences for different approaches when $q = 50$.

In all cases, we computed the recall figures using the influence graph for the top-k similar artist (k=5, 10, 15, 20, 25) with different q-percentile for the artist distance measure in Eq 2 (q=1, 10, 50, 90, 99%). Figure 6 shows this recall curve for the case of $q = 50$ and figure 7 depicts the recall curve of influence finding when $q = 90$.

We computed the performance of different approaches for the task of influence finding when the value of K is fixed($K = 5$). Since these are supposed to be the most similar artists, which can suggest potential influences. Table 2 compares the performance of these approaches for different values of percentile (q) for a given k. Except the case of $q = 10$, gb-LMNN gives the bet performance.

Table 2: Comparison of Different Methods for Finding Top-5 Influence

	q%				
Method	1	10	50	90	99
Euclidean on Classemes features	25	26.3	29	21.1	23.7
Euclidean on GIST features	21.05	**31.58**	32.89	28.95	23.68
Euclidean on HOG features	22.37	22.37	22.37	25	26.32
gb-LMNN on low-level features	**27.63**	22.37	**36.84**	**35.53**	**30.26**
LMNN on low-level features	23.68	22.37	35.53	35.53	28.95
Boost on low-level features	21.05	28.95	31.58	30.26	27.63

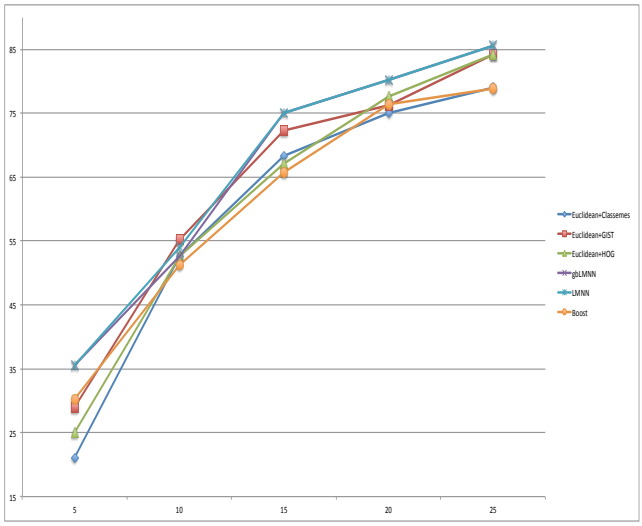


Figure 7: Recall curves of top-k(x-axis values) influences for different approaches when $q = 90$.

As mentioned earlier based on similarity of paintings and following time period of each artist, we are able to build a map of painters. For computing the similarity between collection of paintings of an artist, we looked for the 50 percentile of his works ($q = 50$) and built the map of artist based on LMNN metric (shown in figure 4) and Boost metric (figure 5). For the sake of better visualization, we depict artist from the same style with one color. The fact that artist from the same style stay close to each other verifies the quality of these maps.

Conclusion

In this paper we explored the interesting problem of finding potential influences between artist. We considered painters and tried to find who can be influenced by whom, based on their artworks and without any additional information. We approached this problem as a similarity measurement in the area of computer vision and investigated different metric learning methods for representing paintings and measuring their similarity to each other. This similarity measurement is in-line with human perception and artistic motifs. We experimented on a diverse collection of pairings and reported interesting findings.

Acknowledgment

We would like to appreciate valuable inputs by Mr. Shahriar Rokhgar for his precious comments on painting analysis. Also we thank Dr. Laura Morowitz for her comments on finding the influence path in art history.

References

Abe, K.; Saleh, B.; and Elgammal, A. 2013. An early framework for determining artistic influence. In *ICIAP Workshops*, 198–207.

Arora, R. S., and Elgammal, A. M. 2012. Towards automated classification of fine-art painting style: A comparative study. In *ICPR*.

Belkin, M., and Niyogi, P. 2002. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15:1373–1396.

Cabral, R. S.; Costeira, J. P.; De la Torre, F.; Bernardino, A.; and Carneiro, G. 2011. Time and order estimation of paintings based on visual features and expert priors. In *SPIE Electronic Imaging, Computer Vision and Image Analysis of Art II*.

Carneiro, G.; da Silva, N. P.; Bue, A. D.; and Costeira, J. P. 2012. Artistic image classification: An analysis on the printart database. In *ECCV*.

Carneiro, G. 2011. Graph-based methods for the automatic annotation and retrieval of art prints. In *ICMR*.

Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27.

Dalal, N., and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *International Conference on Computer Vision & Pattern Recognition*, volume 2, 886–893.

Dubuisson, M.-P., and Jain, A. K. 1994. A modified hausdorff distance for object matching. In *Pattern Recognition*.

Fahad Shahbaz Khan, Joost van de Weijer, M. V. 2010. Who painted this painting?

Fichner-Rathus, L. *Foundations of Art and Design*. Clark Baxter.

Graham, D., F. J. R. D. 2010. Mapping the similarity space of paintings: image statistics and visual perception. *Visual Cognition*.

Gungor Polatkan, Sina Jafarpour, A. B. S. H. I. D. 2009. Detection of forgery in paintings using supervised learning. In *16th IEEE International Conference on Image Processing (ICIP)*, 2921 – 2924.

I. Widjaja, W. L., and Wu., F. 2003. Identifying painters from color profiles of skin patches in painting images. In *ICIP*.

Li, J.; Yao, L.; Hendriks, E.; and Wang, J. Z. 2012. Rhythmic brushstrokes distinguish van gogh from his contemporaries: Findings via automated brushstroke extraction. *IEEE Trans. Pattern Anal. Mach. Intell.*

Lombardi, T. E. 2005. The classification of style in fine-art painting. *ETD Collection for Pace University. Paper AAI3189084.*

Oliva, A., and Torralba, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision* 42:145–175.

R. Sablatnig, P. K., and Zolda, E. 1998. Hierarchical classification of paintings using face- and brush stroke models. In *ICPR*.

Sablatnig, R.; Kammerer, P.; and Zolda, E. 1998. Structural analysis of paintings based on brush strokes. In *Proc. of SPIE Scientific Detection of Fakery in Art*. SPIE.

Shen, C.; Kim, J.; Wang, L.; and van den Hengel, A. 2012. Positive semidefinite metric learning using boosting-like algorithms. *Journal of Machine Learning Research* 13:1007–1036.

Torresani, L.; Szummer, M.; and Fitzgibbon, A. 2010. Efficient object category recognition using classemes. In *ECCV*.

Weinberger, K. Q., and Saul, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *JMLR*.

Figure 8: Five most similar pairs of paintings across different styles based on Boost Metric
First row: “The Garden Terrace at Les Lauves” by Cezanne (left) and “View of Delft” by Vermer (right)
Second row: “Portrait of a Lady” by Klimt (left); “Head of a Young Woman” by Da Vinci (right)
Third row: “Head” by Da Vinci (left) and “The Artist and his Wife” by Ingres (right)
Fourth row: “The Wire-drawing Mill” by Durer (left) and “Un village” by Morisot (right)

Figure 9: Five most similar pairs of paintings across different styles based on LMNN Metric
First row: “Girl in a Chemise” by Picasso (left) and “Madame Czanne in Blue” by Cezanne (right)
Second row: “The Burial of Count Orgasz; Detail of pointing boy” by El Greco (left) and “Young Girl with a Parrot” by Morisot
Third row: “Lady in a Green Jacket” by Macke (left) and “Two Young Peasant Women” by Pissaro (right)
Fourth row: “The Feast of the Gods” by Bellini (left) and “Burial of the Sardine” by Goya (right)

Nehovah: A Neologism Creator Nomen Ipsum

Michael R. Smith, Ryan S. Hintze and **Dan Ventura**
Department of Computer Science, Brigham Young University, Provo, UT 84602
msmith@axon.cs.byu.edu, ventura@cs.byu.edu

Abstract

In this paper, we describe a system called *Nehovah* that generates neologisms from a set of base words provided by a user. Nehovah focuses on creating "good" neologisms by evaluating various attributes of a neologism such as how well it communicates the source concepts and how "catchy" it is. Because Nehovah depends on the user to weight the importance of various attributes of the neologism and to choose the source concepts, it is at this point most appropriately considered a collaborative system rather than an autonomous one. To demonstrate the utility of the system, we show several examples of system output and discuss the creativity of Nehovah with respect to several characteristics critical for any computational creative system: appreciation, imagination, skill and accountability.

Introduction

Boden (1994) made one of the first attempts to formalize the notion of creativity. Based on her formalization, computational creativity is often thought of as an exploration of a conceptual space and has been examined in a number of different areas including visual art (Colton, Valstar, and Pantic 2008; Norton, Heath, and Ventura 2011), music (Cope 2005), cooking (Morris et al. 2012), poetry (Rahman and Manurung 2011), metaphor generation (Veale and Hao 2007), and sentence generation (Mendes, Pereira, and Cardoso 2004). In this paper, we describe *Nehovah*, a computational system that generates neologisms.

The generation of neologisms is an important task in many businesses to create a unique brand or company name to distinguish it from its competitors. This often comes in the form of a trademark. Trademarks include words, phrases, symbols and/or designs that identify and distinguish the goods of one party from those of others[1]. According to the United States Patent and Trademark Office, 433,651 trademark applications were filed in 2013; a 4.5% increase from 2012 (The United States Patent and Trademark Office 2014). Thus, developing trademarkable phrases and words is a important step in many businesses.

Additionally, neologisms are often used as a literary device in novels and books to convey meaning more concisely. For example, "cyberspace" was introduced in 1982 by William Gibson to combine the words "cybernetics" and "space" (Gibson 1982). In some cases, neologisms are used to add humor and interest. This technique was used heavily in the many works of Dr. Seuss to help children with limited vocabularies to enjoy reading (Baker 1999).

Neologisms have previously been examined computationally, both from an interpretive standpoint and from a generative one. For example, Cook and Stevenson (2010) propose finding the meaning of neologisms using a statistical model that draws on observed linguistic properties of blends, while Duch and Pilichowski (2007) create neologisms using a neurocognitive model (though, unfortunately, many of the generated neologisms exhibit little to no linguistic/conceptual/cognitive value).

Veale's *Zeitgeist* system rather impressively exhibits both interpretive and generative abilities and is available as a web application. It can be used as a tool for enriching lexical resources such as WordNet (Fellbaum 1998) with modern words that are found in every day speech (Veale 2006) by utilizing Wikipedia[2] to identify neologisms and by reverse engineering their source words using ideas from concept blending (Veale, O'Donoghue, and Keane 2000).

In addition, the Zeitgeist system can be used to generate neologisms by combining prefix and suffix morphemes that overlap by at least one letter (Veale and Butnariu 2006). Morphemes are hand-annotated with their semantic interpretations giving each morpheme a word gloss (such as "astro"="star" and "ology"="study") and a WordNet identifier that indicates where in the WordNet noun taxonomy a neologism with a morphemic suffix should be placed. Given two source words from predefined lists for prefixes and suffixes, the Zeitgeist system creates a set of neologisms that convey the chosen concepts by combining the prefix and suffix morphemes for the source words. The generated neologisms generally have valid word forms and convey the concepts well. On the other hand, Zeitgeist is limited to the morphemes that are annotated. As many of the morphemes are of Greek origin, some of the neologisms are somewhat predictable. For example, if "food" is chosen as a source prefix word, then "gastro" is almost always used. The use of morphemes also requires a knowledge of Greek or Latin word derivatives to understand the neologism. The neolo-

[1] http://www.uspto.gov/trademarks/basics/definitions.jsp

[2] www.wikipedia.com

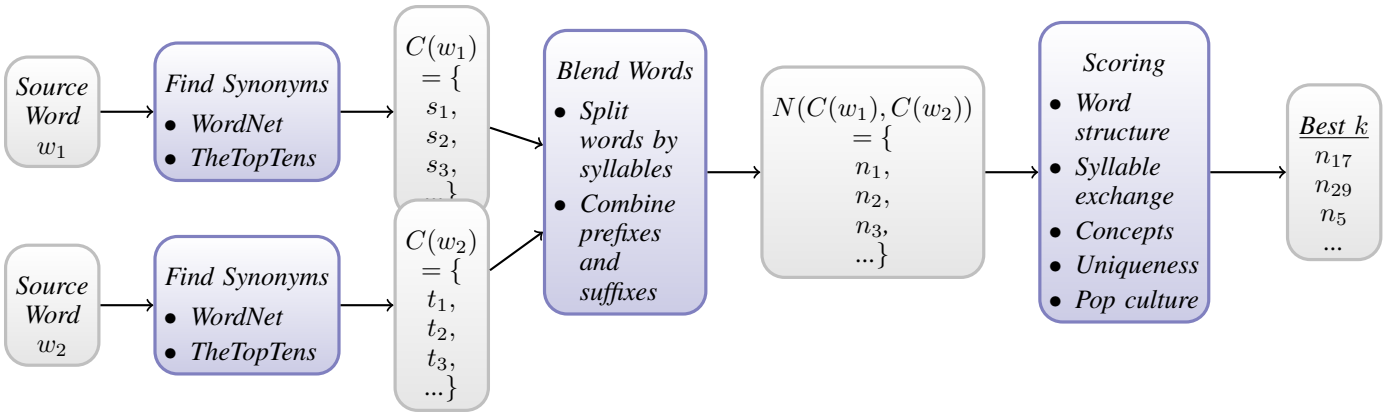


Figure 1: A high-level pipeline view of the process Nehovah uses to generate neologisms through finding synonyms, blending words, and scoring.

gism "ornithoencephalon" is a neologism for "bird-brain" but the meaning is obvious only to the user who knows that the morpheme "ornitho" relates to birds and "encephalon" relates to the brain.

Our system for generating neologisms, Nehovah, is similar to Zeitgeist in that it attempts to preserve the source concepts through blending (as opposed to generating neologisms that represent entirely new ideas by themselves, e.g. "Google"). It differs from Zeitgeist by focusing on blending free-form, user-provided words and their synonyms and by incorporating dynamic web sources of popular cultural information. In addition, the web interface allows a user to weight the importance of several attributes of a neologism, facilitating a creative collaboration between the user and the system.

A Framework for Blending Concepts

The goal of generating neologisms by blending concepts from source words is to convey multiple concepts in a single plausible word, sometimes known as a *portmanteau* (Carroll 1871). We present a framework, containing three major steps, for generating such portmanteau neologisms from two source words:

1. **Finding Synonyms.** Synonyms increase the potential novelty of the neologisms by enriching the set of possible blends that convey the source concept. A greater diversity of synonyms expresses more imagination in the neologism. For example, the word "God" is arguably a more diverse/interesting synonym for "creator" than is the word "maker". We call the set of synonyms for a source word w_i the *concept set* for w_i and denote it as $C(w_i)$. Note that it is always the case that $w_i \in C(w_i)$.
2. **Blending Words.** Once the concept sets for the source words have been generated, the words from each concept set are blended together to create a set of neologisms. Blending the words from the two concept sets consists of three steps. First, each word from the concept sets is split into sets of prefixes and suffixes. Then, each prefix from one concept set is joined with each suffix from the other concept set. Finally, Nehovah checks that the word structure of the neologism is plausible. By plausible, we mean that the letter sequence produced from blending the words is natural compared to other "real" words. Any implausible neologism is discarded. The set of neologisms generated from two concept sets $C(w_1)$ and $C(w_2)$ is denoted $N(C(w_1), C(w_2))$.
3. **Scoring/ranking the Neologisms.** Once a set of neologisms $N(C(w_1), C(w_2))$ is created, they are scored or ranked such that a subset of "best" neologisms can be identified, allowing a potentially large set of neologisms to be quickly filtered. Scoring criteria can be adapted for a particular application and can also potentially incorporate feedback, facilitating online learning and thus dynamic qualification of neologisms.

Nehovah

A functional overview of Nehovah and its implementation of the three steps are shown in Figure 1 and are described in more detail in the following sections. The blue boxes represent each step in the framework for blending concepts and the gray boxes represent sets of words. An on-line version of Nehovah is available at

`http://axon.cs.byu.edu/~nehovah`

from which a screen shot is shown in Figure 2.

Finding Synonyms

In order to populate the set $C(w_i)$, Nehovah searches for synonyms from two different sources: WordNet (Fellbaum 1998) (a lexical database) and TheTopTens[3] (a website of pop culture-inspired "top ten" lists).

Nehovah queries WordNet with each source word w_i (and with its stem) as a noun, verb, adjective, and adverb. If a source word or its stem is defined in WordNet, Nehovah adds to $C(w_i)$ the words contained in the synset for all senses of the word for all parts-of-speech for which it is defined.

[3] `www.thetoptens.com`

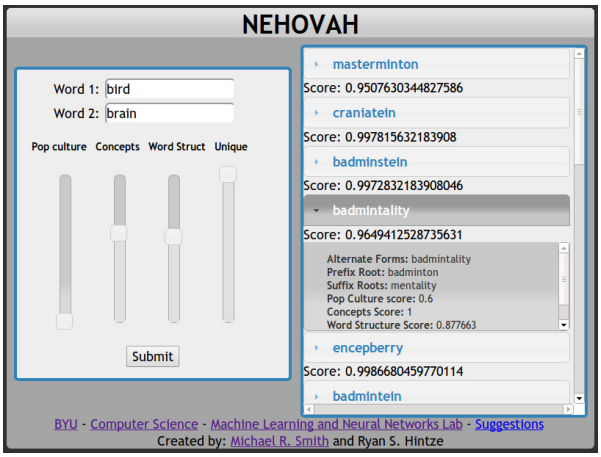


Figure 2: A screenshot of the web interface for Nehovah. Two source words are input in the upper left. The lower left contains sliders that allow relative weighting of the four scoring attributes. On the right is a list of generated neologisms with their scores, in descending order, and these can be expanded to see the base words that Nehovah used to create the neologism and how the neologism is scored for each of the attributes.

For example, the word "school" as a noun has the following senses:

- "school→educational institution"
- "school, schoolhouse→building, edifice"
- "school, schooling→education"
- "school→body"
- "school, schooltime, school day→time period, period of time, period"
- "school, shoal→animal group"

and, additionally as a verb has the following senses:

- "school→educate"
- "educate, school, train, cultivate, civilize, civilise→polish, refine, fine-tune, down"
- "school→swim"

(it has no senses as either an adjective or adverb). Therefore, the set of WordNet-derived synonyms for the word "school", $C(\text{"school"}) = \{$school, educational institution, schoolhouse, building, edifice, schooling, education, body, school time, school day, time period, period of time, period, shoal, animal group, educate, train, cultivate, civilize, polish, refine, fine-tune, down, swim$\}$.

Because a source word is specified without context, neither its part-of-speech nor its intended sense can be inferred, and, as a result the space of possible synonyms is increased, providing greater creative potential in the generated neologisms at the risk of potentially conveying an awkward or unintended conceptual blend.

Nehovah queries TheTopTens with each source word w_i using a custom API that returns lists of words from a set of "top ten" lists that match the query. For example, a query to TheTopTens using the source word "car" would return lists with titles such as "Top Ten Best Car Companies," "Best Car Brands," "Greatest Songs by the Cars" and "Best Car Insurance Companies."

Of course, some lists will be much more relevant than others. To minimize the number of included irrelevant words, Nehovah determines which of the returned lists are relevant based on their titles, by the identifying descriptive and plural words in the title. Descriptive words are identified as words that end with "-est" – as is common practice on TheTopTens. If a descriptive word in a list title directly precedes the source word, then the list is deemed relevant. For example, the list "Top Ten Best Car Companies" would be accepted since the descriptive word "best" is describing the source word "car". Also, if there are multiple plural words in a list title, Nehovah assumes the first plural word in the title identifies the subject of the list. For example, in the list "Greatest Songs by the Cars," there are two plural words: "Songs" and "Cars." The list is determined to be about songs rather than cars since "Songs" appears before "Cars" and because the descriptive word "greatest" proceeds "Songs" rather than "Cars." Nehovah also includes lists that have the source word directly before the first plural word such as "Top Ten Car Movies", inferring that the source word is being used as a descriptor for the plural word.

Once a list is determined to be relevant, the list items also need to be processed. Because TheTopTens is composed of user-defined free-form lists, some list items are more descriptive than others. For example, the "Best Muscle Cars" list may contain items such as "1961 Ford GT Mustang From Gone in 60 Seconds." While this information is beneficial for determining why an item made the list, it is difficult to use to generate neologisms. To compensate, Nehovah parses the list items so that any words or symbols that indicate descriptive information ("from", "in", "–", ",", etc) and any words that follow are not included. Another issue with user-defined lists is the lack of quality control. To filter out obscure (and/or misspelled) words and references, Nehovah only keeps list items that are also found in Wikipedia. Any list entries that survive this level of parsing and filtering are also included in $C(w_i)$. Note that using the words from TheTopTens adds hyponyms (e.g. "Ford Mustang" for "car") rather than synonyms in some cases. We allow the use of hyponyms as the pop culture reference adds to the creativity and uniqueness of Nehovah and because it is difficult to distinguish between hyponyms and synonyms.

Blending Words

Given two concepts sets $C(w_1)$ and $C(w_2)$, Nehovah blends the words from the two concept sets to create a set of neologisms $N(C(w_1), C(w_2))$. Each word $u \in C(w_i)$ is into into a set of prefixes $P(u)$ and a set of suffixes $S(u)$. The words are split between syllables to maintain conceptual coherence and to reduce the likelihood of introducing invalid letter combinations during blending.

Unfortunately, for English it is a non-trivial task to algorithmically identify syllable boundaries because pronunciation information is not (consistently) encoded in the spelling

computational		method	
Prefixes:	Suffixes:	Prefixes:	Suffixes:
-	computational	-	method
co	mputational	me	thod
com	putational	meth	od
compu	tational	method	-
computa	tional		
computati	onal		
computatio	nal		
computation	al		
computational	-		

Table 1: Examples of how Nehovah splits words into prefix/suffix pairs by attempting to split on syllable boundaries.

of the word. For example, "io" could create two separate vowel sounds as in "li<u>io</u>n" or be a diphthong as in "moti<u>on</u>". To account for this, Nehovah conservatively splits each word u after every vowel (except the last) and between any two consecutive consonants (with exception of "sh," "th," and "ch") after the first vowel and before the last vowel. Each such split yields one prefix to be added to the set $P(u)$ and one suffix to be added to the set $S(u)$. In addition, u is also added to both $P(u)$ and $S(u)$. For example, the word "track" would be split up into the prefixes "track" and "tra" and the suffixes "ack" and "track". See Table 1 for additional examples.

Slightly abusing notation, we define the set of neologisms formed by blending two words u and v using the sets $P(u)$, $S(u)$, $P(v)$ and $S(v)$ as

$$\begin{aligned} N(u,v) =& \{yz | y \in P(u) \wedge z \in S(v) \wedge K(yz)\} \cup \\ & \{yz | y \in P(v) \wedge z \in S(u) \wedge K(yz)\} \end{aligned}$$

where $K()$ is a predicate that returns FALSE if its argument contains a letter combination not found in WordNet and TRUE otherwise.

Then, the full set of neologisms for the synonym sets $C(w_1)$ and $C(w_2)$ is generated by iterating over all pairs of words from these synonym sets:

$$N(C(w_1), C(w_2)) = \bigcup_{u \in C(w_1), v \in C(w_2)} N(u,v)$$

Scoring

Nehovah scores each neologism $n \in N(C(w_1), C(w_2))$ using four scoring criteria: word structure, concepts, uniqueness, and pop culture. Each scoring criterion can be assigned a relative weight, allowing the creation of different types of neologism.

Word Structure. The word structure score $\mathcal{W}(n)$ measures how well a neologism retains aspects of the word structure of one or both source words, as maintaining source word structure tends to produce catchier neologisms that better convey the meaning of the base words. For example, "ginormous" is a combination of "giant" and "enormous" created by replacing the first syllable from enormous with the first syllable from giant. Enough of enormous is left that the meaning is still apparent. Another example is "Linsanity," which replaces the first syllable in insanity with the single syllable word "Lin" (the last name of a professional basketball player). In this case, the overlap of "L<u>in</u>" and "<u>in</u>sanity" makes it easy to recognize the source words.

To attempt to capture this kind of desirable structure, given base words $u = y_1 z_1$ and $v = y_2 z_2$, Nehovah calculates a raw structure score for a candidate neologism $n = y_1 z_2$ as

$$S(n) = \sigma(y_1, y_2) + \pi(z_1, z_2) + B(n, u, v)$$

where $\sigma(y_1, y_2)$ is the length of the suffix common to y_1 and y_2, $\pi(z_1, z_2)$ is the length of prefix common to z_1 and z_2 and

$$B(n,u,v) = \max\{\delta(\#(n), \#(u)), \delta(\#(n), \#(v))\}$$

where $\#(x)$ returns the number of syllables in x and δ is the Kronecker delta function [$B(n,u,v)$ equals 1 if neologism n maintains the same syllable count as either base word and 0 otherwise]. $S(n)$ therefore quantifies "catchiness" by measuring base word overlap and syllable count conservation.

Given this, the word structure score $\mathcal{W}(n)$ of neologism n is the normalized raw score, with normalization taken over the set of all candidate neologisms.

$$\mathcal{W}(n) = \frac{S(n)}{\max_{\tilde{n} \in N(C(w_1), C(w_2))} S(\tilde{n})}$$

Concepts. One of the primary goals of Nehovah is to convey the concepts of the source words in the neologism. While word structure *can* aid in conveying a concept, Nehovah also explicitly measures concept clarity for a neologism by scoring how well the base concepts are communicated in its prefix and suffix.

How clearly a concept is conveyed by the prefix or suffix of a base word obtained from WordNet is measured using MoreWords[4], a tool for crossword puzzles and other word games. MoreWords uses the words from the Enable2k North American word list that is used in well-known word games. It contains 173,528 words and does not include any hyphenated words, abbreviations, acronyms, or proper nouns. Querying MoreWords with a prefix/suffix x returns the set of words W_x that have x as a prefix/suffix in MoreWords and the approximate number of times each word $\tilde{u} \in W_x$ occurs per million words ($FPM(\tilde{u})$). $FPM(\tilde{u})$ is estimated from studies on the British National Corpus[5].

Nehovah determines how apparent the concept is in a prefix/suffix by comparing the frequency of the word that the prefix/suffix is derived from with the frequencies of other words that begin/end with the same prefix/suffix. A distinctiveness score for a prefix/suffix x of base word u is calculated by first calculating a distinctiveness ratio:

$$\phi(x,u) = \frac{FPM(u)}{\sum_{\tilde{u} \in W_x} FPM(\tilde{u})}$$

[4] www.morewords.com

[5] http://www.natcorp.ox.ac.uk/

The distinctiveness score is then calculated using (an empirically determined) piecewise linear interpolation on the value of the distinctiveness ratio:

$$\chi(x,u) = \begin{cases} 1, & \text{if } \phi(x,u) \geq 0.1 \\ 0.8 + 2\phi(x,u), & \text{if } 0.01 < \phi(x,u) < 0.1 \\ 80\phi(x,u), & \text{if } 0 \leq \phi(x,u) \leq 0.01 \end{cases}$$

This score differentiates between prefixes/suffixes that do not convey the concept, that partially convey the concept, and that completely convey the concept.

Because many pop culture words are not contained in MoreWords, Nehovah measures how clearly a concept is conveyed by a pop culture base word obtained from TheTopTens as the normalized count of the number of times that a pop culture word u appears in the set of lists $L(w)$ returned from TheTopTens for a given source word w:

$$\psi(u,w) = \frac{\lambda(u, L(w))}{\max_{\tilde{u} \in T(w)} \lambda(\tilde{u}, L(w))}$$

where $\lambda(u, L(w))$ represents the number of times a base word u appears in $L(w)$, and $T(w)$ represents the set of unique pop culture words in $L(w)$.

Note that this distinctiveness score indicates the "popularity" of the concept for a pop culture reference in the neologism by comparing the prevalence of other pop culture words to the prevalence of the entire base word (rather than by considering just some prefix or suffix of the base word).

Under the assumption that these distinctiveness scores correlate with conceptual content, given a source word w, a base word $u \in C(w)$ and a prefix/suffix x of u, a concept score for the base word is computed as

$$c(x,u,w) = \begin{cases} \chi(x,u), & \text{if } u \text{ appears in WordNet} \\ \psi(u,w), & \text{otherwise} \end{cases}$$

Finally, given a concept score for both a prefix y of base word u and a suffix z of base word v, the concept score $\mathcal{C}(n)$ of the created neologism $n = yz$ is simply the average of the concept scores of the base words and their prefix/suffix:

$$\mathcal{C}(n) = \frac{c(y,u,w_1) + c(z,v,w_2)}{2}$$

Uniqueness. A score for uniqueness should place greater value on words that are not commonly used (but still convey the source concept). For example, for the source word "pants," the base word "trousers" is more common than the base word "bloomers," although both convey the same concept. Uniqueness for a base word $u \in C(w)$ is calculated using the frequency per million words score from MoreWords ($FPM(u)$) relative to all of the other synonymous words in the concept set:

$$v(u,w) = 1 - \frac{FPM(u)}{\max_{\tilde{u} \in C(w)} FPM(\tilde{u})}$$

The uniqueness score $\mathcal{U}(n)$ for a neologism n formed from the base words u and v is simply the average of their uniqueness scores:

$$\mathcal{U}(n) = \frac{v(u,w_1) + v(v,w_2)}{2}$$

Neologism	Base Words		Source Words	
Nehovah	neologism	Jehovah	neologism	creator
divinage	divine	coinage	neologism	creator
machinative	machine	creative	machine	creative
Spritependency	Sprite	dependency	soda	addiction
Pepsidiction	Pepsi	addiction	soda	addiction
pisome	pizza	awesome	awesome	pizza
pimazing	pie	amazing	awesome	pizza
iniquitivate	iniquity	cultivate	evil	school
immoralize	immorality	civilize	evil	school
coalesception	coalesce	conception	concept	blend
portmanception	portmanteau	conception	concept	blend

Table 2: A set of example neologisms generated by Nehovah with their base words and the source words that were provided to Nehovah.

Pop Culture. The pop culture score indicates if one or both of the base words are pop culture words, allowing the emphasis of pop culture references. The pop culture score $\mathcal{P}(n)$ for a neologism n created from base words u and v is given by

$$\mathcal{P}(n) = \begin{cases} 1 & \text{if } u \text{ and } v \text{ are pop culture words} \\ 0.5 & \text{if } u \text{ or } v \text{ is a pop culture word} \\ 0 & \text{otherwise} \end{cases}$$

Combining Scores

The final score for a neologism is computed as a linear combination of the four attribute scores, weighted by user-selected coefficients (cf. the sliders in Figure 2):

$$\mathcal{S}(n) = \alpha_{\mathcal{W}}\mathcal{W}(n) + \alpha_{\mathcal{C}}\mathcal{C}(n) + \alpha_{\mathcal{U}}\mathcal{U}(n) + \alpha_{\mathcal{P}}\mathcal{P}(n)$$

Evaluation of Nehovah

We now examine Nehovah in the context of the creative tripod, which consists of skill, imagination, and appreciation (Colton 2008). Skill is the ability of a system to produce something useful. Imagination is the ability of the system to search the space of possibilities and produce something novel. Appreciation is the ability of the machine to self-assess and produce something of worth. We also evaluate Nehovah with respect to its accountability–the ability of the system to explain why it generated the artifact it generated.

Skill

Nehovah demonstrates skill by generating neologisms that convey the concepts in the base words and have proper word structure. First, proposed neologisms with invalid word structure are discarded. Next, Nehovah determines if a pop culture word is valid based on its presence in Wikipedia. Wikipedia is a dynamic source that does contain neologisms (Veale 2006) and consulting Wikipedia provides a safe-guard against low quality user-supplied content in TheTopTens. Finally, only splitting the words on their syllable boundaries aids in creating word fragments that convey

meaning and are able to be blended in a way that forms a plausible word.

The skill of any system is most easily demonstrated in the artifacts that it produces. Exhibit A for the Nehovah system is its own name, which is the direct result of providing the (originally anonymous) system with the source words "neologism" and "creator." The name Nehovah is a mix of the words "neologism" and "Jehovah", and it is readily apparent that Nehovah incorporates the word "Jehovah"; another candidate neologism was "Neohovah," which conveys a bit more of the meaning of "neologism" but is not as structurally pleasing since an additional syllable is added.

Other examples of neologisms created by Nehovah are shown in Table 2. As a further demonstration, consider the following arguably coherent sentence constructed from some of the neologisms from Table 2:

> Spritependency *is a* machinative *neologism created through* portmanception *to describe someone who is addicted to Sprite.*

We also point out that the neologism "immoralize" is an actual word found in some dictionaries (it is not found in WordNet). According to the Merriam-Webster on-line dictionary, it means "to make immoral"[6] which is what is conveyed by the neologism. In other words, the system (re)invented a real word, a nice demonstration of Boden's P-creativity.

Accountability

In addition to producing a set of neologisms, Nehovah also includes the base words that were blended together to produce the neologism (see the expansion of the third neologism in the righthand pane of Figure 2). Therefore, at some level Nehovah can explain how it created a neologism. The perceived creativity of the neologisms in Table 2 is likely increased with the available explanation of which base words were blended together as well as what the source words are. For example, "portmanception" is created from the source words "concept" and "blend" using "portmanteau" and "conception" as base words. Using "portmanteau" in the place of "blend" and "conception" in the place of "concept" conveys similar meaning; revealing the connection between the base words and source words helps justify the quality and creativity of the neologism.

Imagination

A Google search for most of the generated neologisms will show that Nehovah provides novel artifacts. The hits for "Nehovah" contain references to this project and an individual's name. Most of the neologisms have no hits when searched for in Google or the hits returned are names or screen names ("divinage" is a *World of War Craft* user name).

Nehovah explores all possible combinations of prefixes and suffixes derived from the base words. Further, Nehovah also considers the synonyms for all possible senses of each base word for each possible part of speech. Using all of the possible senses for all of the parts of speech for a source word along with an ever-expanding set of free-form, user-defined (pop culture) lists can create a potentially very large search space and produce unpredictable results. For example, if "evil" and "school" are used as the source words with the intended sense of school being an "educational institution", then seeing a neologism such as "Darth_swim" would likely be somewhat unexpected (the base words of the neologism are "Darth_Vader" from the TheTopTens list "The 10 Most Evil Villains in Video Games" and "swim", a hypernym of one of the senses of the verb "school"). This, however, demonstrates the imagination of Nehovah, since it takes into consideration other and unintended senses of a source word to produce more creative neologisms. Of course, the flip side of such imaginative creations is that unintended senses can cause problems, if the main goal is to create a neologism that captures a specific sense of a source word. Thus, there is a tension between creating a rich concept set that includes all of the possible senses for a source word and generating neologisms that convey the concept of the intended sense.

Best Dog Breeds	Best Hot Dog Toppings
Pitbull	Coney Sauce
Rottweiler	Mustard
Chihuahua	Stadium Mustard
Great Dane	Relish
Miniature Pinscher	Ketchup

Table 3: The top five words returned from two lists from TheTopTens for the source word "dog", demonstrating the range of synonyms that Nehovah uses as base words.

Using the pop culture references allows Nehovah to demonstrate imagination in an unusual and contemporary fashion by using social/popular connections between words to convey meaning. Most people who are familiar with the Star Wars series would recognize the word "Darth" as having an evil connotation. As with using all the senses for a base word, some of the words from TheTopTens do not capture the intended concept of the base word. For example, consider the top five entries from two of the TheTopTens lists returned for the word dog shown in Table 3. The "Best Dog Breeds" list conveys the concept of dog to most users better than the "Best Hot Dog Toppings" list. An example set of neologisms is shown in Table 4 that shows the unintended use of the "Best Hot Dog Toppings" versus using "Best Dog Breeds" when blending the source words "robot" and "dog". Despite being irrelevant for the animal dog, these examples demonstrate the imagination of Nehovah in generating neologisms. And, in fact, the neologism "Terminaise" could be a serendipitous discovery for an exciting new condiment if the intended sense of the word"dog" was "hot dog".

Appreciation

Nehovah's appreciation is demonstrated by determining which neologisms are the "best" given a set of base words and which scoring criteria are weighted the highest. Ta-

[6] http://www.merriam-webster.com/dictionary/immoralize

	Neologism	Base Words		Score
Best 10	rottweilers: Revenge of the Fallen	rottweiler Top Ten Best Dog Breeds	Transformers: Revenge of the Fallen Top Ten Best Robot Movies of All Time	0.786
	rottweilerminator 3	rottweiler Top Ten Best Dog Breeds	Terminator 3 Top Ten Best Robot Movies of All Time	0.786
	automaton terrier	automaton	boston terrier Top Ten Best Dog Breeds	0.762
	automatian	automaton	dalmatian Top Ten Best Dog Breeds	0.755
	chihuahuaton	chihuahua Top Ten Best Dog Breeds	automaton	0.754
	automestic	automaton	domestic	0.752
	golden retrievers: Revenge of the Fallen	golden retriever Top Ten Best Dog Breeds	Transformers: Revenge of the Fallen Top Ten Best Robot Movies of All Time	0.750
	dobermansformers: Revenge of the Fallen	doberman Top Ten Worst Dog Breeds	Transformers: Revenge of the Fallen Top Ten Best Robot Movies of All Time	0.714
	doberminator 3	doberman Top Ten Worst Dog Breeds	Terminator 3 Top Ten Best Robot Movies of All Time Rise	0.714
	chihuahuanic attack	chihuahua Top Ten Best Dog Breeds	panic attack Greatest Robot Wars Robots Of All Time	0.714
Worst 10	panicpoodle	panic attack Greatest Robot Wars Robots Of All Time	poodle Top Ten Best Dog Breeds	0.143
	bulroadblock	bull terrier Top 10 Guard Dog Breeds	roadblock Greatest Robot Wars Robots Of All Time	0.143
	cheeatomic	cheese Top Ten Best Hot Dog Toppings	atomic Greatest Robot Wars Robots Of All Time	0.143
	labradorroadblock	labrador retriever Top Ten Best Dog Breeds	roadblock Greatest Robot Wars Robots Of All Time	0.143
	borderrobots	border collie Top Ten Best Dog Breeds	robots Top Ten Best Robot Movies of All Time	0.143
	bulrobots	bull terrier Top 10 Guard Dog Breeds	robots Top Ten Best Robot Movies of All Time	0.143
	borderroadblock	border collie Top Ten Best Dog Breeds	roadblock Greatest Robot Wars Robots Of All Time	0.143
	labradorrobots	labrador retriever Top Ten Best Dog Breeds	robots Top Ten Best Robot Movies of All Time	0.143
	atomustard	atomic Greatest Robot Wars Robots Of All Time	mustard Top Ten Best Hot Dog Toppings	0.143
	shetlandtornado	shetland sheepdog Top 10 Smartest Dogs	tornado Greatest Robot Wars Robots Of All Time	0.143

Table 5: Highest rated 10 and lowest rated 10 neologisms generated by Nehovah using the source words “dog” and “robot” with all scoring attributes equally weighted. The higher rated neologisms tend to flow better and convey the concepts of the base words better than the lower rated neologisms.

ble 5 shows the highest rated 10 and lowest rated 10 neologisms created using the source words “dog” and “robot” as scored with all attributes equally weighted. The source words “dog” and “robot” were chosen for this example because both source words have pop culture references and clearly demonstrate the effects of the different scoring attributes. Comparing the two sets of neologisms in Table 5, the highest rated 10 neologisms flow better and better capture the source concepts. The bottom 10 do not flow as well and this often contributes to (further) obfuscation of the source concepts. For example compare “rottweilerminator” and “cheeatomic”—the former better follows the word structure of both base words and the concepts are more clearly conveyed.

Each of Nehovah’s scoring attributes can be weighted by a user to increase or decrease its relative importance. Table 6 shows a sampling of neologisms derived from blending the source words “robot” and “dog”, when weighting is skewed completely to one of the four scoring factors. Each sub-table gives a set of neologisms weighted exclusively for the factor titled above it. For example, looking at the first sub-table (titled Pop Culture), for all neologisms, both source words are from the TheTopTens, although the word structures may be awkward and the concepts may not

Best Dog Breeds

Neologism	Base Words	
dobermaton	doberman	automaton
rottweilerminator_3	rottweiler	Terminator_3
dobermansformers	doberman	transformers

Best Hot Dog Toppings

Neologism	Base Words	
sauerminator_3	sauerkraut	Terminator_3
Terminaise	Terminator_3	mayonnaise
mustardmaton	mustard	automaton

Table 4: A set of sample neologisms for the source words "dog" and "robot" using two different lists from TheTopTens for the source word "dog".

be apparent e.g. "alasdo" from the source words "alaskan malamute" and "tornado". Neologisms in the list weighting only the Concept score tend to have prefixes and suffixes that are evocative of distinct base words, such as "bot" from the base word "robot". When Word Structure is the sole factor, the created neologisms look the most like real words, e.g., "Terman shepherd", strongly overlaps "Terminator" with "German shepherd" and preserves the number of syllables in "German shepherd." In the case of weighting solely for Uniqueness, the resulting neologisms and their base words are often quite unusual, sometime at the expense of understandability, e.g. "godiron" from "golem" and "andiron". As expected, weighting according to a single factor filters the neologisms, presenting only those that have a particular attribute, often at the expense of other factors.

Overall, we tend to favor the word structure and concepts factors for creating the best neologisms. These help to convey the concepts contained in the base words and also produce more realistic appearing words as they have valid letter sequences and are similar to the base words. While favoring the concept and word structure factors, the pop culture and unique factors can be used as a secondary bias towards certain types of base words to be blended together.

Conclusions and Future Work

In this paper, we have presented Nehovah, a system that generates neologisms from a set of user-provided source words by searching the space of synonyms and then blending two base words. We have argued for Nehovah's ability to demonstrate some necessary characteristics for creativity, including skill, imagination, appreciation and accountability.

Future work includes incorporating a learning mechanism so that users can indicate which neologisms they prefer. Nehovah could then use this information to better score the neologisms. An interesting line of future work includes generating a definition for a neologism using the base words. This would involve solving at least two difficult problems. The first problem is generating the definitions. Candidate definition components could be found by searching Wikipedia, an on-line dictionary, and/or another source for definitions for each source word. A potential definition would then be formed by blending candidate components in a way that both conveys the concept from each source word and is readable (i.e. correct grammar). The second problem is validation of the potential definition, which may be accomplished, for example, through a user study/game where Nehovah could *learn* to match definitions to neologisms based on users' votes.

Pop Culture

Neologism	Base Words	
1 labrador retrogates	labrador retriever	surrogates
1 alasdo	alaskan malamute	tornado
1 lharestorm	lhasa apso	firestorm
1 ketchupsycat	ketchup	pussycat
1 iroadblock	ibizan hound	roadblock

Concepts

Neologism	Base Words	
1 supnism	support	mechanism
1 scountomaton	scoundrel	automaton
1 domesrobot	domestic	robot
1 supbot	support	robot
1 scounrobot	scoundrel	robot

Word Structure

Neologism	Base Words	
1 pomers	pomeranian	transformers
1 automatian	automaton	dalmatian
1 Terman shepherd	Terminator 3	german shepherd
1 firestic	firestorm	domestic
1 Terman pinscher	Terminator 3	doberman pinscher

Uniqueness

Neologism	Base Words	
1 wiegolem	wiener	golem
1 gomiliaris	golem	familiaris
1 bliglem	blighter	golem
1 godiron	golem	andiron
1 gofiredog	golem	firedog

Table 6: Sample of neologisms created from the base words "dog" and "robot" using weighting schemes skewed completely toward a single factor, demonstrating Nehovah's appreciation for each scoring measure. Each set of neologisms possesses the desired attribute, often at the expense of others, e.g., the neologisms weighted for uniqueness are difficult to interpret and those weighted for pop culture have poor structure.

Acknowledgements

We would like to thank Dylan Mills from TheTopTens for providing an API for Nehovah.

References

Baker, K. 1999. Seussisms and violations to universal language constraints. In Hisagi, M., and Bradinova, M., eds., *Working Papers in Linguistics*, volume 6. George Mason University.

Boden, M. 1994. Creativity: A framework for research. *Behavioral and Brain Sciences* 17(3):558–568.

Carroll, L. 1871. *Through the Looking-Glass*. Macmillan.

Colton, S.; Valstar, M. F.; and Pantic, M. 2008. Emotionally aware automated portrait painting. In *Proceedings of the Third International Conference on Digital Interactive Media in Entertainment and Arts*, 304–311.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, 14–20. AAAI.

Cook, P., and Stevenson, S. 2010. Automatically identifying the source words of lexical blends in English. *Computational Linguistics* 36:129–149.

Cope, D. 2005. *Computer Models of Musical Creativity*. The MIT Press.

Duch, W., and Pilichowski, M. 2007. Experiments with computational creativity. *Neural Information Processing – Letters and Reviews* 11(4-6):123–133.

Fellbaum, C., ed. 1998. *WordNet: an electronic lexical database*. MIT Press.

Gibson, W. 1982. *Neuromancer*. Ace Books.

Mendes, M.; Pereira, F. C.; and Cardoso, A. 2004. Creativity in natural language: Studying lexical relations. In *Proceedings of the Workshop on Language Resources for Linguistic Creativity, 4th International Conference on Language Resources and Evaluation (LREC)*.

Morris, R.; Burton, S.; Bodily, P.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the Third International Conference on Computational Creativity*, 119–125.

Norton, D.; Heath, D.; and Ventura, D. 2011. Autonomously creating quality images. In *Proceedings of the Second International Conference on Computational Creativity*, 10–15.

Rahman, F., and Manurung, R. 2011. Multiobjective optimization for meaningful metrical poetry. In *Proceedings of the Second International Conference on Computational Creativity*, 4–9.

The United States Patent and Trademark Office. 2014. *Performance and Accountability Report, Fiscal Year 2013*. Government Printing Office.

Veale, T., and Butnariu, C. 2006. Exploring linguistic creativity via predictive lexicology. In *The Third Joint Workshop on Computational Creativity*, ECAI 2006.

Veale, T., and Hao, Y. 2007. Comprehending and generating apt metaphors: A web-driven, case-based approach to figurative language. In *Proceedings of the Association for the Advancement of Aritificial Intelligence*, 1471–1476. AAAI Press.

Veale, T.; O'Donoghue, D.; and Keane, M. T. 2000. Computation and blending. *Computational Linguistics* 11(3/4):253–281.

Veale, T. 2006. Tracking the lexical zeitgeist with WordNet and Wikipedia. In *Proceedings of the 17th European Conference on Artificial Intelligence*, 56–60.

Reading and Writing as a Creative Cycle: the Need for a Computational Model

Pablo Gervás
Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
28040 Madrid, Spain
pgervas@sip.ucm.es

Carlos León
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, Spain
cleon@fdi.ucm.es

Abstract

The field of computational narratology has produced many efforts aimed at generating narrative by computational means. In recent times, a number of such efforts have considered the task of modelling how a reader might consume the story. Whereas all these approaches are clearly different aspects of the task of generating narrative, so far the efforts to model them have occurred as separate and disjoint initiatives. There is an enormous potential for improvement if a way was found to combine results from these initiatives with one another. The present position paper provides a breakdown of the activity of creating stories into five stages that are conceptually different from a computational point of view and represent important aspects of the overall process as observed either in humans or in existing systems. These stages include a feedback loop that builds interpretations of an ongoing composition and provides feedback based on these to inform the composition process. This model provides a theoretical framework that can be employed first to understand how the various aspects of the task of generating narrative relate to one another, second to identify which of these aspects are being addressed by the different existing research efforts, and finally to point the way towards possible integrations of these aspects within progressively more complex systems.

Introduction

The field of computational narratology has been steadily growing over the recent years. There have been many effort aimed at analysing narrative in computational terms (Mani 2012), and generating narrative by computational means (Gervás 2009). With respect to computational creativity, the latter is more immediately relevant. Though it is possible to argue for a strong role for creativity in the understanding of narrative, this is less obvious than the role of creativity in the generation of narrative. This kind of argument has lead over the years to many research efforts that focus on generation of narrative to the detriment of the understanding of it. This is also supported by an argument of a different kind related to the perceived difficulty of narrative understanding from computational terms, and the lack of success of the efforts accumulated on that topic over the years. Yet it is also very clear to any seasoned reader or writer that the task of generating narrative is intrinsically bound to that of reading it. A writer writes to be read, and a writer aiming to succeed writes with the reactions of possible readers in mind. This point was originally argued in the field of narratology by authors such as Barthes (Barthes, Miller, and Howard 1975) and Ecco (Eco 1984), and in the field of automated storytelling by Paul Bailey (Bailey 1997) but it has taken a long time for the research community to act upon it. In recent times, a number of research efforts arising from an initial focus on narrative generation have started to consider the task of modelling how a reader might consume the story based on the plausible inferences that arise from a narrative discourse. From a technical perspective, these approaches are based on techniques used to obtain a plausible inference of causal and intentional relations in the discourse (Niehaus 2009; Cardona-Rivera et al. 2012; O'Neil 2013). These efforts arise from the need of generation processes to have access to some kind of feedback based on how the results of the construction process will be perceived by a potential reader. The pragmatic needs of research seem to require the implementation of at least some parts of this cycle between writing and reading that are intuitively evident to most people.

The present paper provides a breakdown of the activity of creating stories into five stages that are conceptually different from a computational point of view and represent important aspects of the overall process as observed either in humans or in existing systems. A fundamental hypothesis of the proposed breakdown is that, even though intended as a model of the composing task, it includes two additional processes concerned with modelling the task of interpretation. These processes are aimed at estimating the impression that a composition will make on an asumed interpreter, and they provide a feedback loop to improve the results of composition. This extension provides the means both for including a model of the reader in the composition process, and for explicitly representing evaluation features as part of the construction process. The proposed breakdown into five stages is analysed in terms of its relation to existing models of: creative endeavour from a computational point of view, the writing task from a cognitive perspective, and natural language generation as a set of tasks. The set of five stages is postulated as a possible model to understand how existing efforts in the field of story generation relate to one another and how future progress in the field might explore possible

interactions between them. To this end, a number of existing systems are reviewed in the light of the model.

Previous Work

The set of existing theoretical models or frameworks that may have a bearing on the task of story creation are reviewed in the following order. First, models of creative systems, then models of the writing task, and finally models of natural language generation.

Computational Models of Creativity

Wiggins (Wiggins 2006) takes up Boden's idea of creativity as search over conceptual spaces (Boden 2003) and presents a more detailed theoretical framework intended to allow detailed comparison, and hence better understanding, of systems which exhibit behaviour which would be called *creative* in humans. This framework describes an exploratory creative system in terms of a tuple of elements which include elements for defining a conceptual space as a distinct subset of the universe of possible objects, the rules that define a particular subset of that universe as a conceptual space, the rules for traversing that conceptual space, and an evaluation function for attributing value to particular points of the conceptual space reached in this manner.

The IDEA model (Colton, Charnley, and Pease 2011) assumes an (I)terative (D)evelopment-(E)xecution-(A)ppreciation cycle within which software is engineered and its behaviour is exposed to an audience. An important insight of this model is that the invention of measures of value is a fundamental part of the creative act. In the case of story generation this corresponds to developing models of reader response that can be used to provide feedback to the generation process.

Cognitive Accounts of Writing and Narrative Comprehension

Flower and Hayes (Flower and Hayes 1981) define a cognitive model of writing in terms of three basic process: planning, translating these ideas into text, and reviewing the result with a view to improving it. These three processes are said to operate interactively, guided by a monitor that activates one or the other as needed. The planning process involves generating ideas, but also setting goals that can later be taken into account by all the other processes. The translating process involves putting ideas into words, and implies dealing with the restrictions and resources presented by the language to be employed. The reviewing process involves evaluating the text produced so far and revising it in accordance to the result of the evaluation. Flower and Hayes' model is oriented towards models of communicative composition (such writing essays or functional texts), and it has little to say about narrative in particular. Nevertheless, a computational model of narrative would be better if it can be understood in terms compatible with this cognitive model.

Sharples (Sharples 1999) presents a description of writing understood as a problem-solving process where the writer is both a creative thinker and a designer of text. He provides a description of how the typical writer alternates between the simple task of exploring the conceptual space defined by a given set of constraints and the more complex task of modifying such constraints to transform the conceptual space. Apparently the human mind is incapable of addressing simultaneously these two tasks. Sharples proposes a cyclic process moving through two different phases: engagement and reflection. During the engagement phase the constraints are taken as given and the conceptual space defined by them is simply explored, progressively generating new material. During the reflection phase, the generated material is revised and constraints may be transformed as a result of this revision.

Narrative comprehension involves progressive enrichment of the mental representation of a text beyond its surface form by adding information obtained via inference, until a situation model (representation of the fragment of the world that the story is about) is constructed (van Dijk and Kintsch 1983). A very relevant reference in this field is the work of (Trabasso, vand den Broek, and Suh 1989), who postulate comprehension as the construction of a causal network by the provision by the user of causal relations between the different events of a story. This network representation determines the overall unity and coherence of the story.

Natural Language Generation

The general process of text generation takes place in several stages, during which the conceptual input is progressively refined by adding information that will shape the final text (Reiter and Dale 2000). During the initial stages the concepts and messages that will appear in the final content are decided (*content determination*) and these messages are organised into a specific order and structure (*discourse planning*), and particular ways of describing each concept where it appears in the discourse plan are selected (*referring expression generation*). This results in a version of the discourse plan where the contents, the structure of the discourse, and the level of detail of each concept are already fixed. Although the overall process includes a number of additional stages (*aggregation*, *lexicalization* and *syntactic choice* - collectively referred to as *sentence planning* -, and *surface realization*) these will not be relevant for the purpose of the present paper, which remains focused at the level of discourse.

The ICTIVS model

At its most abstract level, the task of composing a narrative must be considered in the broader context of an act of communication (see Figure 1). The communication takes place as an exchange of a linear sequence of text that encodes a large and complex set of data that correspond to a set of events that take place over a volume of space time, possibly in simultaneous manner at more than one location. To convey this complexity as a linear sequence and recover it again at the other end of the communication process requires a process of condensing it first into a message and then expanding it again into a representation as close as possible to the original. There is a *composer*, in charge of composing a linear discourse from a conceptual source that may also

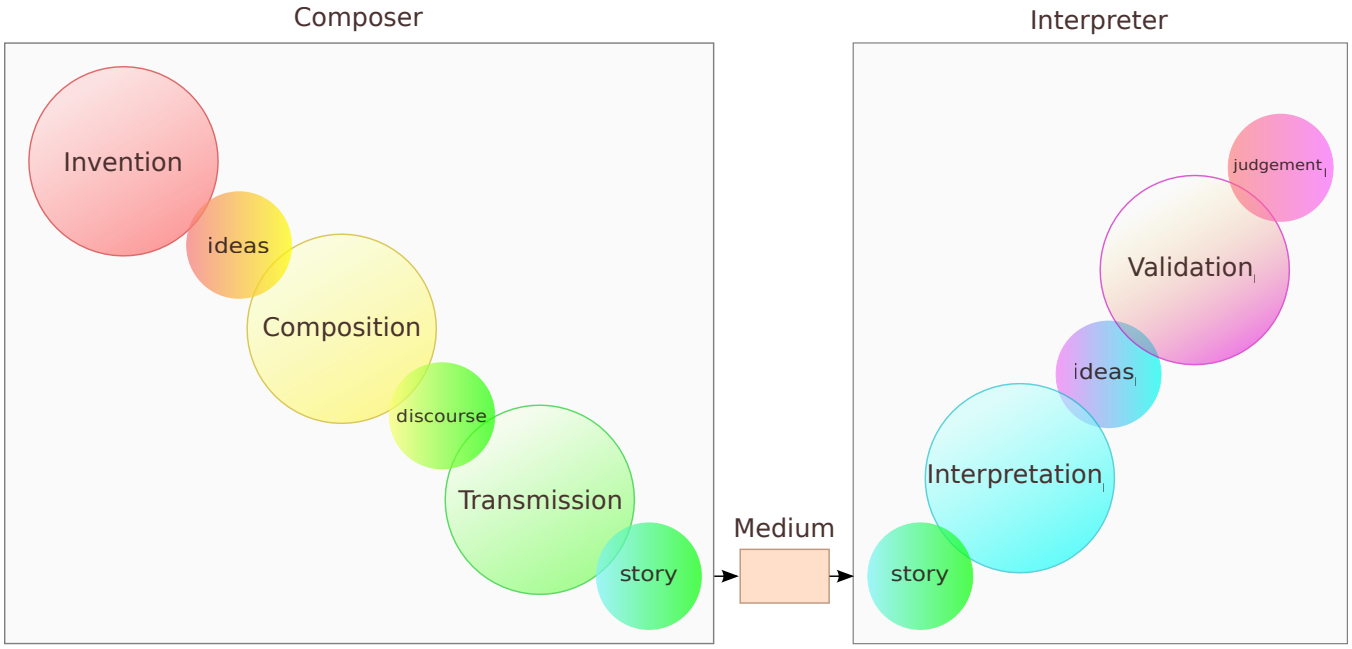


Figure 1: The traditional view of the communication process. Each big circle corresponds to an operation by one of the actors involved, whereas each small circle corresponds to the type of information conveyed from one to another. Note that $\texttt{ideas}_I$ recovered by the interpreter need not correspond faithfully to the `ideas` originally conceived by the composer.

have been produced by himself, and an *interpreter*, faced with the task of reconstructing a selected subset of the material in the conceptual source as an interpretation of the received narrative discourse. [1] The task of the composer involves four facets: the construction of the source material for the message as a conceptual representation, the selection of what subset of the conceptual source to convey, the linearization of that selection as a discourse, and the encoding of the message in a particular medium. The task of the interpreter involves a number of tasks concerned with the process of interpretation of the story into a conceptual representation, and validation of the corresponding content with respect to the criteria of the interpreter. The main hypothesis defended in this paper is that the composer also has the responsibility of ensuring that the discourse she produces is optimized to help the interpreter construct exactly the interpretation she desires to convey. To this end, the composer may need to resort to local models of the processes applied by the interpreter, used to produce copies of the conceptual interpretation and the validation that an interpreter might obtain by applying them. In consequence, the models of the interpretation process considered in this paper are not strictly concerned with the tasks carried out by the interpreter, but rather with how the outcomes of these tasks might best be modelled relying as much as possible on the resources and capabilities already available to the composer.

Based on these ideas, an abstract model for covering these aspects of narrative has been created. It has been called ICTIVS (the name stands for INVENTION, COMPOSITION, TRANSMISSION, INTERPRETATION and VALIDATION of Stories). This model divides the communicative act of narration into five stages carried out by the composer as part of an iterative cycle. Figure 2 depicts this cycle as a refinement of the traditional view of the task of the composer, now extended with an explicit representation of the task of the interpreter. This model of the interpreter provides a feedback loop on the composition process that can be used for progressive refinement of the result. The ICTIVS model does not try to solve or study *how* each process is carried out from a social or psychological point of view, it rather identifies those stages that are important from the Artificial Intelligence point of view, and those that help to model the human behaviour in narratives.

- During the INVENTION stage, the narrative content is created, based on incomplete knowledge or from scratch. Characters, narrative objectives, places and events (the *ideas*) all emerge and get related, thus creating a complex set of facts that constitute the source for the story. These facts could be understood as the log of a simulation run on the set of characters. As in real life, events produced in this way may have happened simultaneously in physically separated locations, and constitute more of a cloud than a linear sequence, a volume characterised by 4 dimensional space-time coordinates.
- The COMPOSITION stage arranges all data from the previous stage (INVENTION) and outputs a *discourse*. Composing a discourse for the source content involves drawing a number of linear pathways through the volume of space

[1] In real life, the role of the composer is usually played by a writer and the role of interpreter by a reader, but in the present case a more generic formulation has been preferred for generality.

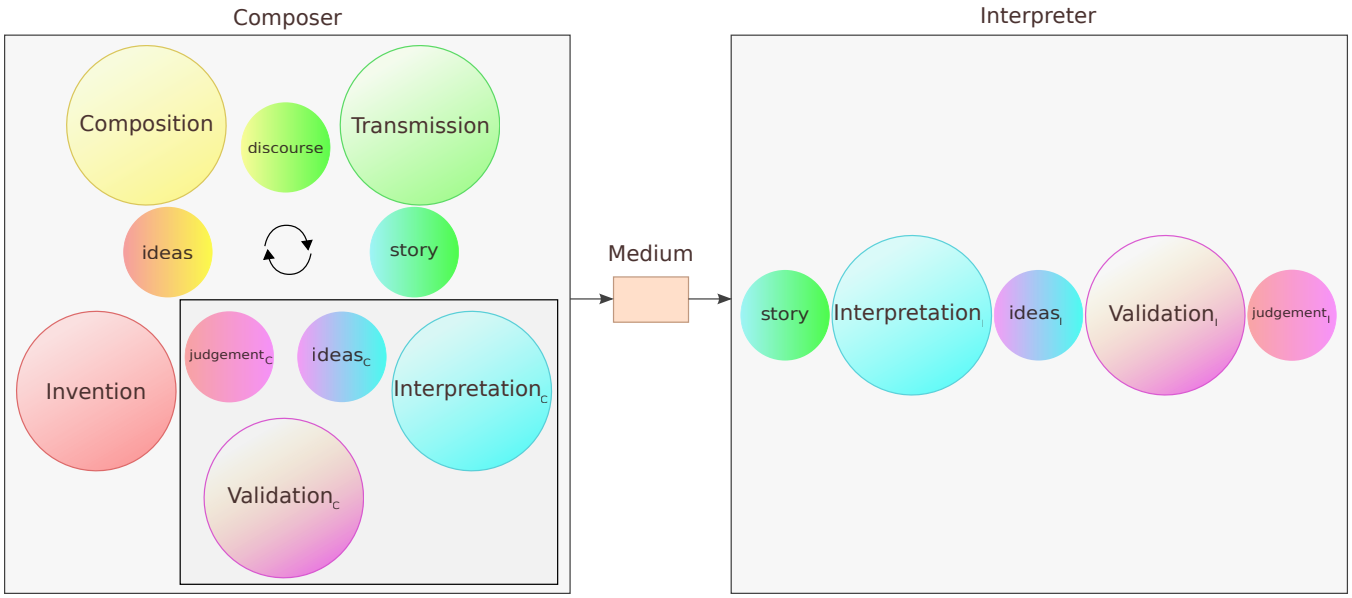


Figure 2: The ICTIVS model. It constitutes a model of the composing task. The picture includes a separate representation of the interpreter to capture two important ideas: that the proposed refinement is intended as a duplication of the interpretation task within the composer, and that the ideas ($\texttt{ideas}_C$) and the judgement ($\texttt{judgement}_C$) obtained by the composer may be different from those developed by the interpreter ($\texttt{ideas}_I$ and $\texttt{judgement}_I$, as a result of the fact that the procedures applied to obtain them are different ($\texttt{Interpretation}_C \neq \texttt{Interpretation}_I$ and $\texttt{Validation}_C \neq \texttt{Validation}_I$).

time produced by the invention stage. This type of linear pathway is sometimes referred to as a *narrative thread*. All the narrative threads deemed relevant from a given input (in truth a selection of all available ones or even a selection of fragments of the interesting parts of some of them) need to be combined together into a single linear discourse. As a result, this discourse is an ordered and filtered set of facts (properties, events, descriptions...) that are to be conveyed to the interpreter. Filtering involves considering the reader's common knowledge and inferential capabilities. Many concepts that the composer intends to convey may be omitted from the actual discourse if they can be considered to be known or obtainable via inference by the reader. It is also possible that the composer prefer to withhold particular items of information over particular stretches of the discourse, to create or enhance effects such as surprise, expectation, or suspense.

- Once a discourse has been composed, it can be rendered in a particular medium that can be consumed directly by the intended audience (whether a single interpreter or many). This stage has been called TRANSMISSION, as it involves the task of rendering the discourse in a given medium and making the medium available to the audience, but the part of the process we want to consider here is that of rendering, which involves constructive decisions and may be informed by reflection.
- The INTERPRETATION stage involves the reconstruction of the content of the message from the discourse for it. This process, when applied to a story received from an external source, constitutes the main task that an interpreter faces. Our stance in this paper is that an integral part of the task of the composer could be to apply a similar procedure to a recently composed discourse, with a view to obtaining feedback on how a hypothetical interpreter might view it. Whether from the discourse itself or from the medium produced to render it, the composer attempts to reconstruct the meaning as a user would to extract feedback on how the result of his composition task satisfies his communication goals.
- Over the reconstruction of the content of a story interpreted from a discourse, interpreters (and composers simulating the reaction of an interpreter) develop judgments on the medium, the discourse or the content of the story. This set of operations we refer to as the VALIDATION stage. As with interpretation, we consider that a composer may rely on a version of this stage to obtain feedback on how his output might be received by an interpreter.

The role of the INTERPRETATION stage is crucial even if the model is nominally restricted to the task of composition. According to the Flower and Hayes model of the writing task, linearization would occur as part of the translation subtask (converting ideas into text), followed by a number of cycles of reviewing and improving the result. The accumulated literature on modelling story generation indicates that this reviewing stage of discourse, based on an attempt at reconstructing the desired content from the discourse and a comparison between the resulting interpretation and the selected subset of the source material, is a fundamental ingredient of the broader context of the task of story generation. We therefore consider that a model of the task of story gen-

eration should include all of the five stages described to be considered complete.

One may be tempted to ascribe creativity within this model only to the INVENTION stage, on the grounds that it is there that new content is put together by combining more basic elements. However, there is also room for creativity in the COMPOSITION stage - to come up with new solutions for encoding a given content, possibly fulfilling additional goals in terms of surprise, suspense, while still meeting the communicative constraints - or the TRANSMISSION stage - to produce alternative novel and valuable renderings for a given discourse. During the INTERPRETATION stage a new instantiation of the narrative message is created. In some cases, the process of COMPOSITION reduces the content so drastically that the INTERPRETATION process requires some creative mechanisms to come up with enough material to make sense of the story. In those cases new *ideas* not considered by the writer may emerge during this stage. The resulting story is not necessarily equal to the story that the writer invented and transmitted. This point aligns very well with the observations of postmodern literary studies - arising from the work of (Barthes 1977) - along the lines that a text does not acquire its ultimate value until it has been interpreted by a particular reader, and that the role of the reader in this process must be valued in comparable terms to that of the writer. The VALIDATION process is particularly interesting in terms of creativity. In line with the insights arising from the IDEA model of Colton et al, a fundamental part of the creative act may be the invention of new measures of value. This would correspond to applying creativity at the VALIDATION stage, and it is a feature that has received little attention in the past in terms of computational creativity research. Finally, it is quite possible that creativity as perceived by external observers arise only as a result of a complex interaction between all these processes. This possibility strengthens the argument in favour of models of the composition task that captures all these aspects in a single framework.

The ICTIVS Model and Existing Related Frameworks

The ICTIVS model is compared to a number of existing frameworks for understanding related processes, of creativity, of the writing task, and of natural language generation.

ICTIVS and Models of Creativity

Processes in the INVENTION and COMPOSITION stages would correspond to what Wiggins in his framework defines as rules for traversing the conceptual space. These stages carry out the identification of new artifacts in the conceptual space of stories of the working domain. On the other hand, both the INTERPRETATION and the VALIDATION stages can be seen as ingredients in an evaluation function function in Wiggins' formalization. They both compose a process in which a story is received and judgments are formed. The TRANSMISSION stage is not explicitly addressed by Wiggins, as his model only considers the generation of creative artifacts.

Although Colton et al's IDEA model is formulated in the context of the development of creative software, its description of the process as an (I)terative (D)evelopment-(E)xecution-(A)ppreciation cycle is applicable to the task of generating a story. Under this view, INVENTION would correspond to Development, COMPOSITION and TRANSMISSION would correspond to Execution, and INTERPRETATION and VALIDATION would correspond to Appreciation.

ICTIVS and Cognitive Models of Writing

From a cognitive point of view, the set of stages that constitute the ICTIVS model aligns reasonably well with the processes described by Flower and Hayes. In terms of Flower and Hayes' model, the INVENTION stage would constitute specific operation of the planning process. The COMPOSITION stage might be considered partly within the planning process (as regards discourse planning decisions) and partly within the translating process (as regards sentence planning processes). The TRANSMISSION stage would fall directly within the translating process, including the particular "restrictions and resources presented by the language to be employed", as Flower and Hayes phrase it. The INTERPRETATION and VALIDATION stages would correspond to the reviewing process of Flower and Hayes' model. The possibility of considering different paths through the various stages of the model would correspond to enriching the model with interaction between the various processes as controlled by a monitor, which is an integral part of Flower and Hayes model.

In terms of Sharples' description of the writing task, it would be simple to say that INVENTION and COMPOSITION would correspond to the engagement phase, and that INTERPRETATION and VALIDATION would correspond to the reflection phase. However, Sharples' analysis indicates that the process of writing is far from being a simple cycle over such stages, and involves coming and going between them over a period of time, before the actual stage of TRANSMISSION is ever contemplated. In fact, it would probably be fair to say that there might be specific phases of engagement associated with INVENTION, combined with phases of reflection over whatever representation is achieved at that stage, followed by iterations of INVENTION and COMPOSITION engagements (with interspersed phases of reflection as INTERPRETATION and VALIDATION of the resulting discourse), followed by iterations of INVENTION, COMPOSITION and TRANSMISSION engagement (also combined with phases of reflection as above). Such a complex process would match the idea of heavy interaction between planning, translating and reviewing (in Flowers and Hayes terms), and should be considered corroboration of the need for a monitor module to govern how these interactions take place. This monitor would also be in charge of deciding when the final product is finally ready to be transmitted to the addressee, or generally made public.

The processes of progressive enrichment of the mental representation of a text beyond its surface form by adding information obtained via inference, as described by Van Dijk and Kintsch (van Dijk and Kintsch 1983) is the main component of the INTERPRETATION stage. This does indeed take

place when a reader attempts to comprehend a given text. However, the ICTIVS model considers this stage also to be a fundamental part of the process of creation applied by the writer. Much in the way described by Colton et al in their IDEA model, the process of creating a story is seen as an interactive cycle of production of a text (through processes of INVENTION, COMPOSITION and TRANSMISSION) followed by a process of appreciation (during INTERPRETATION and VALIDATION). The result of this appreciation process can then be fed back to the next iteration of the productive part of the cycle. Although the cycle is described in full, going all the way to the production of text before entering an appreciation phase, it is perfectly possible (and extremely plausible if considered in terms of how this task is addressed by humans) that appreciation in this sense may be applied much earlier in the cycle: for instance, once a process of INVENTION has taken place, whatever has been obtained, possibly a set of ideas represented conceptually, or a sketch of the fabula - in narratological terms - may be appreciated and the resulting information can be fed back to further processes of INVENTION. As INVENTION does not include a step of selection and encoding of information (these tasks concern the COMPOSITION stage) no stage of INTERPRETATION is required as part of this cycle, and feedback may be obtained by direct VALIDATION. A similar internal loop may occur involving COMPOSITION, with appreciation of the output of a COMPOSITION stage being submitted to appreciation even before entering a stage of TRANSMISSION. In this case, a process of INTERPRETATION may be required before VALIDATION can be applied.

Given that (Trabasso, vand den Broek, and Suh 1989) postulate the existence of a network of causal relations between the different events of a story as fundamental to determining the overall perception of its unity and coherence, it is very likely that VALIDATION of a story involve identification of an appropriate network of this nature. When VALIDATION is applied directly to the result of an INVENTION stage (fabula), it may consist simply of ensuring that such causal relations are present in the story. When applied to a narrative discourse, an intermediate stage of INTERPRETATION may be required to elicit a representation of such a network from the discourse.

ICTIVS and Natural Language Generation

At a first glance, with respect to the classic pipeline structure for natural language generation systems, the ICTIVS stage of INVENTION would correspond to the task of content determination, whereby a fabula is produced (content that may be told), with the discourse planning stage matching the COMPOSITION stage. However, there is a slight misalignment between the two models. The content determination stage of a NLG pipeline assumes all possible content to be present, and applies a selection process to establish what will be included in the communication under consideration. In contrast, the INVENTION stage is concerned with actual production of the content to be considered. In view of these, both content determination and discourse planning - as understood in NLG terms - can be considered as part of the COMPOSITION stage. In truth, all of the NLG pipeline could be considered as part of the COMPOSITION stage, with possibly only surface realization being included in the TRANSMISSION stage.

Grounding the ICTIVS Model in Existing Story Generation Systems

The applicability of the proposed model can be illustrated by using it to analyse existing efforts in story generation, with a view to recasting their apparent diversity into a homogeneous framework of understanding, and to better illustrate how they relate to the more complex aspects of narrative generation and to one another. A number of existing systems are discussed below. The selection is not meant to be exhaustive, and it has been designed to include examples of systems that cover different stages of the ICTIVS model.

MEXICA (Pérez y Pérez 1999) was a computer model designed to study the creative process in writing in terms of the cycle of engagement and reflection (Sharples 1999). It was designed to generate short stories about the MEXICAS (also wrongly known as Aztecs). *MEXICA* pioneered in the realm of automated storytellers the idea of a cycle of generation and evaluation, with the results of the evaluation being fed back to inform the generation process. In this case, the engagement cycle of *MEXICA* can be seen as a particular type of INVENTION process that directly produces a linear discourse. Over this discourse, the *MEXICA* system applies an instance of the VALIDATION stage, which is fed back into the generation process. In addition to this, *MEXICA* had a procedure for building from a set of known stories the knowledge structures called Story Contexts, which represented explicitly the emotional links and tensions between characters in the story. This process would correspond to an ICTIVS stage of INTERPRETATION. Finally, *MEXICA* provide a template-based procedure for rendering the final discourses as text. This would correspond to a stage of TRANSMISSION. There is very little in the operation of the system that might be considered an instance of COMPOSITION.

For ease of exposition, the reviewed systems are grouped into sets based on the stage that they devote most attention to.

Mostly Inventors

The *Virtual Storyteller* (Theune et al. 2003) introduces a multi-agent approach to story creation where stories are created by cooperating intelligent agents. Characters are implemented as autonomous intelligent agents that can choose their own actions informed by their internal states (including goals and emotions) and their perception of the environment. Narrative is understood to emerge from the interaction of these characters with one another. There is a specific director agent who has basic knowledge about plot structure and exercises control over agent's actions by: introducing new characters and objects, giving characters specific goals, or disallowing a character's intended action. There is also a specific narrator agent, in charge of translating the system representation of states and events into natural language sentences. In terms of the ICTIVS model, most of the operation of the *Virtual Storyteller* would correspond to a stage of

INVENTION, with very simple stages of COMPOSITION and TRANSMISSION encapsulated in the narrator agent.

Fabulist (Riedl and Young 2010) was an architecture for automated story generation and presentation. The *Fabulist* architecture split the narrative generation process into threetiers: fabula generation, discourse generation, and media representation. The fabula generation process used a planning approach to narrative generation and it would correspond to an ICTIVS stage of INVENTION. The discourse generation would correspond to an ICTIVS stage of COMPOSITION. The media representation would correspond to an ICTIVS stage of TRANSMISSION.

Inventors-Composers

MINSTREL (Turner 1992) was a computer program that told stories about King Arthur and his Knights of the Round Table. The program was started on a moral that was used as seed to build the story. Story construction in *MINSTREL* operates as a two-stage processes involving a planning stage and a problem-solving stage. At a high level of abstraction, the two processes described for MINSTREL seem to correspond to an amalgamation of the INVENTION and COMPOSITION stages.

BRUTUS (Bringsjord and Ferrucci 1999) was a program that wrote short stories about betrayal. The operation of *BRUTUS* involves three basic processes, carried out sequentially. First a thematic-frame is instantiated. Then a simulation-process is set in motion where characters attempt to achieve a set of pre-defined goals, thereby developing a plot. The process of converting the resulting plot into the final output is carried out by the application of a hierarchy of grammars (story grammars, paragraph grammars, sentence grammars) that define how the story is constructed as a sequence of paragraphs which are themselves sequences of sentences. Of these, the instantiation of the thematic frame and the simulation-process would correspond to an ICTIVS stage of INVENTION, the application of the hierarchy of grammars would blend together stages of COMPOSITION and TRANSMISSION.

Mostly Composers

There have been a number of systems developed that address the task of generating a discourse for a given set of events (León, Hassan, and Gervás 2007; Gervás 2012; Gervás 2013). These systems received as input a broad description of the set of events to consider and produce from it a conceptual representation of the discourse needed to tell them as a story. The main contributions of these systems correspond to implementations of an ICTIVS stage of COMPOSITION. Most of them include an additional stage of TRANSMISSION that renders the resulting discourses as text. In most cases these are intended for ease of evaluation, and little effort is invested in optimising the quality of the resulting texts.

In the *nn* system for interactive fiction (Montfort 2007) (now evolved into the *Curveship* system (Montfort 2009)) the user controls the main character of a story by introducing simple descriptions of what it should do, and the system responds with descriptions of the outcomes of the character's actions. Within *nn*, the Narrator module provides storytelling functionality, so that the user can ask to be "told" the story of the interaction so far. The Narrator module of *nn* was a pioneer among storytellers in that it addressed issues such: order of presentation in narrative and focalization, chronology, and appropriate treatment of tense depending on the relative ordering of speech time, reference time, and event time. In this case, the Narrator module of *nn* combines a very refined instance of a COMPOSITION stage, that deals with the issue of variation in the narrative form, and a much simpler instance of a TRANSMISSION module, which renders the resulting discourse as text.

Mostly Transmitters

STORYBOOK (Callaway and Lester 2002) produced multipage stories in the Little Red Riding Hood domain by relying on elaborate natural language generation tasks. Callaway's system is a realtime narrative prose generator that takes an instance of the presentational ordering desired for the text and an instance of the sum of the factual content that constitutes the story as input, and intelligently combines information found in the two and stylistic directives to produce narrative prose. In this sense, *STORYBOOK* can be said to be centred on the TRANSMISSION stage of the ICTIVS model. The process of devising the presentational ordering desired for the text from the sum of the factual content that constitutes the story would correspond to the COMPOSITION stage of the ICTIVS model. The task of developing the sum of the factual content that constitutes the story - not actually addressed by *STORYBOOK* - would correspond to the INVENTION stage of the ICTIVS model.

Inventors - Validators

Stella (León and Gervás 2011; León and Gervás 2012) performs story generation by traversing a conceptual space of partial world states based on narrative aspects. World states are generated as the result of non-deterministic interaction between characters and their environment. This generation is narrative agnostic, and an additional level built on top of the world evolution chooses the most promising ones in terms of their narrative features. *Stella* makes use of objective curves representing these features and selects world states whose characteristics match the ones represented by these curves. *Stella* is an example of INVENTION based on VALIDATION of internal states.

Composers-Interpreters

A significant example is the *INFER* system (Niehaus 2009), a narrative discourse generation system that employs an explicit computational model of a readers comprehension process during reading to select content from an event log with a view to creating discourses that satisfy comprehension criteria.

Mostly Interpreters

An example is *INDEXTER* (Cardona-Rivera et al. 2012), a cognitive framework which predicts the salience of previously experienced events in memory based on the current

event that the audience exposed to a narrative is experiencing. This system constitutes a model of the experience of the reader, and it involves a process of INTERPRETATION in the sense that it aims to model the online mental state of the audience which experiences the narrative. This requires progressive monitoring of the effect of each increment in the narrative on this model.

A Shortage of Validators

The VALIDATION stage of the ICTIVS model has not seen as many implementations over the years. There has been a significant research effort on the evaluation of results from story generators of various types but these consisted mostly on evaluations carried out by humans over results produced by generation systems. These efforts include: evaluating the effects of text choices on reader satisfaction (Callaway and Lester 2001), evaluating plots in terms of their acceptability and their novelty as perceived by users (Peinado and Gervás 2006), and development of specific frameworks for evaluating aspects of automatically generated narrative (Rowe et al. 2009).

Some existing systems (Pérez y Pérez 1999; Cheong 2007; Bae and Young 2008; Niehaus 2009; León and Gervás 2010) did include a specific module for validating their output as it is constructed. Of these, different systems focused on specific aspects, such as emotional tensions (Pérez y Pérez 1999), suspense (Cheong 2007), surprise (Bae and Young 2008), comprehensibility (Niehaus 2009) or conformance with a user given specification of the evolution over the story of particular parameters (León and Gervás 2012).

All these systems involve some type of cycle of construction of a candidate story (sometimes a partial draft rather than a complete one) and applying some function to validate this before continuing.

It is only in recent times that systems devoted specifically to validating properties of a narrative have been developed, such as the *DRAMATIS* model for evaluating suspense in narratives (O'Neil 2013), which includes a significant stage of interpretation to make validation possible.

Conclusions

The arguments presented in this paper suggest that the inclusion of explicit processes of interpretation and validation to inform and complement the task of constructing narratives is plausible in terms of existing models of the task in terms of human cognition. They also show how existing efforts at modelling various aspects of the story telling task have already addressed computational modelling of the various aspects that would be required to implement such inclusion. The proposed solution would achieve the integration within the computational model of the narrative construction of both a model of the reader and specific procedures for the evaluation of candidates results. This would address longstanding requirements on the storytelling task (Bailey 1997) and more recently voiced requirements on the improvement of scientific rigour in the evaluation of creative systems (Jordanous 2011).

However, it must be said that the ICTIVS model is not intended as a cognitively plausible model of the way humans deal with narratives. Instead, it is proposed as a conceptual framework that might help to understand the diversity of existing efforts in story generation, and how they relate to the more complex aspects of narrative generation and to one another. In this sense, the ICTIVS model is put forward as a rallying call for researchers in the fields of narrative modelling, story generation and computational creativity to start advancing along the difficult road of integrating together existing views and development efforts. The ICTIVS model may contribute to this task in two different ways. First, by naming and clarifying some of the subprocesses involved, it may allow future research efforts to focus on the less well explored aspects of the described cycle, which should help to enrich our overall understanding of the phenomenon. Second, by providing a simple framework for analysing existing systems in terms of a set of common elementary operations, it can help identify parts of existing systems that it might be useful to reuse in future developments or to combine with other existing ones. To this end, a conscious effort has been made to formulate the ICTIVS model at a purely conceptual level. To ensure compatibility with the broad variety of representations employed in existing systems, no detail is given of what specific representations might be considered for the data exchanged between different phases.

Progress along the lines of defining formal interfaces between the various stages is desirable in the long run, but it would require a thorough and detailed review of existing efforts in search of a consensus on possible representations for the various stages. The WHIM project, funded by the European Commission under call FP7-ICT-2013-10 with grant agreement number 611560, is a three year project that sets out to explore technologies for ideation, with a particular focus on the role that narrative generation might play in evaluating the quality of ideas. Among its objectives, it includes an effort to provide a workable specification of narrative oriented towards generation. It is envisaged that this effort will contribute to clarifying some of the details that have been glossed over in the present paper.

The effort invested so far in developing computational solutions aimed at achieving or improving computational generation of narrative has uncovered a number of different aspects to the basic phenomenon of telling a story. Whereas all these approaches are clearly different aspects of the task of generating narrative, so far the efforts to modelled them have occurred as separate and disjoint initiatives. There is an enormous potential for improvement if a way was found to combine results from these initiatives with one another. The model presented in this paper provides a theoretical framework that can be employed first to understand how these various aspects of the task of generating narrative relate to one another, second to identify which of these aspects are being addressed by the different frameworks, and finally to point the way towards possible integrations of these aspects within progressively more complex systems. Systems obtained in this way are more likely to be perceived as models of the human ability to generate stories.

A set of important insights arise from the application of the model to a selection of existing systems:

1. there are several distinct computational processes involved in the generation of a story: invention of the material to be used, composition of the material as a valuable linear discourse, transmission of this discourse using some medium
2. each one of these processes contributes some features to the final story that may be evaluated separately: on the material to be used one may evaluate coherence or originality, on the discourse issues such as comprehensibility, surprise, suspense, on the final medium grammaticality or fluency
3. some of the features arise only as an interaction between the processes and some require an intermediate process of interpretation to bring out to the fore this interaction between the underlying material and the discourse used to convey it

As a result, efforts at computational modelling must take into account the various processes, the interaction between them, and the need for a validation stage as an integral part of the process.

From the point of view of creativity, it is important to note that most existing efforts at story generation have focused on obtaining acceptable stories, with very little attention to the perceived creativity of the process. Even in cases such as (Turner 1992; Pérez y Pérez 1999) that declare an explicit interest in creativity, the actual implementation and evaluation process does not address issues that are considered fundamental in the emerging field of computational creativity, like novelty or sustained creativity. This is largely due to the inherent technical difficulties in achieving results that can be considered as acceptable stories, let alone creative ones. The creativity in story generation may arise from any of the processes involved and further creativity may arise from the interactions between them. Taking the argument above to the extreme, for story generators with an aspiration to being considered truly creative systems the validation stage must include specific solutions for measuring creativity related features beyond those that are elementary requirements of the story form.

Finally, two important ideas arise from the interaction between the proposed model and considerations on creativity. The first one is that creativity may be involved in many of the processes involved in this model, not just in that of inventing the content of a story. Composition and interpretation of stories may involve significant amounts of creativity. The creation of innovative procedures for evaluation or validation of stories may be considered a highly creative achievement. The second one is that a perception of creativity in a storytelling system may arise from the interaction between all these processes rather than be located in a particular one. This constitutes a strong argument in favour of attempting the implementation and study of models of story telling along the lines of the proposed model.

Acknowledgments

This paper has been partially supported by the projects WHIM 611560 and PROSECCO 600653 funded by the European Commission, Framework Program 7, the ICT theme, and the Future and Emerging Technologies FET program.

References

Bae, B.-C., and Young, R. M. 2008. A use of flashback and foreshadowing for surprise arousal in narrative using a plan-based approach. In *Proc. ICIDS 2008*.

Bailey, P. 1997. *A Reader-based Model of Story Generation: Ph.D. Thesis Proposal*. DAI discussion paper. Edinburgh University.

Barthes, R.; Miller, R.; and Howard, R. 1975. *S/Z: An Essay*. Farrar, Straus and Giroux.

Barthes, R. 1977. The death of the author. In *Image, Music, Text*. New York, NY, USA: Hill and Wang.

Boden, M. 2003. *Creative Mind: Myths and Mechanisms*. New York, NY, 10001: Routledge.

Bringsjord, S., and Ferrucci, D. 1999. *Artificial Intelligence and Literary Creativity: Inside the mind of Brutus, a Story-Telling Machine*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Callaway, C. B., and Lester, J. C. 2001. Evaluating the effects of natural language generation on reader satisfaction. In *Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society*, 164–169.

Callaway, C. B., and Lester, J. C. 2002. Narrative prose generation. *Artificial Intelligence* 139(2):213–252.

Cardona-Rivera, R.; Cassell, B.; Ware, S.; and Young, R. 2012. Indexter: A computational model of the event-indexing situation model for characterizing narratives. In *Proc. Workshop on Computational Models of Narrative 2012*.

Cheong, Y. 2007. *A Computational Model of Narrative Generationfor Suspense*. Ph.D. Dissertation, North Carolina State University, Rayleigh, North Carolina, USA.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The face and idea descriptive models. In *In 2nd International Conference on Computational Creativity*.

Eco, U. 1984. *The Role of the Reader: Explorations in the Semiotics of Texts*. A Midland book. Indiana University Press.

Flower, L., and Hayes, J. 1981. A cognitive process theory of writing. *College Composition and Communication* 32(4):365–387.

Gervás, P. 2009. Computational approaches to storytelling and creativity. *AI Magazine* 30(3):49–63.

Gervás, P. 2012. From the fleece of fact to narrative yarns: a computational model of narrative composition. In *Proc. Workshop on Computational Models of Narrative 2012*.

Gervás, P. 2013. Stories from games: Content and focalization selection in narrative composition. In *First Spanish Symposium on Digital Entertainment, SEED 2013*.

Jordanous, A. 2011. Evaluating evaluation: Assessing progress in computational creativity research. *Proceedings of the Second International Conference on Computational Creativity (ICCC 2011)* 102–107.

León, C., and Gervás, P. 2010. The Role of Evaluation-Driven rejection in the Successful Exploration of a Conceptual Space of Stories. *Minds and Machines* 20(4):615–634.

León, C., and Gervás, P. 2011. A top-down design methodology based on causality and chronology for developing assisted story generation systems. In *Proceedings of the 8th ACM conference on Creativity and cognition*, C&C '11, 363–364. New York, NY, USA: ACM.

León, C., and Gervás, P. 2012. Prototyping the use of plot curves to guide story generation. In *Third Workshop on Computational Models of Narrative, 2012 Language Resources and Evaluation Conference (LREC'2012)*.

León, C.; Hassan, S.; and Gervás, P. 2007. From the event log of a social simulation to narrative discourse: Content planning in story generation. In Olivier, P., and Kray, C., eds., *Conference of the Artificial and Ambient Intelligence*, 402409.

Mani, I. 2012. *Computational Modeling of Narrative*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Montfort, N. 2007. *Generating Narrative Variation in Interactive Fiction*. Ph.D. Dissertation, University of Pennsylvania, Philadelphia, PA, USA.

Montfort, N. 2009. Curveship: An interactive fiction system for interactive narrating. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, CALC '09, 55–62. Stroudsburg, PA, USA: Association for Computational Linguistics.

Niehaus, J. 2009. *Cognitive Models of Discourse Comprehension for Narrative Generation*. Ph.D. Dissertation, North Carolina State University, Rayleigh, North Carolina, USA.

O'Neil, B. 2013. *A Computational Model of Suspense for the Augmentation of Intelligent Story Generation*. Ph.D. Dissertation, Georgia Institute of Technology, Atlanta, Georgia, USA.

Peinado, F., and Gervás, P. 2006. Evaluation of automatic generation of basic stories. *New Generation Computing, Computational Paradigms and Computational Intelligence. Special issue: Computational Creativity* 24(3):289–302.

Pérez y Pérez, R. 1999. *MEXICA: A Computer Model of Creativity in Writing*. Ph.D. Dissertation, The University of Sussex.

Reiter, E., and Dale, R. 2000. *Building Natural Language Generation Systems*. Cambridge University Press.

Riedl, M., and Young, M. 2010. Narrative planning: Balancing plot and character. *J. Artif. Intell. Res. (JAIR)* 39:217–268.

Rowe, J. P.; McQuiggan, S. W.; Robison, J. L.; Marcey, D. R.; and Lester, J. C. 2009. Storyeval: An empirical evaluation framework for narrative generation. In *AAAI Spring Symposium*.

Sharples, M. 1999. *How We Write*. Routledge.

Theune, M.; Faas, E.; Nijholt, A.; and Heylen, D. 2003. The virtual storyteller: Story creation by intelligent agents. In *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference*, 204–215.

Trabasso, T.; vand den Broek, P.; and Suh, S. 1989. Logical necessity and transitivity of causal relations in stories. *Discourse Processes* 12:1–25.

Turner, S. 1992. *MINSTREL: A Computer Model of Creativity and Storytelling*. Ph.D. Dissertation, University of California at Los Angeles, Los Angeles, CA, USA.

van Dijk, T. A., and Kintsch, W. 1983. *Strategies of Discourse Comprehension*. New York: Academic Press.

Wiggins, G. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7).

Social Mexica: A computer model for social norms in narratives

Iván Guerrero Román[1], Rafael Pérez y Pérez[2]
[1]Posgrado en ciencia e ingeniería de la computación
Universidad Nacional Autónoma de México, Mexico D.F.
[2]División de Ciencias de la Comunicación y Diseño
Universidad Autónoma Metropolitana, Cuajimalpa, México D. F.
cguerreror@uxmcc2.iimas.unam.mx, rperez@correo.cua.uam.mx

Abstract

Several models for automatic storytelling represent social norms by embedding into their structures social knowledge. In contrast, this model explicitly describes computational structures to represent knowledge related to social norms, mechanisms to identify when a social norm is broken within a narrative and a set of constraints and filters to employ such social knowledge during the narrative generation process. An implementation of the model employing MEXICA, an automatic storyteller based on the Engagement-Reflection creativity model, as source of story plots is presented. Lastly, the results of a survey are presented as a preliminary evaluation of the model.

Introduction

The study of automatic storytelling has served for several purposes: e.g. to cast light on how human creativity works, to identify which cognitive processes are involved, and so on. However, studies about how social knowledge can be explicitly represented and employed during plot generation is mostly absent among the current systems.

A social norm is defined as a general expected behavior with social relevance inside a social group (Durkheim, 1982; Sherif, 1936); when the norm is broken the group sanctions the person responsible of it (e.g. social rejection).

We are interested in studying how social norms can be exploited in the context of plot generation. We have the following hypothesis:

> The rupture of a social norm allows the development of an interesting and novel narrative. Nevertheless, a system that action after action breaks social norms may produce incoherent and uninteresting narratives (Pérez y Pérez, et.al. 2011).

In this way, social knowledge is relevant for the storytelling generation process because it provides valuable information to ensure and evaluate aspects such as coherence, novelty and interestingness of a narrative. The rupture of a social norm may increase the tension of a story making it more interesting, but the abuse of this resource, may affect the coherence and overall quality of the generated narratives. When a story hero breaks a social norm, the novelty may increase; nevertheless, if this strategy is presented several times, the result may be the opposite.

Automatic storytellers, such as Daydreamer (Mueller 1990), MEXICA (Pérez y Pérez 1999), or Fabulist (Riedl 2004), includes tacit social knowledge as part of their general structures. Sometimes, this knowledge is represented as action's preconditions to prevent the inclusion of incoherent material. In other cases, this information is hardcoded. However, none of these systems detect when a social norm has been broken neither take advantage of this information during plot generation.

The purpose of this work is to provide our plot generator, MEXICA, with the capacity to employ social knowledge. Thus, we have developed mechanisms to extract social norms from inspiring stories, detecting the rupture of social norms and for taking advantage of this information during plot generation to improve the interestingness of the story in progress.

Previous work

Thespian (Si 2005), *Comme il Faut* (McCoy 2010) and Mimesis (Harrell 2012) are examples of computer models that include social knowledge into their structures. In this section, the procedures employed by each of the previously depicted systems to create narratives, and the structures employed to represent social knowledge, are briefly reviewed.

Thespian is a system to create interactive narratives in a 3D world. One of the characters, handled by a human, travels through an environment interacting with other available characters. Each character has goals to accomplish, and known facts that conform his state. To fulfill a goal, dynamic functions, which alter the state of the characters, are employed. Thespian describes a model for social norms that guides the conversation between characters. The social norms described in this model serve the purpose of conducting a conversation, thus, a social norm is broken only when the expected conversation flow is broken.

Comme il Faut is a playable computer model of social interactions that provides a set of characters with the ability to interact between them inside a virtual world. Every game starts by defining the characters (traits, basic needs, relations with other characters…) and the set of known

facts inside the virtual world. Every character additionally has a set of goals to fulfill during the game. At the beginning, all the goals are pondered, and one of them is selected to start. A social interaction is then depicted to satisfy the selected goal. Every social interaction has linked a set of possible results. Once a social interaction is performed, one of these results is selected relying on the available information of the world and the characters involved in the interaction. Finally, a new goal from one of the characters is selected and the process moves on until a predefined game goal is accomplished.

This model contemplates social norms inside their knowledge structures in the form of rules (if exists a romantic relation between characters x and y, then x can start dating y). These rules are manually defined by the model designer and its contexts are sometimes not flexible to comprise different scenarios.

Mimesis is a system for interactive narratives which explores the social phenomena of discrimination by employing games and social networks. The system provides with mechanisms to create characters based on the musical preferences of the player, which are retrieved from the information available in social networks. From this information, a set of attitudes are assigned to the character. The system further employs this information to retrieve social aggressions that are presented to the user as gestures in the character or as textual information.

Despite these systems consider the inclusion of social knowledge their approaches still invite contention because of the lack of mechanisms to determine the rupture of social norms. Additionally, mechanisms to automatically incorporate new social norms should be developed, and their constrained potential to use social knowledge during the story generation process can be improved as well.

Model description

This paper describes a computer model for representing social norms, detecting their rupture and providing guidelines during plot generation to improve the interestingness of the story in progress.

As mentioned earlier, a social norm is defined as a general expected behavior with social relevance inside a social group, and its rupture generates a sanction against the action performer.

From all the expected behaviors present inside a social group, some of them are irrelevant to the group. Breathing is an expected behavior, but has no relevance inside a narrative. On the other hand, not preserving the life of a person is relevant to a social group because it jeopardizes their welfare. In this case a social norm arises to preserve the well-being inside the group.

The concept of welfare preservation has multiple interpretations depending of the social group. Some definitions include terms such as happiness, health and prosperity, all of them terms with certain degree of subjectivity. In this work, the rupture of a social norm is delineated in terms of two premises. The first considers learning mechanisms to identify the relevant elements of scenarios where a rupture of a social norm occurs. The second is based upon the following premise:

> A social norm is broken when an action unjustifiably jeopardizes the welfare of a social group.

On grounds of previous studies of social knowledge (Echebarria 1993; Durkheim, Cosman and Cladis 2001), a mechanism to learn social procedures is based on the recognition of the elements present when an action triggers a punishment from a social group against the action performer. The set of these detected elements shapes the context where the action occurred. The first mechanism to identify the rupture of a social norm is based on the detection and representation of such contexts, called social contexts, and its further identification inside a narrative.

The second mechanism employs the concepts of welfare and justified action. To represent the welfare of a social group, the model can be configured with a set of behaviors considered as disturbances of such state. This element provides flexibility to the model and allows the user to determine when the welfare of a social group is threatened.

The concept of justified action is built upon crime and social norms theory (Nieves 2010). These theories contend that the aggressor rights loose relevance in contrast to the defender rights. Based on this idea, the following premise is stated:

> Within a story, an action that threatens the welfare of a social group is justified if, previously during the story, the action receiver had originated a welfare threaten of equal or lower intensity against the action performer.

There are different kinds of social norms employed inside narratives. Certain norms intend to preserve the cohesion inside a social group (a social norm that upholds an initiation ritual serves this purpose), others preserve different values for a group. The scope of our model of social norms is bounded to those that can be represented with a social context and that intend to preserve the welfare within a social group.

This model consists of three parts. The first, called narrative model, presents the required elements to represent a narrative. The second, called social groups' representation, introduces the basic elements to provide the system with social groups. The last, called social norms' model, com-

prises the components employed to identify, represent and employ social norms during the story generation process.

Narrative model

Our model obtains its knowledge structures from MEXICA (Pérez y Pérez and Sharples 2001; Pérez y Pérez 2007) an automatic storyteller. For this reason, this system is explained in the following section.

MEXICA

This storyteller represents the writing process as a succession of two cycles. During the first of them, called engagement, the writer focuses his efforts on producing novel related ideas guided by several constraints, and transforming them into text. On the other hand, the reflection cycle presents a retrospective stage where the agent analyses de produced material, explores feasible modifications, transforms the text, and finally, triggers new constraints that will be employed in future iterations of the process.

MEXICA employs several knowledge structures to implement this creativity model. An actions' library, an inspiring set of stories, and a group of characters and locations available in the system (see Table 1 for the list of available characters).

The actions' library serves as a repository for the basic building blocks of a story, the primitive actions. Each primitive action consists of an action name and the following sets: characters, preconditions and post conditions.

Tlatoani(T)
Prince(P)
Princess(Ps)
Priest(Pt)
Eagle and Jaguar Knights(EJ, JK)
Fisherman(Fs)
Virgin(V)
Slave(S)
Hunter(H)
Lady(L)
Enemy(E)
Trader(Tr)
Warrior(W)
Farmer(F)
Artist(A)

Table 1: Available characters in MEXICA.

The preconditions and post conditions are both samples of relations between characters. The available relations are of two types: emotional links and tensions. Emotional links represent affective reactions between characters. Each link consists of the following elements: type, valence and intensity. The type can be love or friendship between characters, the valence can be positive or negative, and the intensity is an integer number between the range [0, 3]. Tensions represent conflicts between characters, and consist of state (active -on- or inactive -off-) and type. A list with all the relations is shown in Table 2.

Emotional links	Tensions
Love	Actor dead (Ad)
Friendship	Life at risk (Lr)
	Health at risk (Hr)
	Prisioner (Pr)
	Clashing emotions (Ce)
	Love competition (Lc)
	Potential danger (Pd)

Table 2: Available relations between characters in MEXICA.

Figure 3 shows graphical representations for each type of relation between characters. An emotional friendship relation (upper left) is represented by a continuous line with the valence and intensity at the top. An emotional love relation (lower left) is represented by a dotted line with the valence and intensity at the top. A tension between two characters (right) is represented by a saw tooth linking two characters and the abbreviation of the tension type.

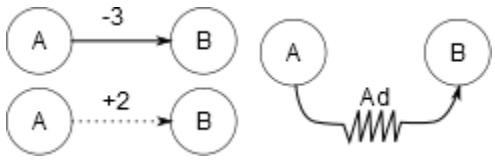


Figure 3: Graphical representation of the relations between characters.

In MEXICA, a story is presented as an ordered sequence of actions. Each story has a knowledge structure associated, called story-context, where all the known facts in the story are registered. Every time an action is performed this story-context is updated.

Another knowledge structure is the inspiring set of stories, which consists of multiple stories created by humans representing well-formed narratives. These stories are written on the same format as a regular story generated by MEXICA, as action sequences.

Each inspiring story is analyzed to create additional computer structures called contextual structures. A contextual structure is a generalization of each story-context obtained by analyzing an inspiring story. They represent a situation that happened in the analyzed story. Each structure has associated a set of actions that can be performed if a similar situation occurs in a new story.

The generalization process for a context consists in the replacement of each character with a variable. Every time a story context is generalized, the next action in the story is generalized as well, and added to the list of following actions of the generated contextual structure.

Story generation process

To create a story in MEXICA, an initial action is instantiated, and added to a new story. Each engagement step initiates by obtaining a list of feasible following actions.

For this purpose, the context of the current story is generalized and compared against each of the available contextual structures. The similar structures are then filtered by a group of constraints activated during the reflective step. Then, the first is selected, and one of its following actions is instantiated and added to the story. A new engagement step begins until the maximum number of actions is reached. If there are no remaining contextual structures after the filtering process, an impasse is declared and a reflection cycle begins.

Each reflective step initiates by determining the unsatisfied preconditions of each action in the story. When a precondition is not equivalent to a relation inside the story context, is called unsatisfied. To solve this problem, a new action with an equivalent post condition is instantiated and added to the story just before the analyzed action. When every single precondition of one action is satisfied, the next action in the story is analyzed.

A story finishes when one of the following criteria is fulfilled: all the characters in the story are dead, a declared impasse couldn't be solved, or the maximum number of actions for a story is reached.

Social groups' representation

The original version of MEXICA doesn't contemplate structures that represent social groups. Its representation is relevant to the model because they constraint the scope of a social norm, and establish relations between the characters that allows the system to identify their ruptures.

In this work, every group consists of an ordered set of hierarchies. A hierarchy is a set of characters, and has a numeric value associated, called level, which is employed to prioritize it inside a group.

Table 4 shows the basic groups inside the model. They are defined by the user in a text file, so new collections can be added to the implementation. The only constraint is to maintain at least two basic components: one for the gender structure and other for the social structure of the characters. These two groups are relevant for the system since social and gender relations are often important to determine if a social rupture occurred.

Social		
Hierarchy	Level	Characters
Nobility	5	Tlatoani, Priest
High Society	4	Prince, Princess
Fighters	3	Eagle and Jaguar knights, Warrior
Workers	2	Farmer, Fisherman, Artist, Lady, Virgin, Hunter, Trader
Low society	1	Enemy, Slave

Gender		
Hierarchy	Level	Characters
Male	2	Tlatoani, Priest, Prince, Eagle and Jaguar knights, Farmer, Fisherman, Artist, Hunter, Enemy, Slave, Trader, Warrior
Female	1	Princess, Lady, Virgin

Table 4: Social groups inside the model.

Social norms' model

In this research we employ social relations, actions and contextual structures to represent norms.

Social relations

A social relation represents the awareness of the rupture of a norm inside a story. Our system works with two types: emotions and tensions. Emotional links represent reactions between characters due to an action with social concern. Each one consists of the following elements: type, sign and intensity. The current implementation only includes one type known as social acceptance between characters; the sign can be positive or negative; and the intensity is an integer number between the range [0, 3]. Tensions represent conflicts due to a norm breakage, and they consist of state (active or inactive) and type. Table 5 displays the available relations.

Emotional links	Tensions
Social acceptance	Social disobedience (Sd) Social burden (Sb) Social threat (St) Social clashing emotions (Sce)

Table 5: Additional social relations between characters for the model.

Social actions

Social actions (s-actions) are employed to emphasize the presence of a socially relevant action inside a story. For instance, the fragment of story presented in Table 6 shows an s-action (in bold) employed to highlight that presence of a social rupture.

The hunter hated the jaguar knight. The hunter attacked the jaguar night. The jaguar knight ran away. **The jaguar knight was a coward fighter.** …

Table 6: Fragment of story presenting an s-action

When an s-action is appended to a story, it serves for adding social relations to the story context and to emphasize the rupture of a social norm; on the other hand, when s-actions are employed in an inspiring story, they serve as markers for social contexts where a rupture has occurred. These actions present evaluative clauses as part of their

associated texts. These clauses can be employed to incorporate author values and valid norms to the story text.

Each social action consists of an action name, a set of characters, a set of associated texts, a post condition, and its relations. The post condition of a social action consists of a social tension and its mode: insert, remove, or justify. The socially relevant character attribute can have one of the following values: Performer, Receiver, None, Both. The socially relevant relation attribute can have one of the following values: Gender, Social, None, Both.

The socially relevant elements of these actions are employed during the story context generalization process to represent the elements of the story context that reflect the rupture of a social norm (see the following section for a detailed description of these elements).

Action name: acted against Mexicas' will with
Character variables: A B
Post condition: insert social rejection towards A
Socially relevant character: Receiver **Socially relevant relation:** None

Table 7: Example of a social action.

Table 7 presents a social action employed to emphasize the rupture of a social norm by a character when he acts against the Mexicas' customs. The effect of this action is to attach a social rejection link against the action performer from every character aware of the rupture.

Social contextual structures

Social contextual structures, which are similar to those employed by MEXICA during the engagement phase, are built to generalize social contexts. They consist of a social context, and a reference to the social action that engendered it. Their generation process initiates by generating the context. This is obtained by generalizing the story context when a social action is found inside an inspiring story. The process consists in the replacement of each character with a variable representing it. This process is constrained by the socially relevant character attribute of the social action.

When the socially relevant character attribute of a social action is set to `Performer' or 'Receiver', that character is not generalized; if is set to `Both', none of the characters are generalized; if is set to `None', both characters are generalized. When the socially relevant relation attribute of a social action is set to `Gender' or 'Social', the distance between the hierarchies of the characters is stored.

Once the social context has been obtained, every emotional link that does not involve both of the social action's characters is removed. In the same way, every tension that does not involve any of these characters is banned. Finally, the removed tensions are retained as part of the context. Finally, the social action detected is linked to the social contextual structure.

The artist was friend of the prince. The enemy had an accident. The artist realized the enemy had an accident. The artist cured the enemy. **The artist acted against Mexica's will with the enemy.** ...

Table 8: Example of a partial story.

Table 8 presents a partial story conformed by four actions and one social action (in bold). Once these actions have been added to the plot, the story-context in Figure 9 is created. In this, the tension *Hr* (health at risk) is marked with a slash to represent that was removed from the story context with the action "The artist cured the enemy". Additionally, a social emotion (represented by alternated short and long line segments) from the prince towards the artist has been added due to the identification of a social rupture on the fourth action of the story. This rupture was originated because the artist acted against the Mexica's will by rewarding the enemy.

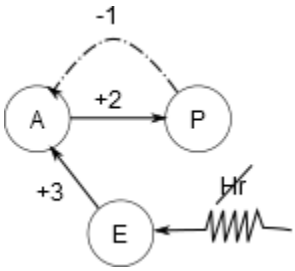


Figure 9: Story-context for the story in Table 8.

From this context, the social contextual structure in Figure 10 is obtained. Inside its context, character variables are represented by upper-case letters, and non-generalized characters are presented with the prefix 'c'.

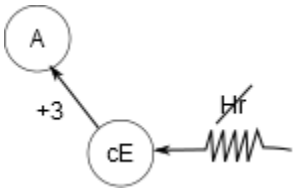


Figure 10: Social contextual structure obtained from the context in Figure 9: Story-context for the story in Table 8.

In our example, only one of the characters of the context was replaced by a variable (the artist), since the social action employed (see Table 7) marked the action receiver as a socially relevant character. Also the relations from and to the prince were banned since he was not part of the social action. This context represents a social rule condemning

positive emotional links from the enemy, even if he is in danger.

Rupture of social norms

Our model presents two mechanisms to determine when a social relation is added to the story context. The first looks up specific relations between the characters inside the story context, and, if present, a social relation is triggered. The second looks for social contexts inside the story context and appends the social relation linked it.

Regarding to the first mechanism, the social emotional link with negative valence is triggered when a character breaks a norm. This link represents social rejection. The same link with positive valence appears when a character performs an action that removes a tension from the story context. These links go from each character that identifies the rupture towards the action performer.

If several emotional links with the same valence but different intensities exist, only the one with the highest absolute intensity remains and the rest are removed. If several emotional links with different valences exist, the social clashing emotions tension is triggered, which represents ambivalent feelings towards a character.

A tension of social disobedience is triggered when a character in a lower social level breaks a social norm against another character in a higher level. A tension of social burden represents malpractices from a character in a higher social level against another character. A tension of social threat identifies a character that has broken norms several times, or has broken an intense norm. The second mechanism is explained in detail during the section related to the rupture of norms.

Mechanisms to identify social ruptures

Two processes are proposed to identify when a social norm is broken inside a story. The first is based on the hypothesis presented to identify a threat to the welfare of a social group. The second consists in the identification inside the story context of any learned social context.

Regards the first process, it is introduced into the system the tensions *Lr*, *Hr*, *Pr* and *Ad*, considered to alter the wellbeing of a social group. The first three tensions are called tensions with moderate social relevance; the last is called tension with intense social relevance.

When a tension with social relevance is unjustifiably triggered inside a story, a social norm is considered to be broken. An action that triggered a moderate or intense tension is justified when, previously in the story, at least one of these two facts stands:

- Another tension was triggered against the action performer (such as in self-defense).
- Another tension was triggered against any positively linked character to the action performer, by the action receiver (as in the case of a father defending his child).

A character is said to be positively linked to another character when, inside the story context, an emotional link with positive valence exist between them.

A justified action is exemplified by the following actions. The princess was sister of the prince. The tlatoani hated the prince and decided to attack him. The last action (the tlatoani decided to attack the prince) causes the prince's health to be at risk, which is a moderate tension. Since previously in the story, no equivalent tension had been triggered, the action breaks a social norm. If the action, the princess attacked back to the tlatoani causing his death, is added to the example, it is justified. This is because, even when it originated an intense tension (a character was death), this tension is justified by the previous action of the tlatoani and because the princess is positively linked to the prince.

The second process to identify a threat to the welfare employs the contextual structures stored. It initiates by analyzing the story context once an action is added to the story. If a social contextual structure, whose context is included inside the story context is detected, the last action in the story is marked as socially relevant. If a justifiable relation to the post condition of the social contextual structure is present inside the story context, the action is marked as justified; otherwise, the action is marked as unjustified. A relation justifies another if it is of the same type, its sign is equal, and its intensity is equal or lower.

When an action is unjustified, the post condition of the social contextual structure, which triggered such state, is instantiated with the action characters, and added to the story context. This social link emphasizes the rupture of the social norm just detected. If the action is marked as justified or normal, no additional relations are added to the context.

Relevance of social norms

A story that presents low levels of tension usually focuses on introducing relations between characters or non-relevant actions, such as location changes. These stories frequently become boring due to the lack of remarkable actions. In Table 11, an example of such type of stories generated by the model is presented.

The inclusion of tensions inside a story according to the Aristotelian tension curve gears into the generation of interesting and coherent stories. Nevertheless, some of the knowledge structures generated by MEXICA, such as contextual structures, still lack of enough information, such as social relations, which originates inconsistencies in the generated stories.

The artist went to Texcoco lake with the lady.
The virgin followed the artist.
The virgin admired and respected the artist.
The artist went to Tlatelolco market.
The lady found by accident the artist.
The artist was brother of the lady.

Table 11: Story plot with low levels of tension generated by the model.

The jaguar knight went hunting with the tlatoani.
The fisherman hated the tlatoani.
The fisherman attacked the tlatoani.
The tlatoani attacked the fisherman.
The jaguar knight made prisoner the fisherman.

Table 12: Sample story generated by the implementation of the model.

In Figure 13, the story context on the left was generated without employing the model after the third action of the story in Table 12. The story context on the right was generated employing the model on the same scenario. The contextual structure generated from the first context contains the same relations between the characters, but replacing them with the variables A and B. When this structure is employed for the generation of a new story, both characters are indistinguishable, since they have the same relations. The following action associated to the contextual structure is "C made prisoner A", but since either of the characters can be selected, in a story where this contextual structure is selected, the tlatoani can be sent to jail.

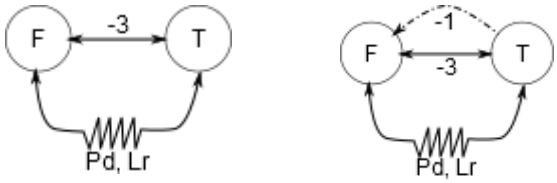


Figure 13: Story contexts from a story.

This last example states an important difference when employing the model of social norms. The problem introduced can be disentangled by the inclusion of a social relation towards the Fisherman, which was the character who broke a social norm.

The fisherman was friend of the princess.
The princess went to Texcoco lake with the fisherman.
The princess had an accident.
The artist realized that the princess had an accident.
The artist did not cure the princess.
The princess, in a sacrifice ritual, ended up with her life.

Table 14: Story with few social norms broken generated by the implementation of the model.

Testing the Model

We employed a questionnaire to cast light on how the model's implementation serves the purpose of generating more interesting narratives. For this purpose, three stories were presented to a group of forty people with at least a bachelor degree (in progress or concluded). The first story (presented in Table 11) was presented with the purpose of representing a scenario where no social norms where broken. The second story (presented in Table 14) proposes a scenario where a few social norms were broken, and the third story (presented in Table 15) provides a plot with multiple social norms broken.

The warrior had an accident.
The tlatoani realized that the warrior had an accident.
The tlatoani cured the warrior.
The virgin mugged the tlatoani.
The warrior killed the virgin.
The warrior sacrificed himself.

Table 15: Story with multiple social norms broken generated by the implementation of the model.

The first questions (see Table 17) focused on the overall evaluation of interestingness for each story. The range employed was from 1 (non-interesting) to 5 (very interesting). The average evaluations obtained for each story were the following: 2.62 for story 1, 3.35 for story 2, 3.43 for story 3. Figure 16 shows these results.

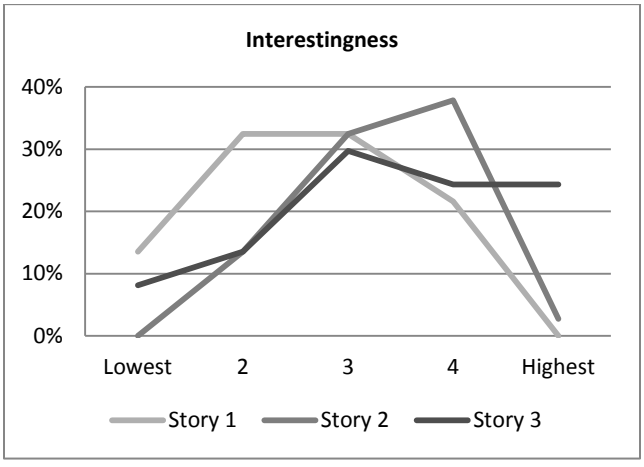


Figure 16: Results of the interestingness evaluation of the stories.

The vertical axis represents the percentage of students that selected each option displayed on the horizontal axis.

In general, how interesting was for you the first story? In general, how interesting was for you the second story? In general, how interesting was for you the third story?

Table 17: Questions for overall evaluation of interestingness.

A second group of questions (see Table 18) focused on the appreciation of social norms ruptures inside each story. Only 23% of the students identified an action that broke a social norm inside the first story, 81% identified at least one social rupture inside the second story, and 86% detected social ruptures inside the last story.

After reading the first story, which actions do you consider that break a social norm? After reading the second story, which actions do you consider that break a social norm? After reading the third story, which actions do you consider that break a social norm?

Table 18: Questions for detecting social norm ruptures.

In Figure 19, the percentages of students identifying an action breaking a social norm inside each story are presented. The vertical axis shows this percentage, and the horizontal axis represents the action number where the social rupture was detected. For the first story, no significant percentages occurred for any action. For the second story, only the last two actions presented significant results. For the third story, its last three actions were identified as representative examples of social norm breakages.

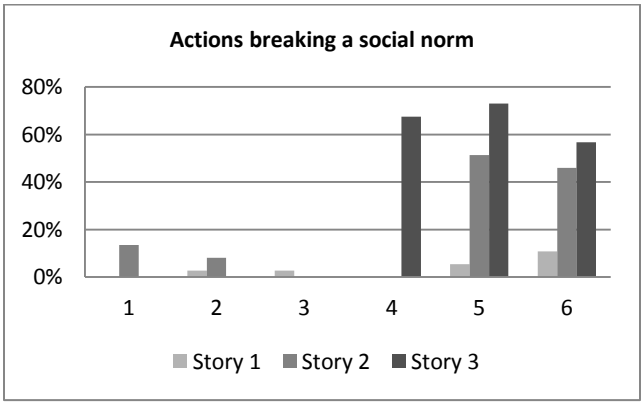


Figure 19: Percentage of students identifying a social norm rupture in an action.

Lastly, an additional question (shown in Table 20) was designed to retrieve the factors contemplated by the respondents to determine their interestingness grading. The results obtained identified that 56% of them recognized that breaking a social norm increases the interestingness of a story.

Which factors did you consider to evaluate the interestingness of a story?

Table 20: Question for determining the factors involved when evaluating the interestingness of a story.

Discussion and Conclusions

The presented values for interestingness of the stories are consistent with the social norms' hypothesis, which stated that the rupture of social norms may increase this value. Despite the fact that the overall interestingness evaluation for the last two stories is similar, the percentage of highest evaluations for the third story is significantly higher than the value obtained by the second story, which indicates that this story had the highest scores.

According to the results presented, most of the students identified the rupture of social norms in the second and third stories, which is consistent with the purpose of the questionnaire.

The implementation of our model was employed to validate our model with the actions identified by the respondents. When running the system, there were no actions identified that broke social norms for the first story; the last two actions of the second story broke a social norm because they unjustifiably introduced tensions; the last three actions of the third story broke social norms as well. These actions identified by the model are consistent with those found by the students in the survey.

We proposed a model to represent, employ and identify for social norms in narratives. To identify when a social norm is broken inside a story, two processes are proposed as part of the model. The first is based on a hypothesis presented to identify a threat to the welfare of a social group. The second consists in the identification of any generalized social context inside the story context.

The concept of unjustified actions has also been coined. When one of such actions is triggered inside a story, a social norm is considered to be broken. The procedure to identify justified actions is inspired in crime and social norm theories. An action that triggers a moderate or intense tension is considered justified when, previously in the story, another moderate or intense tension was triggered against the action performer, or against any positively linked character to the action performer, by the action receiver.

A new kind of actions, called social actions, is proposed. They emphasize the presence of a socially relevant action inside a story, and also serve as containers for evaluative clauses, which incorporate author values and valid norms within the scope of the story.

The implementation of the model has been presented to describe its operation. It introduced new computer structures to represent social knowledge and mechanisms to identify when a social norm has been broken within a narrative.

The structures described to represent the social knowledge employed by the model are social relations between characters and social contextual structures. The last structure is particularly interesting because they represent the generalization of contexts where the rupture of social norms was identified. In this way, it becomes feasible for the system to incorporate new social norms to its knowledge structures from the analysis of inspiring stories.

The results obtained from the survey as well as those retrieved from the analysis of the model of social norms seem to be aligned with the hypothesis related to the correspondence between social norms and the interestingness of a story. Additionally, when comparing the social norms detected by the model with the results from the survey, a correspondence was detected. These results suggest that the information incorporated by the model to the process of generation of narratives turns out to be valuable. Nevertheless, still additional experimentation should be performed to increase the accuracy of the model and to provide elements that can help on processes involved on the story generation and on the evaluation of the interestingness of the generated stories.

Acknowledgements

This research was sponsored by the National Council of Science and Technology in México (CONACYT), project number: 181561.

References

Durkheim, É. 1982. *Rules of sociological method*. Simon ans Schuster.

Durkheim, É.; Cosman, C.; and Cladis, M. 2001. *The Elementary Forms of Religious Life*. Number bk. 3 in Oxford world's classics. Oxford University Press.

Echebarria, A., and Gonzales, J, L. 1993. *Social knowledge,identities and social practices*. Papers on Social Representations 2.

Harrell, F., et. Al. 2012. *Exploring Social Discrimination through Interactive Narrative using Mimesis*, New Media Consortium Summer Conference.

Lajos, E. 1960. *The Art of Dramatic Writing: Its Basis in the Creative Interpretation of human motives.* New York, USA: Simon and Schuster, Inc.

McCoy, J. et.al. 2010. *Authoring game-based interactive narrative using social games and Comme il Faut*. Proceedings of the 4th International Conference & Festival of the Electronic Literature Organization: Archive & Innovate (ELO 2010), Providence, Rhode Island, USA.

Mueller, E. 1990. *Daydreaming in humans and machines: a computer model of the stream of thought.*

Ablex. Nieves, R. 2010. *Teoría del delito y práctica penal.* República Dominicana: Escuela Nacional del Ministerio Público.

Perez y Perez, R. 1999. *Mexica, A computer model of creativity in writing.* Ph.D. Dissertation, University of Sussex, England.

Pérez y Pérez, R. & Sharples, M. 2001. *MEXICA: a computer model of a cognitive account of creative writing*. Journal of Experimental and Theoretical Artificial Intelligence. Volume 13, number 2, pp. 119-139.

Pérez y Pérez, R. 2007. *Employing Emotions to Drive Plot Generation in a Computer-Based Storyteller*. Cognitive Systems Research. Vol. 8, number 2, pp. 89-109.

Perez y Perez, R, et.al. 2011. *Mexica-impro: ideas para desarrollar un modelo computacional de improvisación*. CIENCIA ergo sum, Vol. 18, Número 1.

Pérez y Pérez, R., and Ortiz, O. 2013. *A model for evaluating interestingness in a computer–generated plot.* In Proceedings of the Fourth International Conference on Computational Creativity, 131–138.

Riedl, M. O. 2004. *Narrative Generation: Balancing Plot and Character*. Ph.D. Dissertation, North Carlolina, USA.

Sherif, M. 1936. *The psychology of social norms*. Harper and Brothers.

Si, M.; Marsella, S. C.; and Pynadath, D. V. 2005. *Thespian: An architecture for interactive pedagogical drama.* Proceedings of the 2005 conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology 125.

Turner, S. R. 1993. *Minstrel: A Computer Model of Creativity and Storytelling*. Ph.D. Dissertation, Los Angeles, CA, USA. UMI Order no. GAX93-19933.

Creativity in Story Generation From the Ground Up: Non-deterministic Simulation driven by Narrative

Carlos León
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, Spain
cleon@fdi.ucm.es

Pablo Gervás
Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
28040 Madrid, Spain
pgervas@sip.ucm.es

Abstract

Creativity in narrative requires careful management of knowledge but story generation systems focusing on creativity have typically circumvented this level of detail by using high level descriptions of events and relations. While this has proven effective for plot generation, narrative generation can be drastically enriched with a grounded representation of actions based on low level simulation. This level of detail and robust knowledge representation can form the basis for a conceptual space exploration driven by narrative knowledge, namely by guiding non-deterministic generation of successive simulation states composing a story. This paper presents and updated version of the story generation system *STellA* that implements this hybrid model, along with results and discussion on the relative benefits of the described approach.

Introduction

Instances of story generation systems usually perform at a relatively abstract level, focusing on the plot and aggregating details that, if processed at a lower granularity level, could enrich a story to the point that these details themselves could potentially be the sources for new narrative constructions and unexpected plot twists (Turner 1992; Pérez y Pérez 1999; Riedl and Young 2010). This lack of fine grain detail is usually due to the technical restrictions that the currently available knowledge representation models impose over the design of complete story generation systems. Classic knowledge representation methods have proven to set the same limits on the implementation of this kind of systems as on many other applications like expert systems (Bell 1985) or ontologies (Rosati 2007), to name a few.

Lower level world-modelling techniques, like simulation, have different features than relation-based knowledge representation. In this context, we consider simulation as a process in which the whole world is modelled in a complete structure evolves step by step according to a certain, fully defined set of rules. This definition is broad enough to contain a number of different approaches to knowledge representation in general and plot generation in particular. Simulation-based modelling, as one of these techniques, can provide a good way to represent the needed information for story generation while relatively different from logic-based approaches. Indeed, simulation has been used to model narrative generation, but it has not been widely used to create explicit models of creativity in narrative. This is probably because the most evident use of simulation is the reproduction of the evolution of a static model in order to examine some results, which seemingly contradicts the need for unpredictability, novelty and freedom usually assumed to play a fundamental role in creativity.

The relatively reduced number of systems that use simulation to model creative processes contrasts with the undeniable success of simulation for gathering results and producing data from grounded models. When seen in the appropriate light, simulation becomes a powerful tool for generating a big amount of artifacts, but only if the generative process is able to complement the robust generation of simulation-produced data with techniques that let the generation produce and explore a conceptual space. In fact, simulation has been applied to story generation in several systems, but these have not put the focus on creative generation (Meehan 1977; Theune et al. 2003; Aylett et al. 2005).

This context suggests that enhancing a process grounded in simulation with already available models used in Computational Creativity is a promising method for producing grounded data and at the same time explore a conceptual space. In particular, creative processes heavily influenced by knowledge representation and management, as story generation, can benefit from the features that both fields offer. Generating a story is a complex process where details can make a huge difference and simulation can provide this level detail when used against a proper model. Together with this granularity, explicit means for traversing a conceptual space trying to generate a story with certain properties can provide a useful pattern for story generation.

This hybrid system mixing simulation and creative exploration for story generation is described along this paper. The current system description is an updated version of the story generation system *STellA* (*Story Telling Algorithm*) (León and Gervás 2011) that mixes a non-constrained simulation-based production of world states and narrative actions as source material for a conceptual space exploration engine. The system controls and chooses simulations in a non-deterministically generated space of partial stories until the generation finds a satisfactory progression of simulations

that are rendered as a story.

The previous design of *STellA* did not include a world simulation as a generative solution. Instead, knowledge was represented by means of logic facts and a elaborated set of domain rules. While this approach was carefully structured to permit incremental knowledge inclusion, the engineering effort for modelling the world became too big. We identified that an even more structured representation (a well defined structure resembling the world model used in simulations) could alleviate the required engineering effort. This paper thus describes the modification of the main generation engine to allow for a simulation-based knowledge representation and world evolution. This includes the design of a new representation system and the creation of a narrative-driven conceptual space exploration based on rules (objectives and constraints) and narrative curves. The previous version of *STellA* included curves and rules, but the way in which they were used was fundamentally different.

Related Approaches to Automatic Story Generation

In order to avoid ambiguity, we will restrict our analysis here to three levels of conceptual representation of a story, and refer to these as the *fabula* (the complete set of what could be told, organised in chronological order of occurrence), the *discourse* (what has been chosen to tell, organised in the order in which it is to be told) and the *narrative* (the actual way of telling it). Of all existing effort to build plots, the present review will be focusing on those that construct a fabula by means of a process of simulating the actions of a set of characters.

The first story telling system for which there is a record is the Novel Writer system developed by Sheldon Klein (Klein et al. 1973). Novel Writer created murder stories within the context of a weekend party. It relied on a micro-simulation model where the behaviour of individual characters and events were governed by probabilistic rules that progressively changed the state of the simulated world (represented as a semantic network). The flow of the narrative arises from reports on the changing state of the world model. A description of the world in which the story was to take place was provided as input. The particular murderer and victim depended on the character traits specified as input (with an additional random ingredient). The motives arise as a function of the events during the course of the story. The set of rules is highly constraining, and allows for the construction of only one very specific type of story.

Overall, Novel Writer operated on a very restricted setting (murder mystery at weekend party, established in the initial specification of the initial state of the network), with no automated character creation (character traits were specified as input). The world representation allows for reasonably wide modeling of relations between characters. Causality is used by the system to drive the creation of the story (motives arise from events and lead to a murder, for instance) but not represented explicitly (it is only implicit in the rules of the system). Personality characteristics are explicitly represented but marked as "not to be described in output". This suggests that there is a process of selection of what to mention and what to omit, but the model of how to do this is hard-wired in the code.

TALESPIN (Meehan 1977), a system which told stories about the lives of simple woodland creatures, was based on planning: to create a story, a character is given a goal, and then the plan is developed to solve the goal. TALESPIN introduces character goals as triggers for action. Actions are no longer set off directly by satisfaction of their conditions, an initial goal is set, which is decomposed into subgoals and events. The systems allows the possibility of having more than one problem-solving character in the story (and it introduced separate goal lists for each of them). The validity of a story is established in terms of: existence of a problem, degree of difficulty in solving the problem, and nature or level of problem solved.

Lebowitz's UNIVERSE (Lebowitz 1985) modelled the generation of scripts for a succession of TV soap opera episodes (a large cast of characters play out multiple, simultaneous, overlapping stories that never end). UNIVERSE is the first storytelling system to devote special attention to the creation of characters. Complex data structures are presented to represent characters, and a simple algorithm is proposed to fill these in partly in an automatic way. But the bulk of characterization is left for the user to do by hand.

UNIVERSE is aimed at exploring extended story generation, a continuing serial rather than a story with a beginning and an end. It is in a first instance intended as a writer's aid, with additional hopes to later develop it into an autonomous storyteller. UNIVERSE first addresses a question of procedure in making up a story over a fictional world: whether the world should be built first and then a plot to take place in it, or whether the plot should drive the construction of the world, with characters, locations and objects being created as needed. Lebowitz declares himself in favour of the first option, which is why UNIVERSE includes facilities for creating characters independently of plot, in contrast to Dehn (Dehn 1981) who favoured the second in her AUTHOR program (which was intended to simulate the author's mind as she makes up a story).

The actual story generation process of UNIVERSE (Lebowitz 1985) uses plan-like units (plot fragments) to generate plot outlines. Treatment of dialogue and low-level text generation are explicitly postponed to some later stage. Plot fragments provide narrative methods that achieve goals, but the goals considered here are not character goals, but author goals. This is intended to allow the system to lead characters into undertaking actions that they would not have chosen to do as independent agents (to make the story interesting, usually by giving rise to melodramatic conflicts). The system keeps a precedence graph that records how the various pending author goals and plot fragments relate to each other and to events that have been told already. To plan the next stage of the plot, a goal with no missing preconditions is selected and expanded. Search is not depth first, so that the system may switch from expanding goals related with one branch of the story to expanding goals for a totally different one. When selecting plot fragments or characters to use in expansion, priority is

given to those that achieve extra goals from among those pending.

The line of work initiated by TALESPIN, based on modelling the behaviour of characters, has led to a specific branch of storytellers. Characters are implemented as autonomous intelligent agents that can choose their own actions informed by their internal states (including goals and emotions) and their perception of the environment. Narrative is understood to emerge from the interaction of these characters with one another. While this guarantees coherent plots, Dehn pointed out that lack of author goals does not necessarily produce very interesting stories. However, it has been found very useful in the context of virtual environments, where the introduction of such agents injects a measure of narrative to an interactive setting.

The Virtual Storyteller (Theune et al. 2003) introduces a multi-agent approach to story creation where a specific director agent is introduced to look after plot. Each agent has its own knowledge base (representing what it knows about the world) and rules to govern its behaviour. In particular, the director agent has basic knowledge about plot structure (that it must have a beginning, a middle, and a happy end) and exercises control over agent's actions in one of three ways: environmental (introduce new characters and object), motivational (giving characters specific goals), and proscriptive (disallowing a character's intended action). The director has no prescriptive control (it cannot force characters to perform specific actions). Theune et al. report the use of rules to measure issues such as surprise and "impressiveness".

In general, approaches to Interactive Storytelling have some degree of simulation as conceived in this work (Aylett et al. 2005; Cavazza, Charles, and Mead 2002; Mateas and Stern 2005). While every approach models the problem of story generation in a specific way, there exist some degree of similarity in the way they perform, namely by chaining sequential states that are driven or selected by an implicit or explicit model of plot quality.

Knowledge Representation in the Story Generation System: Simulation

Narratives are known to share a relatively high amount of constructions and the complexity of common sense knowledge (Schank and Abelson 1977). Elaborated narratives are as complex as common human knowledge and thus its representation and processing is a long term problem of Artificial Intelligence. As an example, we can borrow a famous scene from *The Hobbit* (Tolkien 1972) in which Bilbo Baggins, when trying to win the game of riddles against Gollum, asks himself "What have I got in my pocket?". While the scene can seem not very complex for human cognition, this seemingly simple event carries a huge amount of information that requires a fine grain representation of characters (property, clothes, value of items), intentions (trying to escape), self-awareness (asking something to himself), emotions (fear), focus and concentration of characters (focusing on something relatively independent from the current context) and many other aspects that confer relative narrative quality and richness.

The complexity becomes a problem when trying to represent knowledge by classic means. Logic-based knowledge representations methods have been designed from the early years of Artificial Intelligence and, after the initial optimism (revived with the arrival of expert systems) the complexity of such systems became clear to the point that it is widely accepted that knowledge intensive systems are limited and their use is restricted only to very well known domains (Bell 1985). Many different kinds of formalisms for knowledge representation have appeared along the last years (Trentelman 2009; Sloman 1985), but the basic problems of knowledge representation are still present and relatively unsolved (Sowa 2000; Baral 2003).

Logic-based knowledge representations for story generation has nonetheless been used in several story generation system, but with very restricted domains (Pérez y Pérez 1999; Bringsjord and Ferrucci 1999). This has classically lead to systems that perform well in their respective merits and contributions, but a big amount of rich stories has not been produced so far. In order to partially tackle this issue, the presented version of *STellA* follows the hypothesis that grounding knowledge representation as much as possible is determinant for allowing a story generation system to produce rich content. A rich representation complemented by conceptual space exploration guided by narrative are proposed as a solution for creative story generation. According to this hypothesis, making the simulation more complex could provide more complex worlds and interactions and therefore create a larger conceptual space traversable by the narrative-based driving engine. The system will hypothetically be able to generate many different stories and partially identify which ones are "better" according to a set of given objectives.

Grounding Knowledge for Storytelling

For the simulation engine to be able to produce states containing content suitable for narrative generation, an appropriate grounded representation and a corresponding set of rules for creating that information are needed. This is a new addition to *STellA*.

Grounding knowledge representation for story generation requires a low level definition of concepts that are usually defined in a more abstract way by most other generation systems (Turner 1992; Pérez y Pérez 1999; Bringsjord and Ferrucci 1999). This results in an additional effort from the beginning since usual constructions inherited from logics as $in(knight, room)$ must be refined so as to represent data better suited for simulation. In the previous example, in order to represent exact position, the data would have to become $position_{knight} = (10, 20)$, assuming that $(10, 20)$ is a valid coordinate inside the $room$. This is the kind of knowledge representation that the proposed system uses.

This approach requires a fixed representation in which every construction or relation is *grounded* in the sense that the system includes mechanisms to process that construction internally. This grounding permits meta-representation of the world, which means that a mental state of the world, for instance, can be represented using the same formalism.

This meta-representation *STellA* is provided with makes

knowledge representation possible at two different levels: first, characters' reasoning uses a set of rules that manage incomplete knowledge (characters can ignore aspects of their surrounding context). Then, the same set of rules is applied to the simulated world, in which there is no uncertain information since the whole state is available. This implies a relative reduced engineering effort compared with the maintenance of two different rule sets.

Domain rules are a determinant part in this model. Narrative generation is a knowledge hungry process and any domain model is by definition incomplete (given the requirements of narrative this would imply modelling all human knowledge). This makes it almost impossible to recreate the needed amount of information in a single prototype, thus imposing the need to design a flexible, improvable system to let it evolve over time and manage a richer set of knowledge constructions.

In order to keep the rule set maintainable, rule coupling has been reduced to a minimum in terms of the structure of the rule set. Rules are organized in a linear way, meaning that no hierarchical topology is imposed over the design. This lets the maintainer include new rules without taking a big structure into account. Additionally, rules can be enabled or disabled at will without affecting the rest of the system since no rule is dependent on any other by design. The semantic coupling between rules still exist, but this is kept to a minimum.

For this independence of rules to be possible, a domain-specific language for rules has been included as part of the generation engine. The rules can query the world state and output *actions* that represent changes in the story, as the next section explains. Querying current state limits the scope in which rules can act, which constraints rule creation and makes them easier to produce. Rules cannot examine the story but only the current simulation. In this way, narrative processes are isolated.

STellA offers a set of primitives for querying the current story so that the creation of these rules can be made without knowing the representation details. Figure 1 shows an example of objective rule for creating the story and the use of the story-querying primitives that the current version of the system provides the user with.

$$\begin{aligned} \texttt{finished}(story) \leftarrow\ & hs = \texttt{humans}(story) \\ & hs_d = \texttt{inDungeon}(story, hs) \\ & \texttt{length}(hs_d) == 0 \end{aligned}$$

Figure 1: Example of objective rule for the story generation process. A story must satisfy this rule to be valid.

Rules are able to cope with *incomplete knowledge* in the generation system, which is also a new addition in this updated version of *STellA*. The meta-representation of the world that characters have can be incomplete, and thus some properties of the internal representations can have the $uncertain$ value. When characters reason to decide their next action, use a simple unification mechanism to instance the $uncertain$ value with potentially valid grounded values. For instance, a character ignoring whether an enemy is equipped with a weapon searches over the possibilities and acts according the first plausible solution. More powerful inferencing techniques will be used in future versions.

Non-deterministic generation of Narrative Actions

If the described simulation process generates only one single sequence of actions and corresponding states, the room for creativity would be marginal. According to most frameworks of computational and non-computational creativity, the creation or exploration of a conceptual space, trying to produce unexpected and valuable artifacts is a determinant part of the creative process (Boden 1999; 2003).

This update of *STellA* performs the exploration of the corresponding conceptual space generatively, that is, iteratively creating new states for subsequent simulation. This has been modeled and implemented as a non-deterministic process in which a certain simulation step can yield not one but many steps. From a classical Artificial Intelligence perspective, the conceptual space generated by *STellA* is a tree rooted in the original state (the base state from which the generation happens). Each intermediate node of the conceptual tree contains a partial simulation state that, when processed, generates possibly many candidate states that can be subsequently expanded, in this way modelling non-determinism.

While state exploration works for expanding the conceptual space, connecting the simulation with the creation of a narrative structure requires a more detailed process. The grounded data coming from each generation step must be processed carefully because the state changes that a simulation step yields are heterogeneous from a narrative perspective.

The changes happening from a simulation state to the next one that are produced in the non-deterministic expansions are referred to as *narrative actions*, which are a new addition to *STellA*. During the development of the described system the number of these actions has grown as more different kinds were detected. It is important to note that the way in which simulation is implemented in *STellA* affects the kind of actions that are produced and thus its identification, but the next list is likely to be applicable to other approaches as well:

- *Character perception* actions define the parts of the simulation that are perceived by the characters. This includes perceiving the surrounding objects, being aware of health and position, updating or forgetting the position of an object that has moved and so on. The generation of these actions are currently model as a non-deterministic process in which perceptions have a probability to happen. The algorithm then orders perceptions by probability, creating sets of perceived elements non-deterministically. Perception actions are the link between the complete world happening in the simulation and the inner representation of it that every agent in the story (every character) has.

- *Deus ex* actions are generated without any causal requirement. They must be consistent with the current state, but do not need to respond to any character need of model. Deus ex actions model events that are too serendipitous to need a detailed model, like a character stumbling upon a rock when running or raining. These actions are generated non-deterministically and have a probability of happening in their definition that is used by the generator to order these actions by their chance of occurring and not by pure randomness. This has been designed so to keep a complete model not depending on random numbers.
- *Character desires* actions are the output of a reasoning process that emulates character decisions. These decisions include eating if the character is hungry, trying to escape an enemy or maybe attacking him or her. These actions confer a relative degree of believability (Riedl 2004). Character desires actions, which are generated in a non-deterministic way, have both an associated probability and a *priority*. This priority is used by the characters in the next step of simulation to order desires and try to satisfy the most prioritized ones first.
- *Character intentions* complete *desires* and *perception* so as to reproduce a classic agent-like narrative model (Bratman 1987). Intentions are generated according to perceptions (beliefs in the classic model) and desires, which means that the representation of the external world in not taken into account when creating intentions (only the character's internal representation). This allows for a simpler creation of rules since less information must be taken into account. Character intentions actions are non-deterministic too and have an associate probability just like the other kinds of actions. Trying to go to some location that the character desires to be in or trying to attack the enemy that the character desires to be dead are examples of intentions. The difference between *doing* and *trying to do* is subtle but very influential in narrative generation since it permits richer character interaction.
- *Physical world* actions are non-deterministic and model causality of physical events that, under certain conditions, will necessarily happen with a certain probability. Things that fall to the ground if nothing holds them or moving an object if it is pushed with enough force are examples of physical world actions. This kind of actions have the additional role of representing *success* of *failure* of character intentions. In this way, a character can try an action and the physical state will decide whether the intention succeeded or not.

This division makes sense from the point of view of story generation. The focus and detail on character behavior is clear and considered to be very important in narrative. This is complemented with serendipitous events and world physics in a broad sense. Probabilities are used to order actions in such a way that the main algorithm produces candidate updated versions of the current state of the simulation and gives priority to the most likely ones. Creativity can be explored by choosing less likely states, which is planned as part of the future enhancements of *STellA*.

These five kinds of narrative actions are extracted from the simulation. Formally speaking, the output of each step of the simulation non-deterministically yields a set of new states along with their corresponding actions. This can be formally described as:

$$\langle state, e, p, d, i, w \rangle$$

where $state$ is the current state of the simulation, e is the set of *deus ex* actions generated from that step, p is the set of *character perception* actions, d is the set of *character desires* actions, i is the set of *character intentions* actions and w is the set of *physical world* actions.

A *fabula* generated by *STellA* is then a list of tuples:

$$[\langle state, e, p, d, i, w \rangle]$$

The generation can be represented formally in terms of a generative function γ that accepts a $state$ and returns a non-deterministic set of tuples:

$$\begin{aligned}\gamma(state) = \{&\langle state_0, e_0, p_0, d_0, i_0, w_0 \rangle, \\ &\langle state_1, e_1, p_1, d_1, i_1, w_1 \rangle, \\ &\dots \\ &\langle state_n, e_n, p_n, d_n, i_n, w_n \rangle\}\end{aligned}$$

Having explained and formalize how to generate a conceptual space of stories from a grounded simulation, it is still necessary to complete the system by including a way to traverse this space an find valuable artifacts, namely valid stories.

Narrative Drives the Simulation: Curves, Objectives and Constraints

Simulation is a flexible and powerful tool for representing the state of a story and the transitions between states. However, producing a sequence of states that, when appropriately rendered, are acceptable as a narrative, requires control over the generation. *STellA* uses three types of mechanisms to drive the simulation: *objectives*, *constraints* and *narrative curves*.

The presented generation process is fed with a set of *objectives* that the story must satisfy in order to be suitable to be accepted as finished and valuable by the system. This version of the story generation system models objectives as a group of boolean functions receiving a story. The user can thus use these to create declarative definitions of the kind of wanted story. *Objectives* are used post-hoc. When a partial story is reached by the system, it is checked against the set of these objectives and all of them must accept the story as valid. Figure 1 shows an example.

Along with *objectives*, providing the system with means to restrict the generation is needed. Non-determinism in story generation is a powerful modelling tool, but unrestricted production of stories degenerates in a very big conceptual space whose whole traversal is intractable (León and Gervás 2010). This is not only consistent from a computational perspective but also from the point of view of creativity in story generation: the set of stories that can be generated from any starting state is very large.

This characteristic is inherent to the domain of story production and cannot be eluded. The computational generation can, however, filter out those intermediate states that are not promising and should not be explored, as humans seemingly do (Sharples 1999). The current model uses *constraints* for avoiding exploring branches of the traversal process that are unpromising. The implementation of constraints is analogous to the implementation of objectives as constraints are defined in terms of declarative rules using the same kind of formalism and query primitives. Constraints, however, are used in the generation during the expansion of new states to be simulated and forbid the exploration of those candidates states that do not satisfy them.

The use of constraints compared to objectives therefore leads to a less strict definition. In practical terms, constraints are usually less restrictive with regard to their scope: experience suggests that constraints are defined in term of specific features that a story should not have, while objectives tend to describe general aspects of a narration. Figure 3, showing an example of a constraint, exemplifies this.

$$
\begin{aligned}
\texttt{promising}(story) \leftarrow\ & hs = \texttt{humans}(story) \\
& \forall h_i \in hs : \\
& \quad \forall h_d \in hs - \{h_i\} : \\
& \quad\quad d_i = \texttt{distance}(story, h_i, h_d) \\
& av = \texttt{average}(d_0, d_1, \ldots, d_n) \\
& \texttt{av} <= threshold
\end{aligned}
$$

Figure 2: Example of constraint rule. A partial story not satisfying a constraint rule will not be accepted as promising and its corresponding state will not be explored.

STellA uses a generalized version of *tension curves* to drive story generation. The design of these curves as a way to drive plot generation has been studied in previous versions of *STellA* (León and Gervás 2011; León and Gervás 2012). The main objective underlying this method is to represent the evolution of a set of narrative properties of a story as curves. As the conceptual space is traversed to find a suitable story, this evolution is iteratively compared with a set of objective curves. This comparison informs the traversal on every step and this information can be used as an additional source for deciding when a partial story is promising and whether a story is finished.

Previous versions of *STellA* also considered these methods for plot generation, but they were applied differently. Objectives and constraints did were not as powerful as they are in this version regarding their both their expressive power and their scope. While the current version allows for evaluation of a complete story, previously only states were considered, additionally, full access to the world representation is allowed now. Curves have a more general definition now since they define generic metrics (distances, average values and others) and previous versions needed more elaborated definitions. This has been made easier by the use of a simulation-based representation.

Algorithm 1 describes the overall generation algorithm. The non-determinism occurs, as previously described, when generating candidate sets of *deus ex*, *character desires* and *character intentions* actions. The generation algorithm iterates until a satisfying story is found and filters those exploratory branches that are unpromising according to the constraints imposed in the execution.

Data: the current partial story $[\langle state, e, p, d, i, w\rangle]$
objective curves
objective function
constraint function
Result: a set of candidate new tuples
while *current story is not finished according to curves and objectives* **do**
  $\sigma \longleftarrow$ last state tuple from current story
  $p \overleftarrow{\text{non-det}}$ *perception* for σ ordered by probability
  $e \overleftarrow{\text{non-det}}$ *deus ex* for σ ordered by probability
  $d \overleftarrow{\text{non-det}}$ *desire* for σ ordered by probability
  $i \overleftarrow{\text{non-det}}$ *intention* for σ ordered by probability
  $w \overleftarrow{\text{non-det}}$ *physical world* for σ ordered by probability
  $\sigma' \leftarrow$ apply (e, p, d, i, w) to σ
  $curves_{\sigma'} \leftarrow$ compute current curves for σ'
  new story $\longleftarrow$ current story $+ \sigma'$
  if $curves_{\sigma'} \approx curves_{objective} \wedge$ *new story satisfies constraints* **then**
    foreach σ' **do**
      explore generation from σ'
    end
  else
    reject σ'
  end
end
return current story

Algorithm 1: Story generation algorithm in *STellA*

Example Output

The described model has been implemented in three main modules:

1. The core engine for generating stories, containing the non-deterministic algorithms and basic narrative data structures.
2. The simulation engine defining the basic data structures and rules for the simulation to happen.
3. The set of rules both for generating actions and for defining story objectives.

The core engine (1) corresponds to the implementation of Algorithm 1 and the simulation engine (2) has been implemented according to the model previously described. A rule set (3) for an example prototype has been created for demonstration purposes. This rule set and the sample world place the action in a dungeon from which humans must escape.

The simulated world is a two-dimensional grid in which every entity is placed in one single cell. Basic actions of

characters are $move$ in eight directions, $attack$ adjacent enemies, eat food, $escape$, $protect$ themselves and others, $take$ and $drop$ objects and $apply$ objects on other entities (for healing an ally, for instance). Characters and creatures can sense their surroundings and use an A* based pathfinder to go from one place to another. Characters loose energy for being injured and doing things. The initial state includes 3 humans (located at one edge of the dungeon) and 5 creatures (located at the opposite edge, nearby the exit). Humans *desire* to escape and creatures are hungry and will try to eat the humans. Food, shields and weapons are spread out over the dungeon (10 items in total). The layout of the dungeon and the location of objects have been randomized.

Three objective curves have been used to drive the generation in this example. These curves have simple definitions and try to capture the evolution of measurable aspects of the story that, in the current domain, match to some extent specific features of the narrative arc:

- *danger*, the perceived danger in the story, computed as the mean distance between humans and creatures.
- *success*, the level of success of characters, computed as the difference between humans that have escaped the dungeon and the number of humans that have died.
- *richness*, an additional measurement to ensure that the generation is rich enough, computed as the number of different actions that happen in the story. Richness avoids monotonous stories in which characters just find their way to the exit without any conflict.

The input objective curves for the generation are a monotonously increasing line for *danger*, *richness* and *success*, forcing the generation to produce a story with an ending in which many things have happened (richness), the creatures surround the characters at the end (danger) and all characters escape (success).

In order to keep the demonstration prototype simple, one single objective function has been used: no humans must remain in the dungeon (Figure 1). Analogously, the only constraint used for the example forbids states in which the group of humans splits up, the average distance between humans must be lower that a certain threshold (Figure 3).

An example execution would start as follows: the generation starts as shown in Algorithm 1. First, the initial state is tested against the objective function which is not satisfied because there are 3 humans in the dungeon. Perception actions are computed and every cognitive entity (humans and creatures) update their internal representation of the world with their surrounding area. Deus ex rules are processed and no action is triggered, then desire rules are examined. A human with low energy desires to get food with a high priority (escaping is postponed) and the other two still decide to escape. All creatures decide to look for food. When intention actions are generated, all characters decide to move to find what the desire and this move is realized as a successful physic action because no obstacle limits their movement.

After this step, the current values for the objective curves are computed and compared against the objective curves. The difference between the current and the objective curves is acceptable by the system (being the first step yields the resulting comparison negligible according to the thresholds). This state is thus valid and new other candidates from the initial state are similarly generated and filtered. Then one of this states in chosen (the current prototype choses the one with a higher number of actions) and the generation continues until the system has found a satisfying story.

Then, the sequence of states and their corresponding actions are converted into a textual story. The rendering of the generated *fabula* as a *discourse* has been carried out with simple, ad-hoc rules to improve the apparent result. Figure 5 shows an example. Some redundant, easy to infer events and states were filtered (Figure 4) and sequential order was used (that is, events are told in the same order as they occur). The focus on the current prototype has not been put on the quality of the discourse and only a simple method has been used. Better narrative discourse planning, however, will be tackled in future versions of *STellA*. Figure 6 shows a fragment of the rendered output. The fragment has been selected by hand, but the whole story has been taken as-is without any form of curation or human intervention. Figure 7 shows part of the underlying representation corresponding to the text in Figure 6. The example shown corresponds to the sentence "the knight was hungry".

$$
\begin{aligned}
\texttt{promising}(story) \leftarrow\ & hs = \texttt{humans}(story) \\
& \forall h_i \in hs : \\
& \quad \forall h_d \in hs - \{h_i\} : \\
& \quad\quad d_i = \texttt{distance}(story, h_i, h_d) \\
& av = \texttt{average}(d_0, d_1, \ldots, d_n) \\
& \texttt{av} <= threshold
\end{aligned}
$$

Figure 3: Example of constraint rule. A partial story not satisfying a constraint rule will not be accepted as promising and its corresponding state will not be explored.

The fragment chosen and shown in Figure 6 exemplifies the level of detail that *STellA* is able to achieve. Specific focus on some generated events can shed more light on what *STellA* is able to do. For instance, when the knight is injured by the attack of the red creature, a new set of possible next steps in the simulation are generated. In some of them, the barbarian is not aware of the event and thus there is no reaction. According to the rules, these have a low probability of happening because the barbarian and the knight are nearby. In some others, chosen before because of their higher probability, the barbarian detects the attack. Since there is a rule stating that humans defend themselves against the creatures, the barbarian could non-deterministically choose what to do, either defend or ignore the knight. The system performs a space search to choose the best option among these two, that is, non-deterministically explores partial simulations from the current one and chooses the chain that fits the curves better. Since defending the knight maximizes the number of alive heroes, that one is chosen. In this way, the simulation and the narrative-based conceptual space search pro-

```
[...]
if event action is "pass" then
    filter event
end

if event action is "move" and
       character does not face enemy then
    filter event

if event action is "get tired" and
       character's energy > 100 then
    filter event
end
[...]
```

Figure 4: Simple event filtering for demonstration purposes. The current prototype includes ad-hoc rules for redundant or excessively detailed events.

```
[...]
if kindOf(entity) = "knight" then
    print "the knight "
end

if energy(entity) < 1500 then
    print "was hungry"
end

if energy(entity) < 1500 then
    print "blocked "
    print attackerOf(entity)
    print " with "
    print objectDefense(entity)
end
[...]
```

Figure 5: Example rule for discourse and textual generation in *STellA*. The current version addresses simple text for demonstration purposes.

duce rich, meaningful stories.

The grounded representation allows a fine level of granularity in the action and the narrative information leads to relatively interesting scenes according to the formal metrics described in term of narrative curves and specific requirements encoded as objectives and constraints. Generating detailed interactions can provide rich content that an accurate discourse planner can aggregate where needed. However, this does not mean that any form of verbose or redundant generation can be easily fixed by a discourse planner. The content generator should be able to provide reasonably meaningful and useful content letting the discourse planner decide what is relevent for each kind of discourse.

Discussion

The empirical evidence during the development suggests that the initial effort needed for grounding knowledge pays off soon. While more research and comparable measurements are needed to make any strong claim, the development process and the relative effort to include rules in the system is relatively reduced as the system evolves.

As previously detailed, many simulation-based story generation systems have already been created. *STellA* contributes to the field by focusing on *creativity* and exploration of a conceptual space. More specifically, several studied story generation systems perform a guided simulation in which some sort of general objectives (be it author or character goals) are pursued and fulfilled in a valid story (Lebowitz 1985; Dehn 1981; Theune et al. 2003). While the conjunction of goals and simulation links these systems with the presented version of *STellA*, the taken approach here is conceptually different: the simulation happens with *no narrative information* and the simulation is let to progress nondeterministically thus producing a growing tree of plausible states. Narrative is only included as an external process in which these successive simulations are selected as partial artifacts in the conceptual space. This puts a clear division between content generation with robust grounded generation and detailed filtering based on narrative rules. This somehow resembles the engagement and reflection model described by Sharples (Sharples 1999) and implemented in MEXICA (Pérez y Pérez 1999) in the sense that a model of creativity receives the focus.

Other story generation systems rely on the underlying narrative-like features of logging the simulation of character actions and put little or no effort on making an explicit narrative model (Klein et al. 1973; Meehan 1977). This clearly contrasts with the approaches taken by *STellA*, which specifically focus on using narrative to control which simulations are plausible according to the current objectives.

STellA explicitly addresses creativity both as a model and as objective. From a theoretical point of view and according to the theoretical framework described by Boden (Boden 2003) and formalized by Wiggins (Wiggins 2006), the nondeterministic simulation process would generate the conceptual space, and the mechanisms described to select and filter states would match the definition of the traversal function. The evaluation function would be composed by a mix of the curves and the objective function. The current prototype, however, is not reaching any high form of narrative creativity. The kind of story generation that *STellA* tries to achieve necessarily implies a complex management of knowledge and narrative structures. Before trying to create highly valu-

[...]
the knight was hungry.
the barbarian was injured.
the knight desired to protect the barbarian.
the green creature wanted to eat the barbarian.
the green creature tried to attack the barbarian.
the knight blocked the green creature with the shield.
the red creature tried to attack the knight.
the red creature succeeded when trying to attack the knight.
the knight was injured.
the barbarian desired to protect the knight.
the barbarian used the healing potion on the knight.
the barbarian desired to attack the green creature.
the knight desired to protect the barbarian.
the green creature tried to attack the barbarian.
the knight failed to block the green creature with the shield.
the green creature succeeded when trying to attack the barbarian.
the barbarian died.
the knight took the sword.
the knight desired to attack the green creature.
the knight tried to attack the green creature.
the knight succeeded when trying to attack the green creature.
the green creature died.
[...]

Figure 6: Fragment of a resulting story generated by *STellA* after the narrative-driven simulation process.

```
"knight0" : {position : (5, 51),
             energy : 1288,
             desire : {
                 desire : "escape",
                 agent : "knight0"
             },
             items : {"shield0"},
             kindOf : "knight",
             strength : 100,
             speed : 3,
             sight : 7,
             weigth : 90,
             known : {
                 "knight0" : {...},
                 "creature0" : {...},
                 "wall26" : {...},
                 "wall27" : {...},
                 [...]
             }
            }
```

Figure 7: Fragment of the underlying representation corresponding to the text in Figure 6.

able stories, the detailed development line tries to build a robust framework that can be further improved with more knowledge. The preliminary results show that world representation can be made richer by simulation and that a creative process can be model by non-deterministic generation and explicit filtering and identification of valuable artifacts.

Conclusions and Future Work

Simulation is a powerful tool for modelling interactions and can produce grounded information. This information, when properly identified, can be used for driving story generation if enriched with narrative knowledge and generate a conceptual space of stories.

This paper has described the development of an updated version of *STellA*, a story generation system that implements this model that mixes simulation and conceptual space exploration driven by narrative constructions. An example output generated by the current implementation is described and the relative benefits and drawbacks of the proposed solution are discussed.

The system will continue to be developed according the discussed assumptions, namely that generating successive story states by simulating relations between characters and constructing a conceptual space by using narrative information is a plausible method for generating rich stories that can be deemed as creative by unbiased observers (Colton and Wiggins 2012). Thorough work, however, is still to be done for the system fully support these assumptions: the simulation must support richer constructions and the generation process based on narrative must be improved with more general information about narrative, probably with general models borrowed from narratology.

Studying how driven non-determinism and probabilities can lead to better results in terms of novelty is a key aspect of the future improvements of *STellA*. The future work contemplates producing and evaluating stories that include unlikely events in such a way that novelty and quality are ensured to some measurable extent.

Acknowledgments

This paper has been partially supported by the projects WHIM 611560 and PROSECCO 600653 funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program.

References

Aylett, R. S.; Louchart, S.; Dias, J.; Paiva, A.; and Vala, M. 2005. Lecture notes in computer science. London, UK, UK: Springer-Verlag. chapter Fearnot!: An Experiment in Emergent Narrative, 305–316.

Baral, C. 2003. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. New York, NY, USA: Cambridge University Press.

Bell, M. 1985. Why expert systems fail. *The Journal of the Operational Research Society*.

Boden, M. 1999. Computational models of creativity. *Handbook of Creativity* 351–373.

Boden, M. 2003. *Creative Mind: Myths and Mechanisms*. New York, NY, 10001: Routledge.

Bratman, M. E. 1987. *Intention, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.

Bringsjord, S., and Ferrucci, D. 1999. *Artificial Intelligence and Literary Creativity: Inside the mind of Brutus, a Story-Telling Machine*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Cavazza, M.; Charles, F.; and Mead, S. J. 2002. Planning characters' behaviour in interactive storytelling. *Journal of Visualization and Computer Animation* 13:121–131.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *ECAI*, 21–26.

Dehn, N. 1981. Story generation after tale-spin. In *In Proceedings of the International Joint Conference on Artificial Intelligence*, 16–18.

Klein, S.; Aeschliman, J. F.; Balsiger, D.; Converse, S. L.; Court, C.; Foster, M.; Lao, R.; Oakley, J. D.; and Smith, J. 1973. Automatic novel writing: A status report. Technical Report 186, Computer Science Department, The University of Wisconsin, Madison, Wisconsin.

Lebowitz, M. 1985. Storytelling as Planning and Learning. *Poetics* 14:483–502.

León, C., and Gervás, P. 2010. The Role of Evaluation-Driven rejection in the Successful Exploration of a Conceptual Space of Stories. *Minds and Machines* 20(4):615–634.

León, C., and Gervás, P. 2011. A top-down design methodology based on causality and chronology for developing assisted story generation systems. In *Proceedings of the 8th ACM conference on Creativity and cognition*, C&C '11, 363–364. New York, NY, USA: ACM.

León, C., and Gervás, P. 2012. Prototyping the use of plot curves to guide story generation. In *Third Workshop on Computational Models of Narrative, 2012 Language Resources and Evaluation Conference (LREC'2012)*.

Mateas, M., and Stern, A. 2005. Structuring content in the Faade interactive drama architecture. In *Proceedings of AIIDE*, 93–98.

Meehan, J. R. 1977. Tale-spin, an interactive program that writes stories. In *In Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 91–98.

Pérez y Pérez, R. 1999. *MEXICA: A Computer Model of Creativity in Writing*. Ph.D. Dissertation, The University of Sussex.

Riedl, M., and Young, M. 2010. Narrative planning: Balancing plot and character. *J. Artif. Intell. Res. (JAIR)* 39:217–268.

Riedl, M. 2004. *Narrative Planning: Balancing Plot and Character*. Ph.D. Dissertation, Department of Computer Science, North Carolina State University.

Rosati, R. 2007. The limits of querying ontologies. In *In Proceedings of the Eleventh International Conference on Database Theory (ICDT 2007)*, 164–178. Springer-Verlag.

Schank, R., and Abelson, R. 1977. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. Hillsdale, NJ: L. Erlbaum.

Sharples, M. 1999. *How We Write*. Routledge.

Sloman, A. 1985. *Why We Need Many Knowledge Representation Formalisms*. Cognitive studies research papers. University of Sussex, Cognitive Studies Programme.

Sowa, J. 2000. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Pacific Grove, CA: Brooks/Cole.

Theune, M.; Faas, E.; Nijholt, A.; and Heylen, D. 2003. The virtual storyteller: Story creation by intelligent agents. In *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference*, 204–215.

Tolkien, J. R. R. 1972. *The Hobbit ; There and back again*. George Allen and Unwin London, [3d ed.]. edition.

Trentelman, K. 2009. *Survey of knowledge representation and reasoning systems*. Defence Science and Technology Organisation Edinburgh, S. Aust.

Turner, S. 1992. *MINSTREL: A Computer Model of Creativity and Storytelling*. Ph.D. Dissertation, University of California at Los Angeles, Los Angeles, CA, USA.

Wiggins, G. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7).

Baseline Methods for Automated Fictional Ideation

Maria Teresa Llano, Rose Hepworth, Simon Colton, Jeremy Gow and John Charnley
Computational Creativity Group, Department of Computing, Goldsmiths, University of London

Nada Lavrač, Martin Žnidaršič and Matic Perovšek
Department of Knowledge Technologies, Jožef Stefan Institute

Mark Granroth-Wilding and Stephen Clark
Computer Laboratory, University of Cambridge

Abstract

The invention of fictional ideas (ideation) is often a central process in the creative production of artefacts such as poems, music and paintings, but has barely been studied in the Computational Creativity community. We present here three baseline approaches for automated fictional ideation, using methods which invert and alter facts from the ConceptNet and ReVerb databases, and perform bisociative discovery. For each method, we present a curation analysis, by calculating the proportion of ideas which pass a typicality evaluation. We further evaluate one ideation approach through a crowdsourcing experiment in which participants were asked to rank ideas. The results from this study, and the baseline methods and methodologies presented here, constitute a firm basis on which to build more sophisticated models for automated ideation with evaluative capacity.

Introduction

Ideation is a portmanteau word used to describe the process of generating a novel idea of value. Fictional ideation therefore describes the production of ideas which are not meant to represent or describe a current truth about the world, but rather something that is in part, or entirely, imaginary. As such, their purposes include unearthing new truths and serving as the basis for cultural creations like stories, advertisements, poems, paintings, games and other artefacts. Automated techniques for the derivation of new concepts have been important in Artificial Intelligence approaches, most notably machine learning. However, the projects employing such techniques have almost exclusively been applied to finding concepts which somehow characterise *reality*, rather than some fictional universe. While some concepts may be purported as factual, i.e., supported by sufficient evidence, others may only be hypothesised to be true. In either case, however, the point of the exercise is to learn more about the real world through analysis of real data, rather than to invent fictions for cultural consumption.

A major sub-field of Computational Creativity research involves designing software that exhibits behaviours perceived as creative by unbiased observers (Colton and Wiggins 2012). However, in the majority of the generative systems developed so far within Computational Creativity research, there is no idea generation undertaken explicitly. An exception to this was (Pereira 2007), who implemented a system based on the psychological theory of Conceptual Blending put forward by Fauconnier and Turner (2008). By blending two theories about different subject material, novel concepts which exist in neither domain emerge from the approach. Using blending to reason about such fictional ideas was harnessed for various creative purposes, including natural language generation (Pereira and Gervás 2003), sound design (Martins et al. 2004), and the invention of character models for video games (Pereira and Cardoso 2003). Similarly, the ISAAC system (Moorman and Ram 1996) implements a theory for creative understanding based on the use of an ontology to represent the dimensions of concepts. By altering the dimensions of existing concepts within the ontology, for instance considering a temporal object as a physical one, the system is able to create novel concepts.

In addition, in some projects, especially ones with application to natural language generation such as neologism production (Veale 2006), which are communicative in nature, it is entirely possible to extract ideas from the artefacts produced. However, it is fair to say that such software is not performing ideation to produce artefacts, but is rather producing artefacts that can be interpreted by the reader via new ideas. The work in (Goel 2013) shows the use of creative analogies in which problems of environmental sustainability are addressed by creating designs inspired by the way things work in nature. For instance, birds' beaks inspired the design of trains with noise reduction. Although ideation here is being used for inspiration and not to create literal representations, this work shows the potential of using creative analogies for fictional ideation.

As part of the WHIM project[1] (an acronym for the *What-if Machine*), we are undertaking the first large-scale study of how software can invent, evaluate and express fictional ideas. In the next section, we present three straightforward approaches to fictional ideation which manipulate material from internet sources. These will act as our baseline against which more sophisticated ideation methods will be tested as the project progresses. In order to draw that baseline, we conducted a *curation analysis* of the ideas produced by each method, whereby we calculated the proportion of ideas which were typical in the sense of being both understandable and largely fictional, with details given below. We also

[1] www.whim-project.eu

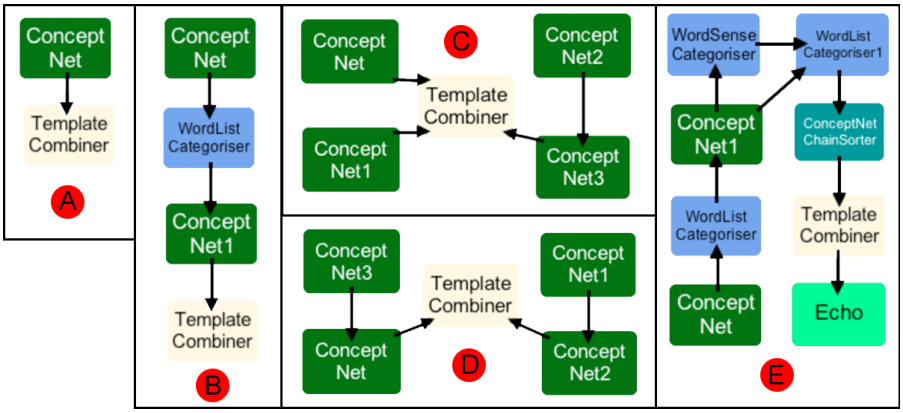


Figure 1: Ideation flowcharts using ConceptNet.

present here a baseline methodology for estimating the true value of the ideas produced by our systems. To do this, we conducted a crowd-sourcing exercise involving 135 participants, where people were exposed to ideas in a controlled way, with the aim of evaluating components of ideas that could be used to predict overall value.

A good fictional idea distorts the world view around it in useful ways, and these distortions can be exploited to spark new ideas, to interrogate consequences and to tell stories. A central hypothesis of the WHIM project is that the narrative potential of an idea can be estimated automatically, and used as a reliable estimate of the idea's worth. Hence the crowd-sourcing study had narrative potential as a focal point, and we tested an automated approach which estimates whether an idea has much narrative potential, or little. As discussed below, we found that, in general, people ranked those ideas that were assessed as having much potential higher than those assessed as having little. We present further statistical analysis of the results, which enables us to conclude by describing future directions for the WHIM project.

Baseline Ideation Methods

We investigate here three methods which use data mined from the internet for generating What-if style fictional ideas. In the next section, we analyse the results from each method.

Fictional Ideation using ConceptNet

ConceptNet[2] is a semantic network of common sense knowledge produced by sophisticated web mining techniques at the MIT media lab (Liu and Singh 2004). Mined knowledge is represented as facts, which comprise relations between concepts in a network-like structure, e.g., [*camel, IsA, animal, 7.0*], [*animal, CapableOf, hear_sound, 2.0*]. Currently, ConceptNet has 49 relations, including UsedFor, IsA, AtLocation, Desires, etc., and each fact is given a score, from 0.5 upwards, which estimates the likelihood of the relation being true, based on the amount of evidence mined. We have studied fictional ideation by *inverting* the world view modelled by ConceptNet, i.e., facts are transformed by negating their relations. For example, this can be done by introducing an action which was not previously possible, e.g., 'people can't fly' becomes *What if people could fly?* or stopping an action or desire which was previously common, e.g., 'people need to eat' becomes *What if people no longer needed to eat?*, etc. We investigated various inversion methods such as these, carried out using the FloWr flowcharting system described in (Charnley, Colton, and Llano 2014).

Working in a story-generation context, we took inspiration from the opening line of Franz Kafka's 1915 novella *The Metamorphosis*:

> "One morning, as Gregor Samsa was waking up from anxious dreams, he discovered that in his bed he had been changed into a monstrous verminous bug".

In figure 1, we present five flowcharts we used to generate ideas by inverting and combining ConceptNet facts about people, animals, vegetables and materials.

Flowchart A finds instances of animals by searching ConceptNet for facts [X, IsA, animal]. These are then rendered in the *TemplateCombiner* process as questions of the form: "What if there was a person who was half man and half X?" Flowchart B employs ConceptNet similarly, then uses a *WordListCategoriser* process to remove outliers such as [my_husband,IsA,animal]. Then, for a given animal, A, facts of the form [A,CapableOf,B] are identified and rendered as: "What if there was a person who was half man and half X, who could Y?" Switching the CapableOf relation to NotCapableOf enabled us to produce ideas suggesting a person who became an animal, but retained some human qualities. We augmented this by using the LocatedNear relation (not shown in figure 1) to add a geographical context to the situation, producing ideas such as "What if a woman awoke in the sky to find she had transformed into a bird, but she could still speak?" We found that these ideas had much resonance with the premise in *The Metamorphosis*.

Taking our lead next from the surrealistic artworks of Dali, Magritte and colleagues, in flowchart C, we looked at bizarre visual juxtapositions. ConceptNet is used here to find an occupation, a vegetable and a location related to some animal, and the flowchart produces ideas such as: "What if there was a banker underwater with a potato for a face?" Similarly, in flowchart D, we produced ideas for paintings by finding materials, M, using facts of the form

[2] conceptnet5.media.mit.edu

[X,IsA,thing] and [X,MadeOf,M], then finding organisms, O, with pairings of [X,IsA,live_thing] and [O,IsA,X] facts. This led to ideas such as painting a dolphin made of gold, a reptile made of wood, and a flower made out of cotton. In the baseline evaluation section below, we describe the raw yield of flowcharts A to D, and the proportion of the results which were both understandable and mostly fictional.

As mentioned above, we are particularly interested in estimating the *narrative potential* of an idea, by which we mean the likelihood that the idea could be used in multiple, interesting and engaging plots for stories. As a baseline method for estimating such potential, we investigated a technique consisting of building inference chains of ConceptNet facts whose starting point is the fact that is inverted in the idea. To illustrate the approach, from the seed idea *"What if there was a little bug who couldn't fly?"*, the following chain of relations can be obtained through ConceptNet:

[bug,CapableOf,fly] → [fly,HasA,wing] → [wing,IsA,arm] → [arm,PartOf,person] → [person,Desires,muscle] → [muscle,UsedFor,move_and_jump]

Here, one can imagine a bug who can't fly, but instead uses his muscle-bound human like arms for locomotion.

Our hypothesis is that, while each chain might be rather poor and difficult to interpret as a narrative, the volume and average length of such chains can indicate the potential of the idea. We implemented a *ConceptNetChainSorter* process to take a given idea and develop chains up to a specified length with no loops or repetitions. Flowchart E uses this process to order the facts from ConceptNet in terms of the sum of the lengths of the chains produced. Hence facts with many chains are ranked higher than chains with fewer, and longer rather than shorter chains will also push a fact up the rankings. Often there are no chains for a fact, and if there are, the number depends on the nature of the objects being related, and the relation. Looking at facts [X,R,Y], where [X,IsA,animal] is a ConceptNet fact, for each R, we found these percentages of facts had non-trivial chains:

CapableOf	Desires	HasA	HasProperty	IsA	LocatedNear
20	50	63	28	48	100

Fictional Ideation using ReVerb

The Washington ReVerb project (Fader, Soderland, and Etzioni 2011) extracts binary relationships between entities from text, like the ConceptNet relations described above. Output produced by running the system over a large corpus of web texts (ClueWeb09, ∼1 billion web pages) is publicly available and we use it here to generate fictional ideas. Lin, Mausam, and Etzioni (2012) have linked the first argument (LHS concept) of a subset of ReVerb extractions with identifiers of an entity in Freebase (Bollacker et al. 2008). This provides a means of unifying the various names by which a particular entity might be referred to (*cow, cattle*, etc.) and disambiguating entities that have the same name. In the ideation method described here, we use this dataset, and the input to the process is a Freebase ID.

The relations vary in generality, as well as reliability. For example, some relations express a particular one-off event during which the entities interacted (*Tony Blair converted to Catholicism*), while others express general properties of the entities (*cows eat grass*). Both types of relations may be of interest to building world views for ideation, and we do not attempt to distinguish them currently. Using facts from ReVerb, we can generate fictional ideas by substituting one of the arguments for an alternative entity. For example, the extractions relating to *cattle* include *[Cattle, were bred for, meat]*. Looking at other facts that use the same relation (*be bred for*), with different LHS entities, we find things that are bred for *speed*, suggesting a possible fictional fact: *[Cattle, were bred for, speed]*.

The following are desirable properties of such alterations:

1. They should be fictional (e.g., *[Cattle, were bred for, meat]* $\not\Rightarrow$ *[Cattle, were bred for, milk]*).
2. They should make sense (e.g., *[Cattle, were bred for, meat]* $\not\Rightarrow$ *[Cattle, were bred for, rule of thumb]*).
3. They should have a substantial effect on the narratives that could be generated (e.g., *[Cattle, were bred for, meat]* $\not\Rightarrow$ *[Cattle, were bred for, hamburgers]*).

Establishing whether this last desideratum holds is a hard task which we leave for now to future work.

Given an extraction $[X, r, Y]$, we wish to generate a fictional $[X, r, Y']$. The following requirements might serve to approximate the first two desiderata above:

- $[X, r, Y'']$ is common for some Y'', i.e., r is a common type of fact to say about X.
- $[X', r, Y']$ is common, i.e., Y' is commonly seen as the second argument of r (with different first arguments).
- $[X, r, Y']$ is rarely or never seen, i.e., this is likely not a fact we are already aware of. As we cannot rely on the dataset to contain all relevant facts, we impose a strong version of this, that $[X, r, Y']$ is completely unattested.

As an example, the following alteration is well supported by these criteria: *[Michael Jackson, was still the king of, pop]* ⇒ *[Michael Jackson, was still the king of, Kong]*. The initial fact is chosen because Michael Jackson is frequently said to have been the king of things (popular music, music video, etc.) – the first requirement. Kong is chosen as an alternative second argument, because Kong ranks highly among things that people are described as being still king of[3] – the second requirement. Finally, we have never seen Michael Jackson described as being still the king of Kong.

The first two requirements given above can be expressed, and combined, as conditional probabilities. $P(r|X)$ represents the probability of the relation given the first argument (the input). This will be high for the relations most often seen with X as the first argument (the most common things to say about X). $P(Y'|r)$ will likewise be high for the most common second arguments of the relation in question, regardless of which X they have been seen with. To eliminate attested facts, we exclude any Y' seen at all in $[X, r, Y']$. For each of the top 100 facts about X found in the ReVerb extractions, all alterations Y' with a non-zero $P(Y'|r)$ are ranked according to $P(r|X) \times P(Y'|r)$.

[3] High-scorers in the game Donkey Kong are described as such.

Below are some examples of the alterations the system performs, with an analysis of the proportion of usable alterations given in the next section. The following are the top five alterations for entity *cattle*, showing the fact in its extracted form, then the system's alteration, which could be rendered as a What-if style idea:

1. Cattle evolved to eat grass ⇒ Cattle evolved to eat meat
2. Cattle occupy a unique role in human history ⇒ Cattle occupy a unique role in Israelite history
3. Cattle occupy a unique role in human history ⇒ Cattle occupy a unique role in modern distributed systems
4. Cattle occupy a unique role in human history ⇒ Cattle occupy a unique role in society
5. Cattle were bred for meat ⇒ Cattle were bred for speed

Similarly, the top five for *Scotland* are:

1. Scotland is steeped in history ⇒ Scotland is steeped in tradition
2. Scotland is a part of the United Kingdom ⇒ Scotland is a part of life
3. Scotland is in Britain ⇒ Scotland is in trouble
4. Scotland is in Britain ⇒ Scotland is in order
5. Scotland is in Britain ⇒ Scotland is in progress

In other tests, we produced ideas that express fictional histories, which is a mainstay of creative writing, for instance: "What if John F. Kennedy had been elected Pope?"

Fictional Ideation using Bisociative Discovery

Koestler (1964) stated that different types of invention all share a common pattern, to which he gave the term "bisociation". According to Koestler, bisociative thinking occurs when a problem, idea, event or situation is perceived simultaneously in two or more "matrices of thought" or domains. When two matrices of thought interact with each other, the result is either their fusion in a novel intellectual synthesis, or their confrontation in a new aesthetic experience.

The developers of the CrossBee system (Juršič et al. 2012) followed Koestler's ideas by exploring a specific form of bisociation: finding terms that appear in documents which represent bisociative links between concepts of different domains, with a term ranking method based on the voting of an ensemble of heuristics. We have extended this methodology with a banded matrices approach, described in (Perovšek et al. 2013), which is used in a new CrossBee heuristic for evaluating terms according to their bridging term (b-term) potential. The output from CrossBee is a ranked list of potential domain bridging terms. Inspecting the top-ranked b-terms should result in a higher probability of finding observations that lead to the discovery of new links between different domains. Here, the creative act is to find the links which cross two or more different domains, leading out of the original 'matrix of thought'.

In the simplified ideation scenario addressed here, we used CrossBee for b-term ranking on documents from two domains to discover bridging terms, with the aim of combining statements from two domains. The first domain consists of 154,959 What-if sentences retrieved from Twitter with query 'what if', assisted by the Gama System® PerceptionAnalytics platform.[4] The tweets were filtered through the following steps, reducing the number to 65,811:

- All non-ASCII characters were deleted.
- Repeated letters were truncated, so that any character repeating consecutively more than twice in a word was ignored after the second repetition. For example, the word *cooooool* would be truncated to *cool* (but also *looooooove* would be truncated to *loove*).
- All characters are transformed to lower case.
- Non-English tweets were removed.
- Vulgar words were removed by comparison with a list of such words.[5]
- From all items, only the sub-strings starting with the term 'what if' and ending with a period, question mark or exclamation mark were considered.
- Items shorter than 9 characters were removed.
- Exact duplicates were removed.

The second dataset is a collection of 86 moral statements from Aesop's fables, which was created by crawling the Aesop's fables online collection. Each What-if sentence and each moral statement was treated as a separate document, and all documents were further preprocessed using standard text mining techniques. We then applied our methodology to the data from the two domains to estimate the b-term potential of common terms. We used this indicator for ranking (a) single What-if sentences and (b) bisociatively linked What-if sentences and moral statements.

Inspection of the What-if sentences obtained from tweets revealed that a great number of them make very little sense in general or are related to very specific contexts. Aesop's morals, on the other hand, tend to be very general in nature. By composing sentences from these two domains using the terms with the best b-term potential indicator value, we hoped to produce a ranking mechanism that favours generally meaningful fictional ideas that might be useful for ranking individual What-if sentences. We used the mechanism to rank both single sentences and compound pairs, to test the hypothesis that using the b-term potential as an ranking coefficient can estimate which What-if sentences will be evaluated more favourably by people, both as individual sentences and in bisociatively combined sentence pairs.

The effectiveness of b-term potential used as a ranking tool of single What-if sentences was evaluated as follows: we randomly shuffled the 10 best ranked sentences and 10 random What-if sentences. The collection of 20 sentences was then independently assessed by 6 human evaluators who used scores from 1 (bad) to 5 (very good) in answering the question: "*How good (generally interesting) do you find the following idea?*" The top 10 b-term ranked What-ifs received an average score of 2.92, whereas the randomly chosen ones scored 2.80 on average. Application of an Unpaired

[4] demo.perceptionanalytics.net

[5] urbanoalvarez.es/blog/2008/04/04/bad-words-list/

T-test suggests that the difference among these two scores is not significant (p=0.6736). The best ranked What-if, according to the b-term potential was: "*What if a called myself the pope then charged into the vatican and demanded a duel to the death with an old man?*" This was also the sentence that achieved the best average score from the human evaluators.

The impact of b-term potential ranking on compound sentence pairs was evaluated similarly. To do this, we took the top 4 What-ifs and the top 4 moral statements that contained the strongest b-term. By combining them, we created a collection of 16 pairs of sentences. This collection was compared to two other collections: (i) a collection of 16 pairs of sentences (What-if + moral) that shared a b-term regardless of its strength, and (ii) a collection of 16 randomly paired What-if and moral sentences. Our hypothesis was that the top ranked collection will score higher on average than the one with randomly ranked b-terms and significantly better than the one which was randomly put together, ignoring b-terms. The pairs were randomly shuffled and independently assessed by 6 human evaluators answering the question: "*How good do you find the combination of the two sentences?*", scored again from 1 to 5.

Surprisingly, the top ranked collection was scored significantly (p=0.0076) lower than the randomly ranked one, with average score of 2.43, compared to 2.96. Also, in an independent comparison, it scored lower than the randomly paired sentences, having an average score of 2.70 compared to 2.78, although this was not significantly lower (p=0.6677). The compound sentence pair with the best b-term rank was: "*What if a called myself the pope then charged into the vatican and demanded a duel to the death with an old man? Every man should be content to mind his own business*". However, this sentence pair was ranked only 8th best among 32 manually evaluated compound pairs.

Given the encouraging result of the ranking mechanism for single What-if sentences, and the bad performance on its target compound data, the usefulness of the bisociative discovery methods for ideation and idea assessment cannot be confirmed. Hence, we plan further implementation and experimentation. In particular, we will enlarge the dataset of moral statements, to strengthen the bisociation approach.

Curation Analyses

Recall that we plan to use the above ideation methods as a baseline against which to compare more sophisticated approaches as the WHIM project progresses. Colton and Wiggins (2012) introduce the term *curation coefficient* as an informal reading of the *typicality*, *novelty* and *quality* measures put forward in (Ritchie 2007). In essence, this involves a project team member examining the output from their generative software, and calculating the proportion that they would be happy to present to others. For our purposes here, we used slightly lower criteria: we took all the ideas from each method, or a sample when there were too many, and recorded how many were suitable for assessment, i.e., the proportion of ideas that were both understandable and fictional, without any judgement of quality.

In figure 1, we presented flowcharts A to D for generating fictional ideas using ConceptNet. Facts in ConceptNet are scored for truth likelihood, and flowchart A is parametrised by a threshold, T_1, for the minimum score that ConceptNet facts must achieve to be used. Flowchart B uses ConceptNet twice, hence has thresholds T_1 and T_2. Flowcharts C and D were not parametrised, and used a fixed ConceptNet threshold of 1. Table 1 shows the number of ideas (yield) that each flowchart (FC) produced, with various threshold settings. The table also shows the curation coefficient (C-Coeff), i.e., the proportion of understandable and (largely) fictional ideas. We see that the yield reduces as higher thresholds T_1 and T_2 are imposed, but the curation coefficient increases, because fewer spurious or nonsensical facts are inverted for the ideas. In one case for flowchart B, by setting T_1 and T_2 to 5, we were able to produce a set of 27 ideas with a 100% curation coefficient. We noted an average yield of 190.4 and an average curation coefficient of 84.1%.

FC	Example	T_1	T_2	Yield	C-Coeff(%)
A	He was half man, half bird	1	-	97	72
		3	-	21	90
		5	-	14	93
B	He was half man, half fish, who could live in a lake	5	1	453	78
		5	2	94	88
		5	5	27	100
B	He was a cat, but he could still write	5	1	48	88
		5	3	7	100
C	Composer in a nest with turnip for a face	-	-	272	56
D	Dolphin that is made out of gold	-	-	871	76
	Average			190.4	84.1

Table 1: Curation analysis: ConceptNet approach.

Criteria	Yield	C-Coeff(%)
Fictional	500	90.9
Understandable	500	94.6
Non-duplicate	500	73.6
Overall	500	59.1

Table 2: Curation analysis: ReVerb approach.

Evaluation	Yield	C-Coeff(%)
What-if + moral (b-term)	32	28.1
What-if + moral (random)	16	6.25

Table 3: Curation analysis: bisociative discovery approach.

We generated 500 ideas with the ReVerb approach, using as seed queries the top six names from an online list of the most famous people of all time[6]. There were three issues with the ideas: (i) some happened to be true facts, or very close to a true fact (e.g., *What if John Kennedy was elected vice president?*); (ii) some happened to be nonsensical (e.g., *What if Elvis Presley is inducted into St?*), and (iii) some were an exact or very close duplicate of one already seen in the output (e.g., *What if Leonardo da Vinci was born in New York?* and *What if Leonardo was born in New York?*). In table 2, we report the curation coefficients with each of

[6] www.whoismorefamous.com

these three issues in mind, and an overall coefficient for the ideas which have none of these issues. We see that each issue reduced the curation coefficient, which was 59.1% overall.

For the bisociative discovery approach, we performed an analysis of the ideas that combine a What-if sentence with a moral statement, since these are automatically generated, rather than just mined from Twitter. We compared the 32 sentence pair ideas where there was a shared b-term with the 16 randomly concatenated pairs of sentences. Table 3 shows the results of the curation analysis for the ideas from the bisociative discovery approach. We found that the ideas generated by the bisociative discovery method were entirely understandable, as they were concatenations of two already understandable sentences. However, the results were often non-fictional, because the method doesn't explicitly attempt to distort reality. This explains the low curation curation coefficient of 28.1% for the b-term method, but it is important that it significantly outperformed the random approach.

With the ConceptNet and ReVerb approaches, data-mined notions of reality were inverted and altered respectively, hence the ideas were largely fictional. With respect to nonsensical ideas, for the ConceptNet-based ideas, we learned that control over quality could be exerted, at the expense of yield, through the usage of the ConceptNet thresholds. For the ReVerb results, completely nonsensical ideas were rare, since we used only arguments that are well attested with the relation. Errors were generally due to the open-domain IE extraction method used to compile the original facts. With the ReVerb approach, many of the (almost) true ideas occur because of substitutions for similar arguments, e.g., substituting 'president' with 'vice-president'. The system cannot recognise that the two are similar, and consequently the output contains a high proportion of almost exact duplicates: often almost the same thing is substituted many times over. This suggests that the results could be improved by incorporating a measure of semantic similarity which prefers dissimilar substitutions. Alternatively, the data integration technique from (Yao, Riedel, and McCallum 2012) could be used by the system to rule out ideas that, although not seen explicitly before, are highly probably repeats, given the observed facts.

A Crowd-Sourcing Evaluation

Ultimately, the fictional ideas we want to automatically produce will be for general consumption. Hence a large part of the WHIM project will involve crowd-sourcing responses to fictional ideas and using machine learning techniques to derive an audience model that can predict whether generated ideas are going to be of value. To study a baseline methodology for this, and to get a first tranche of feedback from the general public, we focused on the ConceptNet approach within the context of anthropomorphised animal characters which could feasibly appear in a Disney animated film. This context was chosen because Disney movies are familiar to most people and somewhat formulaic, hence we could be reasonably confident that when we surveyed people, our questions would be interpreted appropriately.

During a pilot study reported in (Llano et al. 2014), we focused on ideas generated by the CapableOf relation in the second ConceptNet node of flowchart B in figure 1, i.e., we studied ideas of the type: "What if there was a little X, who couldn't Y?" With an online survey of four questions, we asked 10 English speaking participants to rank the same list of 15 such Disney characters, in terms of (a) general impression (b) emotional response provoked (c) *narrative potential*: number and quality of potential plot lines imaginable for the character, and (d) how surprising they found the character to be. Our aim was to measure the influence of emotional provocation, narrative potential and surprise on general impression. Recall that we wrote routines to produce chains of ConceptNet facts. The 15 Disney characters in the survey comprised 5 from ideas with no chains, 5 from ideas with multiple chains, and 5 ideas where the RHS of a ConceptNet fact was replaced with a randomly chosen verb.

This pilot study showed that ConceptNet ideas were ranked much higher than the random ones for three questions, with average ranks of 5.21 vs. 10.98 for general impression, 6.08 vs. 11.5 for emotional provocation and 5.00 vs. 11.32 for potential for narrative potential. Within the ConceptNet examples, those with chains were ranked slightly higher than those without: average ranks of 4.78 vs. 5.21 for general impression, 3.42 vs. 6.08 for emotional response and 4.68 vs. 5.00 for narrative potential. However, when assessing levels of surprise, the random ideas were ranked as best with an average rank of 4.48 vs. 8.18 for ConceptNet ideas with no chains, and 8.44 for those with chains. On reflection, we determined that this resulted from an inconsistent interpretation of the word 'surprising'. We also found in the pilot study that there was a strong positive correlation r between general impression and both emotional response (r=0.81) and narrative potential (r=0.87), confirming that both these elements are key components of participants' general impressions of value. However, we found a strong negative correlation between general impression and surprise (r=-0.77). Hence, this suggests that more surprising ideas aren't generally well received.

Building on and learning from the pilot study, we undertook a larger scale experiment. For this, we used three sets of Disney characters generated using ConceptNet facts with the CapableOf (CO) relation as before, in addition to the Desires (D) relation ("What if there was a little X who was afraid of Y?") and the LocatedNear (LN) relation ("What if there was a little X who couldn't find the Y?") In order to evaluate participants' preferences, we designed four surveys: one per relation, and a fourth that mixed Disney characters from the three relations. In order to prevent bias or fatigue, each participant completed only one of the surveys.

Each survey consisted of four questions that asked participants to rank Disney characters in order of their general impression (GI) of the character's viability, the degree of emotional response (ER) they felt upon reading and interpreting the idea of the character, the quantity and quality of the plot lines; i.e., narrative potential (NP), that they felt might be written about each, and to what level each character met their expectation (LE) of a Disney character. This last question replaced the final question from the pilot study. The relation-focused surveys had a set of 14 ideas, eight ConceptNet non-chaining (NC) ideas (i.e., only one associated

Q	CO		D		LN		Avg	
	NC	CC	NC	CC	NC	CC	NC	CC
GI	7.41	7.62	7.76	7.15	8.05	6.77	7.74	7.18
ER	7.88	7.00	8.03	6.80	7.85	7.03	7.92	6.94
NP	7.85	7.04	8.03	6.80	7.95	6.90	7.94	6.91
LE	7.95	6.90	8.15	6.63	8.01	6.81	8.04	6.78

(a) Average participant rankings for three relation-focused surveys by type of idea: Non-Chaining (NC) and ConceptNet Chaining (CC).

Q	Mixed		
	CO	D	LN
GI	7.48	7.70	8.81
ER	6.55	8.44	9.01
NP	7.86	7.48	8.66
LE	7.24	8.46	8.30

(b) Average participant rankings for Mixed survey by inverted relation.

	GI&ER	GI&NP	GI&LE
Avg. Corr. (τ)	0.34	0.36	0.31

	ER&NP	ER&LE	NP&LE
Avg. Corr. (τ)	0.35	0.32	0.37

(c) Average rank correlation between all the questions of the four surveys: General Impression (GI), Emotional Response (ER), Narrative Potential (NP) and Level of Expectation (LE) .

Q	Correlation (τ)				
	CO	D	LN	Mixed	Avg
GI	0.09	0.25	0.27	-0.24	0.09
ER	0.17	0.25	0.26	0.26	0.23
NP	0.22	0.22	0.21	0.23	0.22
LE	0.14	0.27	0.22	0.08	0.17

(d) Rank correlation between av. participant rankings & chaining rankings.

Q	Correlation (τ)														
	CapableOf			Desires			LocatedNear			Mixed			Avg		
	IsA	CO	CB	IsA	D	CB	IsA	LN	CB	IsA	Rel	CB	IsA	Rel	CB
GI	0.25	0.19	0.31	0.42	0.17	0.40	-0.17	0.34	-0.17	0.20	0.27	0.31	0.17	0.24	0.21
ER	0.18	0.22	0.25	0.51	0.10	0.49	-0.07	0.21	-0.03	0.22	0.40	0.39	0.21	0.23	0.27
NP	-0.02	0.07	0.03	0.46	0.07	0.44	-0.07	0.27	-0.03	0.23	0.26	0.26	0.15	0.16	0.17
LE	0.39	0.11	0.44	0.46	0.10	0.44	0.02	0.17	0.06	0.18	0.29	0.31	0.26	0.16	0.31

(e) Rank correlation between average participant rankings and ConceptNet relations rankings.

Figure 2: Crowd-sourcing experiment results for four surveys: CapableOf (CO), Desires (D), LocatedNear (LN) and Mixed.

chain) and six ConceptNet chained (CC) ideas (i.e., with multiple associated chains) – random ideas were not evaluated as they scored significantly worse in the pilot study. The mixed-survey used a set of 15 CC-ideas, five per relation. These ideas were chosen by sampling systematically at equal intervals in terms of chaining score.

Results

A total of 135 participants completed the crowd sourcing experiment, with at least 27 participants per survey. Contrary to the pilot study, the crowd sourcing evaluation was not restricted to native English speakers. Therefore, we had respondents with different levels of fluency: 1 was at a basic level, 12 consider themselves at an intermediate level, 68 participants were fluent and 54 were native English speakers. These figures show that at least 90% of the participants were fluent or native, which provides a high level of confidence in the reliability of the results. Moreover, 64 participants were female, 70 were male and 1 person preferred not to specify their gender. This shows an almost even participation from both genders. The participants were between 18 and 74 years old; more specifically, 12 were in the age range between 18 and 24 years old, 74 in the range 25-34, 33 in the range 35-44, 7 in the range 45-54, 7 in the range 55-64 and 2 in the range 65-74. The highest concentration is seen in participants between 25 and 34 years old; however, most age ranges were represented in the surveys. After completing the surveys we asked the participants to select their level of confidence, between *very low, low, medium, high* and *very high*, when answering each question. Table 4 shows that most of the participants answered each question with a medium level of confidence or higher. This increases the confidence we have in the results.

Figure 2(a) shows the average rankings given for each class of ideas in the relation-focused surveys. As suggested in the pilot study, in general, the CC-ideas are ranked around

	Percentage of Participants			
Question	CO	D	LN	Mixed
GI	97	90	94	96
ER	97	90	88.5	92.5
NP	78	82.5	83	85
LE	85	80	80	78

Table 4: Percentage of participants who answered each question with a medium level of confidence or higher.

1 position higher than the NC-ideas. This supports the hypothesis that the ConceptNet chaining evaluation technique provides a reliable measure of value for fictional ideation using ConceptNet. Using a Friedman test comparing the mean ranks for CC and NC ideas in each response, we found that the difference between their ranks is highly significant overall ($p<0.001$). This effect remained significant across all question and survey subgroups.

Figure 2(b), which presents the results from the fourth survey, shows that, in general, the CO-ideas were ranked highest, followed by the D-ideas and then the LN-ideas. A Friedman test showed these differences to be highly significant overall ($p=0.001$). Our interpretation is that participants considered that, in some cases, the D-ideas and LN-ideas failed with respect to the feasibility of the fictional characters they portrayed, therefore, they were ranked lower. More specifically, respondents suggested that they felt apathy towards anthropomorphisations such as *'a little goat who is afraid of eating'* (D-idea), which threatened fundamental aspects of animals' lives, as well as ideas such as *'a little oyster who couldn't find the half shell'* (LN-idea), which were found difficult to interpret. On the contrary, participants pointed out that some of the CO-ideas were *"reminiscent of existing cartoons"*, placing them into a higher rank, e.g., *'a little bird who couldn't learn to fly'* (which resembles the plot of the animated film *Rio*). These type of participant

judgements played an important role when ranking the ideas, resulting in a clear overall preference for the CO-ideas.

We also wanted to confirm the pilot study suggestion that emotional response, narrative potential and level of expectation are key components of participants' general impression of value. We used a Kendall rank correlation coefficient (τ) for this analysis. Figure 2(c) shows the average correlation results between all the components, showing a positive correlation between all the surveyed components. However, a Friedman rank sum test indicated that the particular differences between correlation values are not significant (p=0.2438), i.e., all question pairs were similarly correlated.

Figure 2(d) shows the correlation between the chaining scores and the overall rankings of the participants. We see that weak positive correlations were found for most of the aspects evaluated in the four surveys and the chaining scores. These results confirm that, as suggested in the pilot study, the chaining technique can be used as a measure to evaluate fictional ideas, and we plan to investigate the value of generating other semantic chains to increase the effectiveness of this technique. Figure 2(d) also shows that a weak negative correlation exists between participants' general impression and the chaining scores for the mixed-survey. This suggests that participants found it more difficult to decide on the rankings when the rendering of the ideas was mixed.

Finally, two facts are used for each idea generated with ConceptNet: facts that tagged words as animals with the *IsA* relation, and facts to be inverted, which use the CapableOf, Desires and LocatedNear relations. Figure 2(e) shows the results of calculating the correlation between the average participants' rankings and each ConceptNet fact score, as well as the combination of both (CB). We see that, except for the LN-survey, most of the results show a weak positive correlation. This supports the finding from the pilot study that the values people project onto ideas is somewhat in line with the score assigned by ConceptNet to the underlying facts. Moreover, the highest correlations are presented in the D-survey with the *IsA* relation. We believe that people tend to rank higher ideas associated with more common animals, such as dogs or cats, used in multiple ideas of the D-survey, than ideas involving relatively uncommon animals, such as ponies, moles or oxen, which were used in the LN-survey.

The correlations between the participants' rankings and the chaining and ConceptNet scores (Figures 2(d) and 2(e)) led us to believe that these scores could be used to predict people's preferences when ranking fictional ideas. To test this hypothesis, we used the Weka machine learning framework (Hall et al. 2009). We provided Weka with the scores of: ConceptNet chaining, ConceptNet strength for the IsA relation, ConceptNet strength for the inverted relations, word frequencies for the LHS and RHS of inverted facts, and semantic similarity between the LHS and RHS of inverted facts, obtained using the DISCO system[7]. We classified each idea into *good* (top 5), *bad* (bottom 5) or *medium* (middle 5) based on the average participants' rankings. We tested a variety of decision tree, rule-based and other learning mechanisms, with the results given in Table 5, along with the name of the learning method which produced the best classifier. We found that the RandomTrees approach consistently performed well, but was only the best method for two aspects of evaluation. We used Weka to perform a Paired T-Test, which showed that the predictors are significantly better than the majority class classifier (MCC) – which simply assigns the largest class as a prediction – with up to 95% confidence.

[7] www.linguatools.de/disco/disco_en.html

	MCC	GI	ER	NP	LE
Method	ZeroR	Ridor	RandTree	NBTree	RandTree
Accuracy(%)	35.08	49.12	56.14	43.85	54.38

Table 5: Predictive accuracy for general impression, emotional response, narrative potential and level of expectation. Note that MCC value was the same for all evaluated aspects, i.e., GI, ER, NP and LE.

Conclusions and Future Work

While essential to the simulation of creative behaviour in software, fictional ideation has barely been studied in Computational Creativity research. Within the WHIM project, we have implemented three approaches to automated fictional ideation which act as a baseline to compare future ideation methods against. We presented baseline methodologies for assessment, in the form of a curation analysis and a crowd-sourcing study where participants ranked fictional ideas. The curation analysis showed that when guided in a strong context such as Disney characterisations, automated ideation methods work well, but they degrade when the context becomes weaker. The crowd sourcing study showed that an inference chaining technique – inspired by the hypothesis that ideas can be evaluated through narratives involving them – provides a reliable measure of value with which to assess the quality of fictional ideas. Also, we found positive correlations between the rankings of general impression and each of emotional response, narrative potential and expectation, showing that these are key elements of participants' general impression of fictional ideas. Finally, we demonstrated that machine learning techniques can be used to predict how people react to a fictional idea along these axes, albeit with only around 50% predictive accuracy.

The baselines presented here provide a firm foundation on which to build more intelligent ideation methods. We plan to improve open information extraction techniques for web mining, and to investigate ideation techniques involving metaphor and joke generation methods and the subversion of category expectations. Also, we plan to use extrapolation to explore scenarios that arise from a fictional idea. For instance, from the seed idea *What if there was an elevator with a million buttons?* we could extrapolate the distance the elevator can reach and come up with a scenario in which elevators can reach as high as space. Identifying that the current distance reached by elevators is significantly lower than the distance to space is crucial in order to select this idea as an interesting scenario. Using quantitative information can help achieve this goal. The Visuo system (Gagné and Davies 2013) uses semantic similarity to estimate quantitative information for input descriptions of scenes by transferring quantitative knowledge to concepts from distributions of familiar

concepts in memory. We will explore the use of Visuo in the production of scenarios from a fictional idea.

The generation and assessment of narratives will be a key factor, enabling the system to curate its output. We will derive a theory of idea-centric narratives and implement methods for generating them and assessing ideas in terms of the quality/quantity of narratives they appear in. Our ConceptNet chaining technique shows much promise. Based on the correlation found between general impression and emotional response, we plan to improve the predictive power of the technique using sentiment analysis, as in (Liu, Lieberman, and Selker 2003), where the affect of a concept is assessed through a chaining process. The final major aspects will be to experiment with rendering methods where obfuscation and affect are used to increase audience appreciation of an idea; and the machine learning of a detailed audience model which will influence the entire ideation process.

The WHIM project is primarily an engineering effort to build a What-if Machine as a web service and interactive engine, which generates fictional ideas, and provides motivations and consequences for each idea, potential narratives involving it, and related renderings such as poems, jokes, neologisms and short stories. The first version of the What-if Machine is available online[8], and uses Flowchart E from figure 1. Users can parametrise the method for exploration, or simply click the *'I'm feeling lucky'* button. This online implementation will be used to gather feedback for audience modelling, and hopefully help promote fictional ideation as a major new area for Computational Creativity research.

Acknowledgements

We would like to thank the members of the Computational Creativity Group at Goldsmiths for their feedback, Jasmina Smailović for preprocessing the tweets used in the bisociative approach, the participants of the crowd sourcing study for their time, and the anonymous reviewers for their constructive comments. This research was funded by the Slovene Research Agency and supported through EC funding for the project WHIM 611560 by FP7, the ICT theme, and the Future Emerging Technologies FET programme.

References

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the International Conference on Management of Data*.

Charnley, J.; Colton, S.; and Llano, M. T. 2014. The FloWr framework: Automated flowchart construction, optimisation and alteration for creative systems. In *Proceedings of the 5th International Conference on Computational Creativity*.

Colton, S., and Wiggins, G. 2012. Computational Creativity: The final frontier? In *Proceedings of the 20th European Conference on Aritificial Intelligence*.

Fader, A.; Soderland, S.; and Etzioni, O. 2011. Identifying relations for open information extraction. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*.

Fauconnier, G., and Turner, M. 2008. *The way we think: Conceptual blending and the mind's hidden complexities*. Basic Books.

Gagné, J., and Davies, J. 2013. Visuo: A model of visuospatial instantiation of quantitative magnitudes. *The Knowledge Engineering Review* 28(3):347–366.

Goel, A. K. 2013. Biologically inspired design: A new program for computational sustainability. *IEEE Intelligent Systems* 28(3):80–84.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA data mining software: an update. *SIGKDD Explorations* 11(1):10–18.

Juršič, M.; Cestnik, B.; Urbančič, T.; and Lavrač, N. 2012. Cross-domain literature mining: Finding bridging concepts with crossbee. In *Proceedings of the 3rd International Conference on Computational Creativity*.

Koestler, A. 1964. *The act of creation*, volume 13. Hutchinson & Co.

Lin, T.; Mausam; and Etzioni, O. 2012. Entity linking at web scale. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*.

Liu, H., and Singh, P. 2004. Commonsense reasoning in and over natural language. In *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering Systems*. Springer.

Liu, H.; Lieberman, H.; and Selker, T. 2003. A model of textual affect sensing using real-world knowledge. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*.

Llano, M.; Hepworth, R.; Colton, S.; Charnley, J.; and Gow, J. 2014. Automating fictional ideation using ConceptNet. In *Proceedings of the AISB14 Symposium on Computational Creativity*.

Martins, J.; Pereira, F.; Miranda, E.; and Cardoso, A. 2004. Enhancing sound design with conceptual blending of sound descriptors. In *Proceedings of the 1st Joint Workshop on Computational Creativity*.

Moorman, K., and Ram, A. 1996. The role of ontology in creative understanding. In *Proceedings of the 18th Annual Conference of the Cognitive Science Society*.

Pereira, F., and Cardoso, A. 2003. The horse-bird creature generation experiment. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behavior* 1(3):257–280.

Pereira, F., and Gervás, P. 2003. Natural language generation from concept blends. In *Proceedings of the AISB Symposium on AI and Creativity in Arts and Science*.

Pereira, F. 2007. *Creativity and AI: A Conceptual Blending Approach*. Mouton de Gruyter.

Perovšek, M.; Cestnik, B.; Urbančič, T.; Colton, S.; and Lavrač, N. 2013. Towards narrative ideation via cross-context link discovery using banded matrices. In *Proceedings of Advances in Intelligent Data Analysis XII*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.

Veale, T. 2006. Tracking the lexical zeitgeist with WordNet and Wikipedia. In *Proceedings of 17th European Conference on Artificial Intelligence*.

Yao, L.; Riedel, S.; and McCallum, A. 2012. Probabilistic databases of universal schema. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*.

[8] www.whim-project.eu/whatifmachine

The Three Layers Evaluation Model for Computer-Generated Plots

Rafael Pérez y Pérez
Departamento de Tecnologías de la Información
Universidad Autónoma Metropolitana, Cuajimalpa
Av. Vasco de Quiroga 4871, Col. Santa Fe Cuajimalpa, México D. F., C.P. 05300
rperez@correo.cua.uam.mx, www.rafaelperezyperez.com

Abstract

This paper describes a model for evaluating a computer-generated plot**.** The main motivation of this project is to provide MEXICA, our plot generator, with the capacity of evaluating its own outputs as well as assessing narratives generated by other agents that can be employed to enrich its knowledge base. We present a description of our computer model as well as an explanation of our first prototype. Then, we show the results of assessing three computer-generated narratives. The outcome suggests that we are in the right direction, although much more work is required.

Introduction

The engagement-reflection (ER) computer model of writing (Pérez y Pérez and Sharples 2001) represents creativity as a constant interplay between the generation of ideas and their evaluation. As a core characteristic, such processes strongly interact and influence each other. Thus, from the ER perspective, assessment is an integral part of the creative process. In the same way, evaluation plays an essential role after the creative process has ended: i.e. following a particular criterion, it provides elements to establish the value of an agent's output. In this way, we can distinguish two different goals for the same process: 1) to contribute to the development of a story in progress; 2) to estimate if the system's output might be classified as creative. The work reported in this paper concentrates in the latter. From now onwards, we refer to a computer agent that is capable of assessing a product as evaluator. The main motivation of this project is to provide MEXICA, our plot generator, with the capacity of evaluating its own outputs as well as assessing narratives generated by other agents that can be employed to enrich its knowledge base. We can summarise it as follows: MEXICA = plot generator + evaluator.

What are the elements that need to be considered in a computer model of evaluation? In this work we present three. The following lines describe each of them.

1) A creative process generates at least two types of outputs: a final product (e.g. a solution to a problem, a poem, a story, a piece of music) and novel knowledge that expands the expertise of the creator. It is not possible to think of creativity without these two elements. Sometimes, authors engage in creative tasks with the main purpose of expanding their expertise in particular topics. For example, Picasso developed several sketches in preparation to paint El Guernica. Based on these observations, we claim that computerised creativity (c-creativity) occurs when as a result of the creative process an agent generates knowledge that does not explicitly exist in its original knowledge-base and which plays an important role in the produced output (Pérez y Pérez and Sharples 2004); such novel knowledge becomes available within the agent's knowledge base for the generation of more original outputs (Pérez y Pérez under revision). That is, an essential aim of creativity is the generation of expertise and experience that is useful for the creative process itself. We believe that the same principle can be applied during the assessment of a narrative. A computer model of evaluation must consider if the evaluator, as a result of the assessment process, incorporates new knowledge structures into its knowledge base. This idea seems to echo the thoughts of some writers about the importance of reading. For instance, David Lodge claims that reading other authors is the best way to learn about the world and about the technical abilities required for writing (Lodge 1996). Thus, a good narrative allows discovering new perspectives in a given situation, new features that had not been seen before, novel ways of understanding a situation. In other words, it generates new knowledge in the reader.

2) The second aspect to be considered is related to the concept of story. Different authors agree that a story is defined as a sequence of actions that follow the classical Aristotelian structure: setup, conflict, complication, climax and resolution (e.g. see Claude Bremond 1996; Clayton 1996, p.p. 13-15). Usually, conflict is described as obstacles that oppose a more satisfactory state or desire. During complication, the difficulties introduced by the conflict arise incrementing the tension produced in the reader, until the climax is reached. Then, all conflicts are sorted out releasing all accumulated tensions. In other words, if one follows the Aristotelian concept of a story, a narrative must produce in the reader increments and decrements of the dramatic tension. Thus, a computer model of plot evaluation must be able to recognise if the events that comprise a narrative satisfy the Aristotelian requirements. In order to achieve this goal, one needs an agent capable of representing affective responses. (It is worth pointing out that, although in this work we adopt the Aristotelian view, there are other valid options to represent narratives).

3) The third aspect considers that an agent must be able to determine if the sequence of actions that comprise a story satisfies common sense knowledge.

In sum, a computer model of plot evaluation requires a story to be evaluated, and an agent capable of transforming the sequence of actions that comprise the story into internal representations that allows detecting novel knowledge structures (cognitive changes), its coherence (common sense knowledge) and representing increments and decrements of the dramatic tension of the tale (affective responses). In the same way, it is necessary to determine how these components influence each other.
This type of model requires an agent's knowledge-base that represents the experience of the evaluator: a structure is novel when it does not previously exist in its knowledge-base; the information necessary to evaluate the coherence and the story's tension resides within this repository. Thus, different agents with different knowledge and beliefs should produce different evaluations of the same product. Even the same agent, if its knowledge base is modified, might produce different evaluations of the same product. The following lines describe a computer model for plot evaluation that subscribes to these ideas. It is built on top of the results we obtained from previous research on this topic.

Related Work

Ritchie (2007) suggests criteria for evaluating the products of a creative process (the process is not taken into consideration); in general terms such criteria evaluate how typical and how valuable the product is. The goal is, using existing evaluations of typicality (and atypicality) and value, to construct more complex criteria. Colton (2008) considers that skill, imagination and appreciation are characteristics that a computer model needs to be perceived to have (see also Pease et al. 2001). Jordanous (2012) employs a group of human experts to develop criteria for evaluation of a computer generated product. It includes characteristics like Spontaneity and Subconscious Processing, Value, Intention and Emotional Involvement, and so on. All these are interesting ideas, although some are too general and difficult to implement (e.g. see Pereira et al. 2005). Some work has been done in evaluation of plot generation. Peinado et al. (2010) also have worked in evaluation of stories, although they work was oriented to asses novelty. I am not aware of any model of plot generation that includes the characteristics of the present work.
In your review of related work, Ritchie's criteria aren't merely evaluating how typical/valuable products are, but using existing evaluations of typicality (and atypicality) and value to construct more complex criteria. Also, although Jordanous's case study example uses human expert evaluations to evaluate different criteria, she does not insist that her criteria are measured by human experts - quantitative/automated tests could also be used.

Our Plot Generator

Our research in generation and evaluation of narratives is based on the MEXICA agent (Pérez y Pérez and Sharples 2001; Pérez y Pérez 2007). We claim that, as a result of engagement-reflection cycles, our storyteller produces plots that are novel, coherent and interesting. MEXICA employs a dictionary of story-actions and a set of Previous Stories, both defined by the user as text files, to construct its knowledge base. Story-actions have associated a set of preconditions and post conditions that represent common sense knowledge. For example, the precondition of the action character A heals character B is that B is injured or ill. Otherwise, the action does not make sense.
In MEXICA, a story is defined as a sequence of actions that follows the next format: character performing the action, description of the action, object of the action (another character); for instance, the jaguar knight attacked the enemy. The format allows some variations, e.g. only one character performing an action; for instance, the princes went to the forest. We refer to this way of organising a narrative as MEXICA's format. The Previous Stories represent well-constructed narratives and provide information about how the story-world works. They represent the experience and knowledge of the agent. Any new story generated by MEXICA can be added to the Previous Stories.
The Contextual Structures are the main representation of knowledge within the system. They associate emotional links and tensions between characters with logical actions to perform. For instance, a Contextual Structure might register that when a character A is in love with a character B (an emotional link between two characters) something logical to do is that A buy flowers to B, or that A serenades B, and so on. Contextual Structures are built from the set of Previous Stories; later, they are employed to generated new outputs during plot generation. Employing the same process, knowledge structures can be built from any new story created by the system or by any other agent (as long as the story follows the MEXICA's format).
Tensions represent conflicts between characters. When the number of conflicts grows the value of the tension rises; when the number of conflicts decreases the value of the tensions goes down; when the tension is equal to zero all conflicts have been solved. Thus, the storyteller keeps a record of the dramatic tension in the story. The following are examples of situations that trigger tensions: when the life of a character is at risk; when the health of a character is at risk; when a character is made a prisoner; and so on. Every tension is assigned a value. So, each time an action is performed by a character the system calculates and records the value of all active tensions. With this information the storyteller is able to graph the curve of tension of the story. Such a curve is referred to as the Tensional Representation.

Description of the Model

The work reported in this paper employs and extends the results we obtained in previous efforts to understand automatic plot evaluation. The approach we have followed is to break this complex problem into relatively simpler sub problems. Thus, we developed a computer model for assessing novelty (Pérez y Pérez et al. 2011) and a computer model for assessing interestingness (Pérez y Pérez and Ortiz 2013) as first steps before building

the integral model of evaluation (we did not publish the result of our model for assessing coherence) . Based on those results, we came out with a general model that I present here. The following lines provide a general view of this work.

We exploit the infrastructure built for MEXICA. Thus, a dictionary of story-actions and a set of Previous Stories, both defined by the user as text files, are used to construct the evaluator's knowledge base. It is interesting to notice that our agent employs the same information to generate a plot and to evaluate a plot.

We have been successful in developing tools that are capable of transforming a sequence of actions (i.e. a story in MEXICA's text format) into internal structures that our computer agent can manipulate. Employing such tools, it is possible to perform an analysis of the dramatic tension of the story under evaluation and of the changes that such a plot produces into the agent's knowledge structures. I refer to the process of transforming a sequence of actions into structures that represent knowledge and affective reactions as *Interpretation* (see figure 1).

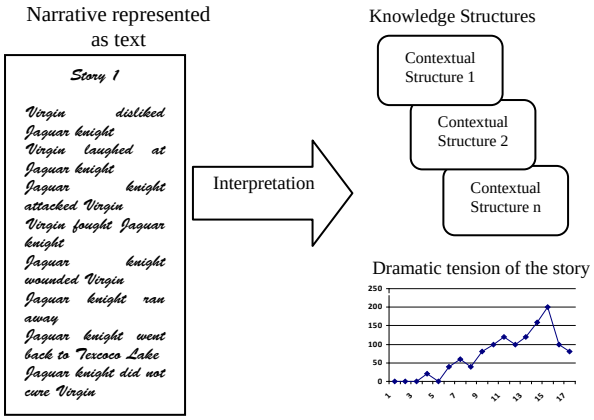


Figure 1. The Interpretation Process transforms a sequence of actions in a text format into a set of knowledge structures and affective reactions (dramatic tension).

Once the interpretation has been performed the agent has the necessary information to analyse the attributes of the story under assessment. Based on our previous work, we have selected a set of eight features, known as the story-characteristics, which are useful for evaluating a plot: opening, closure, climax, reintroducing complications, satisfaction of preconditions, repetition of sequences of actions and two types of novel knowledge structures. Typically, they have a value ranging from zero to one, where one is the most desirable value. They represent knowledge structures and affective reactions. Details of the story-characteristics are given some lines ahead.

In order to implement the model for assessing the novelty it was necessary to choose a set of story-characteristics that were associated to the production of original plots; the same applies for the model of evaluation of interestingness and coherence. Some story-characteristics are used in more than one of those systems. For instance, sorting out all the problems that characters have at the end of the story (correct closure of the narrative) is important for both, the model of coherence and the model of interestingness; the generation of unusual situations (new knowledge structures) is important for the model of interestingness and the model of novelty; and so on.

It is possible to employ the three models mentioned above to obtain a global evaluation of a story. That is, given a plot, we can run the system that evaluates novelty, then the system that evaluates interestingness and lastly the system that evaluates coherence; finally, we can calculate the average result. However, this procedure has some flaws. As mentioned earlier, some story-characteristics are employed in more than one model. As a result, they might be overrepresented in the overall calculus distorting the final value. In the same way, story-characteristics might be linked in ways that individual models cannot represent. For instance, one story might get a high score in novelty but a low score in coherence. However, it does not make sense to claim that a story is very original when it is unintelligible. A famous example of a similar situation is the sentence "Colorless green ideas sleep furiously" (Chomsky, 1957); this sentence does not seem mean anything coherent but sound like an English sentence. Thus, it seems sensible to have one model for a general evaluation, where all story-characteristics can interact, rather than three individual ones.

Some of the story-characteristics, although useful, are not essential for a good plot. So, if they are present they help to enhance the story; if not, the story still can be a good narrative. We referred to such characteristics as Enhancers. For instance, if the problems of a character seem to be solved and out of the blue new conflicts arise (reintroducing complications) the plot might be considered as more exciting. This characteristic is not required to develop a good plot but its presence helps. So, Enhancers add extra points to the evaluation. The use of Enhancers might be conditioned to the good results of other characteristics. For instance, if a given story is unoriginal it does not make sense to consider it more interesting only because there is a reintroduction of complications. Following the same logic, the model contemplates the use of Debasers, i.e. story-characteristics that, when they are missing, they decrement in some points the global evaluation of a plot.

In our previous models the relationships of the story-characteristics are defined by expressions like the following:

$$E = C1W1 + C2W2 + C3W3+ \ldots CnWn$$

where E represents the result of the evaluation, C one of the characteristics to be assessed and W its weight. However, this expression lacks flexibility. For example, it is not possible to represent conditioned Enhancers or Debasers. In the same way, some characteristics might play a more relevant role during one stage of the assessment than during others. For example, a story must be lucid; otherwise, it is not worth evaluating the plot. So, at this point those characteristics associated to coherence have a high priority for the evaluation process. However, once this requirement is satisfied, other characteristics start to

take precedence. To illustrate this situation the reader can picture a logic story that is boring, i.e. it lacks increments and decrements of tension. In this case, those characteristics associated to interestingness became more relevant for the evaluation process. As a result, the global assessment probably would produce a low value even if the coherence is pretty good.
The model also considers what we refer to as the compensation effect. In the overall evaluation, characteristics highly rated might compensate those with lower grades by adjusting their weights. For example, picture a story that shows exceptional original situations; even if the plot suffers for some coherence problems, the overall rate might still be pretty high.

Description of the Story-Characteristics

The following lines describe the story-characteristics that I employ in this work and how to calculate their values.

Opening: We consider that a story has a correct opening when at the beginning there are no active dramatic tensions in the tale and then the tension starts to grow. If at the beginning of the story the value of the tension is zero, then Opening is set to one; if at the beginning of the story the value of the tension is equal to the main peak (the climax), then Opening is set to zero; otherwise, Opening is set to a proportional value between zero and one.

Opening = 1 – (Tension at the first action /Peak)

Closure: We consider that a story has a correct closure if all the dramatic tensions in the story are solved when the last action is performed. That is, following Pérez y Pérez and Sharples, a story "should display an overall integrity and closure, for example with a problem posed in an early part of the text being resolved by the conclusion" (Pérez y Pérez and Sharples 2004). If at the end of the story the value of the tension is equal to the main peak (the climax) then Closure is set to zero; If at the end of the story the value of the tension is equal to zero (all problems are solved), then Closure is set to 1; otherwise, Closure is set to a proportional value between zero and one.

Closure = 1 – (Tension at last action/Peak)

Climax: All stories should include a climax. In the graphic of tensions the climax is represented by highest peak. However, it is not the same a story with an incipient peak that a story with a clear elevated crest. In order to evaluate the peak, MEXICA calculates the average value of all Previous Stories' climax and employs it as a reference. Thus, if the peak's value is equal or major than the reference, then Climax is set to 1; if there is no peak, then Climax is set to zero; otherwise, it is set to a proportional value between zero and one.

Climax = (Current climax/Reference value climax)
If Climax > 1 then Climax = 1

Reintroducing Complications: We refer to the situation where a narrative has a resolution and then tensions start to rise again as reintroducing-complications. In this work, we appreciate narratives that seem to end and then new problems for the characters emerge, i.e. where all tensions are solved and then they rise again. This formula can be observed in several examples of narratives like films, television-series and novels. MEXICA calculates the average number of complications that are reintroduced in the Previous Stories and employs it as a reference. Thus, if the number of times that the current story reintroduce complications is equal or major than the reference, then Reintroducing Complications is set to 1; if there is no reintroduction of complications, then Reintroducing Complications is set to zero; otherwise, it is set to a proportional value between zero and one.

Novel Contextual Structures: In this work a new story generates new knowledge when it generates structures that did not exist previously in the knowledge base of the system and that can be employed to build novel narratives. Each action within a plot has the potential of introducing an unknown context for the agent. So, if all actions that comprise the story under evaluation generate unknown contexts, then Novel Contextual Structures is set to one; if none of the actions produce an unknown context, then Novel Contextual Structures is set to zero; otherwise, Novel Contextual Structures is set to a proportional value between zero and one.

Original Value: Besides calculating the number of novel contextual structures, it is necessary to determine how original they are with respect to the information that already exists in the knowledge base. With this purpose we define a parameter known as the Limit of Similitude (LS) that represents the maximum percentage of alikeness allowed between two knowledge structures. If the percentage of similitude between a given Contextual Structure and all structures in the knowledge base is minor to LS, we refer to such Contextual Structure as original. In this way, we can distinguish between novel situations and really original ones. Thus, the Original Value is equal to the ratio between the total number of original structures and the total number of contexts produced by the tale.

Preconditions: All actions have associated preconditions that represent common sense knowledge. If the preconditions of all story actions are fulfilled, then Preconditions is set to one; if none of the preconditions of all story actions are fulfilled, then Preconditions is set to zero; otherwise, it is set to a proportional value between zero and one.

Repetition of Sequences: There are some attributes that contribute to the lack of coherence in a plot. The repetition of sequences of actions performed by the same characters illustrates this situation. We include this feature to show some of the problems that computer generated narratives might suffer. Thus, in this implementation, Repetition of Sequences is set to one when there are no repetitions; otherwise, it is set to zero.

The Three-Layers

The model described in this paper represents evaluation as a process organized in three layers (see figure 2).
Layer-0 includes those characteristics that a plot must satisfy in order to be considered for evaluation. These characteristics do not add points to the evaluation; they are requirements that need be satisfied in order to proceed to evaluate the plot. Otherwise, the process is ended. They are known as the *required-characteristics*.
Layer-1 includes what I refer to as the *core-characteristics*. They are the backbone of the evaluation process and represent those essential features that form a plot.
Layer-2 includes what I refer to as the *Enhancers* and the *Debasers*. Enhancers are characteristics that add extra points to the result obtained from the previous layer. Debasers represent features that decrement the result obtained from Layer-1. Their use might be conditioned to the result of other story-characteristics.

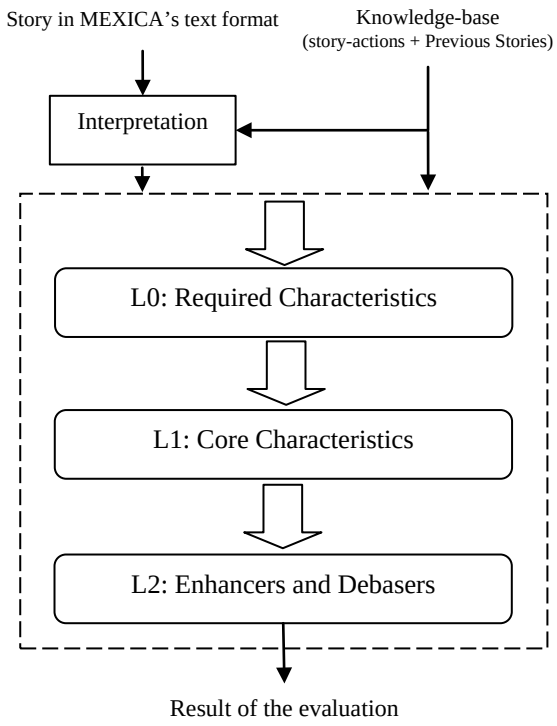


Figure 2. The three layers evaluation model.

A story-characteristic can be employed in more than one layer. Actions' preconditions illustrate this situation: it is not worth to evaluate an unintelligible story (Preconditions in Layer-0); however, a mainly sounded story with few inconsistencies might only be penalized with some negative points (Preconditions in Layer-2).

The following lines provide details about the implementation.
Layer-0: In the current implementation, the number of Fulfilled Preconditions and the number of Novel Contextual Structures are selected as the Required-Characteristics. If most actions within a story have unfulfilled preconditions or the story under evaluation is too similar to any of the previous stories, then the systems considers that is not worth evaluating the plot. The user provides the minimum rates that the story-characteristics Fulfilled Preconditions and Novel Contextual Structures must reach to continue with the evaluation process.

Layer-1: In the current implementation, the following elements have been selected as the core-characteristics: Climax, Closure and Novel Contextual Structures. All they have been assigned the same weight. These characteristics have been chosen because: a narrative without climax is not a story; Closure is important to keep the coherence and interestingness of the tale; novelty is an essential feature of any story. The result of the evaluation in Layer-1 is the average value of the three core-characteristics.
Layer-2: In the current implementation, Preconditions and Repeated Sequences have been chosen as Debasers. They represent features that we take for granted; however, if they are missing within a narrative we immediately notice them. Thus, if they have a value lower than a reference provided by the user, the result of the evaluation obtained in Layer-1 is decremented by n units, where n is a parameter defined by the user.

```
IF Preconditions < Reference-Preconditions THEN
   Decrement-Result-Evaluation-1
IF Repetition-Sequences < Reference-RS THEN
   Decrement-Result-Evaluation-1
```

The following characteristics have been chosen as Enhancers: Opening, Reintroducing Complications and Original Value. Thus, if they have a value higher than a reference provided by the user, then the result of the evaluation obtained in Layer-1 is incremented by m units, where m is a parameter defined by the user. Enhancers are only employed when there are not repetition of sequences of actions, the evaluation in Layer-1 and the Closure reach a minimum value defined by the ser.

```
IF (Repetition-sequences = 1) and (Result-Layer-1 > Reference-L1) and (Closure > Reference-Closure) THEN
BEGIN
  IF Opening > Reference- Opening THEN
     Increment-Result-Evaluation-1;
  IF Reintroducing-Complications > Reference-RC THEN
     Increment-Result-Evaluation-2;
  IF Original-Value > Reference-OV THEN
     Increment-Result-Evaluation-3;
END
```

As a final step, the evaluator generates a report to explain the criteria employed during the process of evaluation. The report is divided in four sections: section one includes a general comment about the whole narrative; section two provides observations about the story's coherence; section three incorporates notes about the story's interestingness; and section four offers comments about the narrative's novelty.
The report is generated by matching the value of some of the story-characteristics with predefined texts. In general, there are at least five possible options that can be employed for each of such story-characteristic.

```
IF Value-Story-Characteristic > 0.9 THEN
  Employ-Text-1
ELSE IF Value -Story-Characteristic > 0.8 THEN
  Employ-Text-2
ELSE IF Value -Story-Characteristic > 0.7 THEN
  Employ-Text-3
ELSE IF Value -Story-Characteristic > 0.6 THEN
  Employ-Text-4
ELSE   Employ-Text-5;
```

The following lines describe the way each section is built.

Section one. The system employs the final result of the evaluation process (output of Layer 2) to select the right text.

Section two. The coherence section includes three types of comments: one associated to the satisfaction of preconditions, one related to the right closure and the last one connected to the repetition of sequences of actions. The first two types of comments are always printed; the last type of comment is omitted when the tale does not include repeated sequences of actions. Thus, the system employs the story-characteristics Preconditions, Closure and Repetition of sequences to generate the text.

Section three. The interestingness section includes five types of comments, each one related to the following story-characteristics: Opening, Climax, Reintroducing complications, Closure and Original value. The first two comments are always included in the report while the last three comments are only printed when some requirements are satisfied. The next lines explain the conditions that need to be satisfied in order to incorporate the last three remarks into the report. If the story-characteristic Climax ≥ 0.7 then the system adds comments about the closure. This makes sense because the climax represents the conflicts in the story and the closure indicates how those conflicts are sorted out.

If the story-characteristic Closure ≥ 0.7 then comments regarding the original value are inserted in the report. That is, the system only includes comments about singular features of the plot when it has an adequate ending. That is, in the current implementation originality loses importance when the story has a bad finale.

If the story-characteristic Closure ≥ 0.7 and the Reintroduction of complications ≥ 0.75 then the system inserts some comments about the reintroduction of complications in the report. In this case, besides considering the closure, the system requires that the story includes a clear instance of the reintroduction of complications. Otherwise, it is no point to make comments about this feature.

All these parameters can be modified by the user.

Section four. The novelty section includes comments about the originality of the story. The system selects the appropriate text depending on the value of the story-characteristic Novel contextual structures.

Testing the Model

To test the model we evaluated three stories: two generated by MEXICA and one generated by another story teller.

In Layer-0 we established the following conditions to continue with the evaluation process: Preconditions > 0.7 and Novel Contextual Structures > 0.35.

In Layer-2 we established the following requirements for the Debasers:

```
IF Preconditions < 0.7 THEN
  Decrement-Result-Evaluation-in-2points;
IF Repetition-Sequences < Reference-RS THEN
  Decrement-Result-Evaluation-in-3points;
```

In Layer-2 we established the following requirements for the Enhancers:

```
IF (Repetition-sequences = 1) and (Result-Layer-1 ≥ 0.7) and
(Closure > 0.75) THEN
BEGIN
  IF Opening = 1THEN
    Increment-Result-Evaluation-in-0.5points;
  IF Reintroducing-Complications > 0.8 THEN
    Increment-Result-Evaluation-in-1point;
  IF Original-Value > 0.5 THEN
    Increment-Result-Evaluation-in-1.5points;
END
```

The values of the parameters are the result of several tests we have performed.

Story 1.

This story was developed by MEXICA-impro and reported in (Pérez y Pérez et al. 2010).

Jaguar knight is introduced in the story
Princess is introduced in the story
Hunter is introduced in the story
Hunter tried to hug and kiss Jaguar knight
Jaguar knight decided to exile Hunter
Hunter went back to Texcoco Lake
Hunter wounded Jaguar knight
Princess cured jaguar knight
Enemy kidnapped Princess
Enemy got intensely jealous of Princess
Enemy attacked Princess
Jaguar knight looked for and found Enemy
Jaguar knight had an accident
Enemy decided to sacrifice Jaguar knight
Hunter found by accident Jaguar knight
Hunter killed Jaguar knight
Hunter committed suicide

The following lines show the values of the story-characteristics:

Preconditions: 1
Opening: 1
Closure: 0.6
Climax: 1
Novel Contextual Structures: 0.71

Original Value: 0.71
Repeated Sequences: 1
Reintroducing Complications: 0
Result-Layer-1: 0.77

Figure 3 shows the graphic of tension of story 1. Because the Closure did not reach the value of 0.75 the Evaluator decided not to employ the Enhancers.

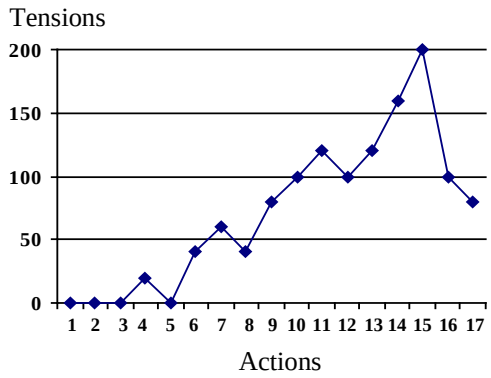


Figure 3. Tensional Representation of story 1.

The following lines produced by the agent provide the reasons of the final result:

EVALUATION OF THE STORY

This is a good effort. With more practice you will be able to create nice plots. Here are some comments about your work that I hope will be a useful feedback.

COHERENCE
The story is very logical; all actions are nicely integrated and form a coherent unit. It requires that all complications that characters faced are sorted out by the end of the last part. You need to pay more attention to this aspect.

INTERESTINGNESS
The text has a good introduction. The story reaches a nice climax with a good amount of tension. This is an important characteristic of a good narrative. Great! Sadly, the bad closure damages the interestingness of a story.

NOVELTY
The plot is kind of inventive.

My evaluation of your story is ->77/100

Story 2.
This story was produced by MEXICA for this paper.

Virgin disliked Jaguar knight
Virgin laughed at Jaguar knight
Jaguar knight attacked Virgin
Virgin fought Jaguar knight
Jaguar knight wounded Virgin
Jaguar knight ran away
Jaguar knight went back to Texcoco Lake
Jaguar knight did not cure Virgin
Tlatoani was an inhabitant of the Great Tenochtitlán
Tlatoani and Jaguar knight were rivals
Tlatoani fought Jaguar knight
Jaguar knight ran away
Jaguar knight went back to Texcoco Lake
Jaguar knight did not cure Virgin

The following lines show the values of the story-characteristics:

Preconditions: 1
Opening: 0.8
Closure: 0.28
Climax: 1
Novel Contextual Structures: 0.86
Original Value: 0.86
Repeated Sequences: 0
Reintroducing Complications: 1
Result-Layer-1: 0.71

Figure 4 shows the graphic of tension of story 2. The story has a really bad Closure; however, the good Climax and the relatively good result of Contextual Novel Structures push the result in Layer-1. However, Repeated Sequences are highly punished (the succession of actions 6, 7 and 8 is repeated at the end of the tale) and therefore the evaluator decrements in 3 point the final result.

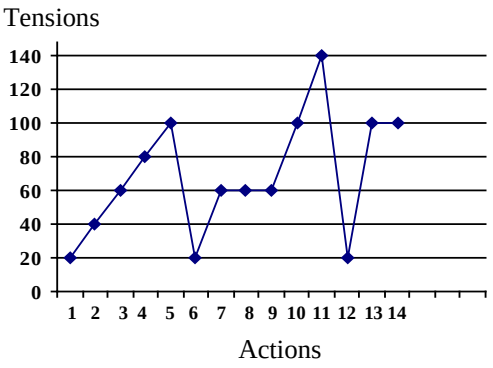


Figure 4. Tensional Representation of story 2.

The following lines show the report explaining the evaluation process.

EVALUATION OF THE STORY

Sorry, but this story is not good.

Here are some comments about your work that I hope will be a useful feedback.

COHERENCE
The story is very logical; all actions are nicely integrated and form a coherent unit.
Unfortunately, there are several loose ends that need to be worked out (it reminds me of the really bad end of the TV show “Lost”). As a result the plot lacks an adequate conclusion, an important characteristic of a good narrative. You are repeating sequences of actions; as a consequence the plot is confusing!

INTERESTINGNESS
The plot starts with some tension. The story reaches a nice climax with a good amount of tension. This is an important characteristic of a good narrative. Great! Sadly, the bad closure damages the interestingness of a story.
NOVELTY
I find this story pretty original! I love it!

My evaluation of your story is ->41/100

Notice the last sentence in the report. Because the Original Value got a high rate the evaluator includes this sentence. It is necessary to correct this problem.

Story 3.
This story was produced by MINSTREL (Turner 1993, p. 622). The original tale narrates the story of a knight, known as Lancelot, how was hot tempered. Andrea was a lady of the court and one day she went to the woods to pick berries. By accident, Lancelot found Andrea in the woods and he fell in love with her. Sometime later, Lancelot found again Andrea in the woods, and he saw that she was kissing another knight known as Frederik. So, Lancelot thought Andrea was in love with Frederik and got really jealous; so, he killed Frederik. Andrea told Lancelot that Frederik was her brother. Lancelot hated himself and became and hermit; Frederik was buried in the woods and Andrea became a nun. In the following lines we show the same narrative but as a MEXICA plot:

Lady and Eagle Knight were brothers
Lady went to Chapultepec Forest
Jaguar knight found by accident Lady
Jaguar knight was very impressed by Lady
Jaguar knight fell in love Lady
Lady went to Tlatelolco Market with Eagle Knight
Jaguar knight found by accident Lady
Jaguar knight got intensely jealous of Eagle knight
Jaguar knight attacked Eagle knight
Jaguar knight killed Eagle knight
Jaguar knight realised that Lady and Eagle Knight were brothers
Jaguar knight hated Jaguar Knight
Jaguar knight exiled Jaguar knight

We transformed this narrative by trying to find similar actions in MEXICA's dictionary to those described in the original tale. The following lines show the values of the story-characteristics:

Preconditions: 1
Opening: 1
Closure: 0.75
Climax: 0.8
Novel Contextual Structures: 0.54
Original Value (surprise): 0.54
Repeated Sequences: 1
Reintroducing Complications: 0
Result-Layer-1: 0.70

Figure 5 shows the graphic of tension of story 3. In this case, it is possible to employ the Enhancers and as a result the evaluation reaches the value 0.9. This happens because the opening and the original value contribute with two points.

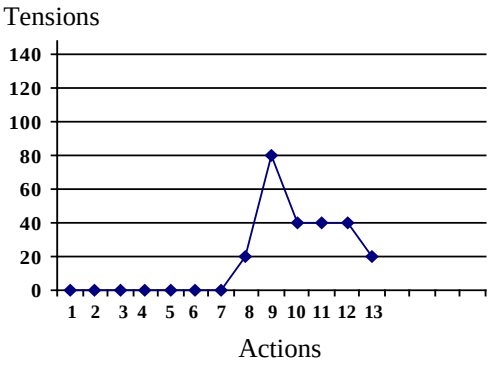


Figure 5. Tensional Representation of story 3.

The following lines show the report explaining the evaluation process.

EVALUATION OF THE STORY

This is a good story. Great! Soon you will become a real writer. Here are some comments about your work that I hope will be a useful feedback.

COHERENCE
The story is very logical; all actions are nicely integrated and form a coherent unit. At the end there are still some tensions that are not solved; it would help to the coherence and interest of the narrative if characters worked them out by the conclusion. I recommend you to avoid repeating actions (e.g. Jaguar knight Found by accident the Lady).

INTERESTINGNESS
The text has a good introduction. The climax of the story is good, although for my taste I would prefer a little extra tension. A better end would contribute to have a more interesting tale. There are surprising events that make the story appealing. I enjoyed that!

NOVELTY
The plot is kind of inventive.

My evaluation of your story is ->90/100

Discussion and Conclusions

This paper reports a computer model for plot evaluation. The model is based on the idea that affective reactions and the generation of new knowledge are important characteristics of plot evaluation. It requires a story and a process that allows transforming a sequence of actions into structures that the agent can manage. In this way, it is possible to evaluate any story produced by any agent, as long as the narrative fulfils the constraints of the format.
I refer to the process of transforming a sequence of actions into structures that represent knowledge and affective reactions as Interpretation. This work shows the im-

portance of interpretation and its role during evaluation. If a group of agents share similar interpretations, and similar knowledge structures and beliefs (knowledge bases), they probably will produce similar evaluations. Otherwise, they will generate different outputs, maybe even contradictory ones.

The three layers provide a flexible way to work with the story-characteristics. It allows giving different weights to some features during one stage of the assessment than during others; employing what we refer to as the compensation effect; conditioning the use of the Enhancers and Debasers; and so on.

The work reported in this paper is based on an Aristotelian view of what a story is. Under this framework, the model proposes a way to understand how the evaluation process might work. However, it is well known that there are other valid approaches to build, and therefore to assess, interesting narratives. Unfortunately, it is not possible yet to develop a model that comprises all of them. Evaluation is a very complex task and we are far to understand it. So, it makes sense to develop achievable programs and then start to build on top them. Hopefully, in few years we will be able to incorporate different approaches in our system.

In the current model there are several aspects that need to be revised. For instance, it is necessary to represent features like suspense, flashbacks, and so on. Similarly, it is necessary to incorporate mechanisms that allow the system to manipulate in more creative ways the structures that are already represented; e.g. we would like to provide the evaluator with the capacity of explicitly leaving unsolved conflicts as part of an interesting closure within a narrative (when this resource is properly employed it has very positive effects on the reader). So, there is much work left to be done.

Some colleagues seem to be concerned about some characteristics of this work. Their main objection has to do with the fact that "The implementation of the used metrics is based on features certainly not present in all plot generation systems" (anonymous reviewer). There is a misunderstanding here. Our model evaluates plots; we do not necessarily care about the characteristics of the storyteller. That is, the system assesses the features present in the narrative, not in the program that generated it. So, we do not see a problem here. Nevertheless, clearly this research has been developed around our storyteller.

The main goal of this project is to provide MEXICA with the capacity of evaluating its own outputs. As explained earlier, the system can also evaluate a plot produced by any other agent as long as it is represented as text with the following format: character performing the action, description of the action, object of the action (another character). (It is also necessary that all story actions employed in the plot are declared in the dictionary of the system). That is the scope of our model.

It is necessary to consider that some plot-generators might produce outputs in the MEXICA's format that include features that cannot be interpreted by our system and therefore cannot be included as part of the assessment (e.g. suspense). So, in these cases the evaluation performed by our model might be considered as incomplete.

Can this model be employed in other domains? We believe that the answer is yes. The model requires a product to be evaluated and a way to interpret such a product, i.e. a mechanism to perceive its relevant characteristics. The three layers provide a flexible method to organise and analyse such characteristics. As a result of the evaluation process the agent incorporates new structures into its knowledge base and represents affective responses. We believe that all these essential features of our model apply in other areas like, for instance, visual composition. Hopefully, this document will encourage some researchers to test the model in novel areas.

Acknowledgements

This research was sponsored by the National Council of Science and Technology in México (CONACYT), project number: 181561.

References

Bremond, C. 1996. 'La lógica de los posibles narrativos' (trad.) In *Análisis Estructural del Relato*, pp. 99-121. México, D.F: Ediciones Coyoacán.

Chomsky, N. 1957. *Syntactic Structures*. The Hague: Mouton.

Clayton , J. J. 1996. 'Introduction: on fiction'. In *The heath introduction to fiction*, pp. 1-32. USA: D.C. Heath and company.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. Creative Intelligent Systems: Papers from the AAAI Spring Symposium. 14–20.

Deckers, L. 2005. Motivation Biological, Psychological, and Enviromental. Pearson.

Jordanous, A. 2012. A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. Cognitive Computation, 4(3): 246-279

Lodge, D. 1996. The practice of writing: essays, lectures, reviews and a diary. London: Secker & Warbug.

Pease, A.; Winterstein, D.; and Colton, S. 2001. Evaluating machine creativity. In Weber, R. and von Wangenheim, C. G., eds., *Case-based reasoning: Papers from the workshop programme at ICCBR 01Vancouver*. Canada 129–137.

Peinado, F.; Francisco, V.; Hervás R. and Gervás, P. 2010. Assessing the Novelty of Computer-Generated Narratives Using Empirical Metrics. *Minds and Machines*. 20(4):565-588.

Pereira, F. C.; Mendes, M.; Gervás, P., and Cardoso, A. 2005. Experiments with assessment of creative systems: An application of Ritchie's criteria. In Gervás, P. Veale, T. and Pease, A., eds., *Proceedings of the workshop on computational creativity*, 19th international joint conference on artificial intelligence.

Pérez y Pérez, R. (under review). Computer-based Model for Collaborative Narrative Generation.

Pérez y Pérez, R. 2007. Employing Emotions to Drive Plot Generation in a Computer-Based Storyteller. Cognitive Systems Research 8(2): 89-109.

Pérez y Pérez R. and Ortiz, O. 2013. A Model for Evaluating Interestingness in a Computer–Generated Plot. In *Proceedings of the Fourth International Conference on Computational Creativity*, Sydney, Australia, pp.131-138.

Perez y Perez, R., Ortiz, O., Luna, W. A., Negrete, S., Peñaloza, E., Castellanos, V., and Ávila, R. 2011. A System for Evaluating Novelty in Computer Generated Narratives. In *Proceedings of the Second International Conference on Computational Creativity*, México City, México, pp. 63-68.

Perez y Perez, R., Negrete, S., Peñaloza, E., Castellanos, V., Ávila, R. and Lemaitre, C. 2010. MEXICA-Impro: A Computational Model for Narrative Improvisation. In *Proceedings of the international conference on computational creativity*, Lisbon, Portugal, pp. 90-99.

Pérez y Pérez, R. and Sharples, M. 2004. Three Computer-Based Models of Storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge Based Systems Journal*. 17(1):15-2.

Pérez y Pérez, R. and Sharples, M. 2001 MEXICA: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence* 13(2):119-139.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:76–99.

Turner, S. R. 1993. *MINSTREL: A computer model of creativity and storytelling*, PhD Dissertation, University of California LA, 1993

Poetic Machine: Computational Creativity for Automatic Poetry Generation in Bengali

Amitava Das
Department of Computer Science and Engineering
University of North Texas
Denton, Texas, USA
amitava.santu@gmail.com

Björn Gambäck
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
gamback@idi.ntnu.no

Abstract

The paper reports an initial study on computational poetry generation for Bengali. Bengali is a morpho-syntactically rich language and partially phonemic. The poetry generation task has been defined as a follow-up rhythmic sequence generation based on user input. The design process involves rhythm understanding from the given input and follow-up rhyme generation by leveraging syllable/phonetic mapping and natural language generation techniques.

A syllabification engine based on grapheme-to-phoneme mapping has been developed in order to understand the given input rhyme. A Support Vector Machine-based classifier then predicts the follow-up syllable/phonetic pattern for the generation and candidate words are chosen automatically, based on the syllable pattern. The final rhythmic poetical follow-up sentence is generated through n-gram matching with weight-based aggregation. The quality of the automatically generated rhymes has been evaluated according to three criteria: poeticness, grammaticality, and meaningfulness.

Introduction

Cognitive abilities can be divided into three broad categories: intelligence, aesthetics, and creativity. Suppose someone has read a sonnet by Shakespeare and is asked the following questions:

- *Do you understand the meaning of this sonnet?*
  If the reader says yes, s/he has used her/his intelligence together with knowledge of the English language and world knowledge to understand it.
- *Do you like this sonnet?*
  Whatever is answer, the reader is using a subjective model of liking — and this is what is called aesthetic appreciation or sentiment.
- *Can you add two more lines to this sonnet?*
  So the reader has to write some poetry — and has to use her/his creative ability to do it.

Artificial Intelligence is a now six-to-seven decades matured research field. The majority of the research efforts until now have concentrated on the understanding of natural phenomena. During the latest two decades, we have witnessed a huge rise of research attention towards affect understanding, that is, the second level of cognition. However, there have so far been pretty few attempts towards making machines truly creative. The paradigm of computational creativity is actually still in infancy, and most of those efforts that have been carried out have concentrated on music or art. Still, computer systems have already made some novel and creative contributions in the fields of mathematical number theory (Colton 2005; Colton, Bundy, and Walsh 2000) and in chess opening theory (Kaufman 2012).

In this paper, in contrast, we look at computational linguistic creativity, and in particular poetry generation. Computational linguistic creativity has only in the last few years received more wide-spread interest by language technology researchers. A book on linguistics creativity was recently written by Veale (2012), and in particular the research group at Helsinki University is very active in this domain (Toivanen et al. 2012; Gross et al. 2012; Toivanen, Toivonen, and Valitutti 2013; Toivanen, Järvisalo, and Toivonen 2013). Some other interesting research attempts have also been made (Levy 2001; Colton, Goodwin, and Veale 2012, e.g.,), but the approaches still vary widely.

The field of automatic poetry generation was pioneered by Bailey (1974), although Funkhouser (2009) quotes work going back to the 1950s. These systems were written by actual poets who were keen to explore the potential of using computers in writing poetry and were not fully autonomous. Thereafter, Gervás and his colleagues were the first to discuss sophisticated approaches to automatic poetry generation (Gervás 2000; 2001a; 2001b; 2002a; 2002b; Díaz-Agudo, Gervás, and González-Calero 2002; Gervás et al. 2007). Gervás' work established the possibility of automatic poetry generation and has in the last decade been followed by a moderate number of attempts at linguistics creativity and in particular at automatic poetry generation.

The system developed by Manurung (2004) uses a grammar-driven formulation to generate metrically constrained poetry out of a given topic. In addition

to the scientific novelty, the work defined the fundamental evaluation criteria of automatic poetry generation: meaningfulness, grammaticality, and poeticness. A complete poetry generation system must generate texts that adhere to all these three properties. An alternative approach to evaluation would be to adopt the criteria specified by Ritchie (2007; 2001) for assessing the novelty and quality of creative systems in general based on their output.

All these previous efforts were inspiration points for the present work, but as we are unable to conclude what method performs best, we decided to propose a new architecture by following the rules and practices of Bengali poems and writings. There is no previous similar work in Bengali, nor on other Indian languages, except attempts at automatic analysis and generation of Sanskrit Vedas (Mishra 2010) and at automatic Tamil lyric generation (Ramakrishnan A, Kuppan, and Devi 2009; Ramakrishnan A and Devi 2010).

The basic strategy adopted here is not to try to make the system create poetry on its own, but rather in *collaboration* with the user. And not a complete poem, but rather *one poetry line* at a time. The user enters a line of poetry and the system generates a matching, rhyming line. This task then in turn involves two subtasks: rhyme understanding and rhyme generation. Rhyme understanding entails parsing the input line to understand its poetic structure. Rhyme generation is based on the usage of a Bengali syllabification engine and a Support Vector Machine (SVM) based classifier for predicting the structure of the output sentence and candidate word generation, combined with bigram pruning and weighted aggregation for the selection of the actual words to be used in the generated rhyming line.

The rest of the paper is laid out as follows: To give an understanding of the background, we first discuss the Bengali language as such and the different rhythms and metres that are used in Bengali poems. Thereafter the discussion turns to the chosen methods for poetry line understanding and generation, starting by giving details of a corpus of poems collected for rhyme understanding, and then in turn describing the rhyme understanding and the rhyme generation tasks, and their respective subparts. Finally, an evaluation of the poetry generation model is given, in terms of the three dimensions poeticness, grammaticality, and meaningfulness.

Bengali and Bengali Poetry

Bengali (ethnonym: Bangla) is the seventh largest (in terms of speakers) language worldwide. It originates from Sanskrit and belongs to the modern Indo-Aryan language family. Bengali is the second largest language in India and the national language of Bangladesh. Bengali poetry has a vibrant history since the 10^{th} century and the modern Bengali poetry inherited its basic ground from Sanskrit. As the first non-European Nobel Literature Laureate and known mainly for his poems, Rabindranath Tagore (1861–1941) was the pioneer who founded the firm basis of modern Bengali poetry.

Bengali Orthography and Syllable Patterns

Bengali, just as all Modern Indo-Aryan languages being derived from Sanskrit, is partially phonemic. That is, its pronunciation style depends not only on orthographic information, but also on Part-of-Speech (POS) information and semantics. Partially phonemic languages use writing systems that are in between strictly phonemic and non-phonemic. Bengali — and many other modern Indo-Aryan languages — still uses Sanskrit orthography, although the sounds and the pronunciation rules have changed to varying degrees.

The modern Bengali script contains the characters (known as *akṣara*) for seven vowels (/i/ /u/, /e/, /o/, /æ/, /ɔ/, /a/), four semi-vowels, (/j/, /w/, /e̯/, /o̯/), and thirty consonants. Many diphthongs are possible, although they must always contain one semi-vowel, but only two of the diphthongs are represented directly in the script (i.e., have their own *akṣara*: /oi/and /ou/). All vowels can be nasalized (written as /ā/, etc.) and vowel deletion (e.g., schwa deletion) is common, particularly in word medial and final positions.

A phonetic group of Bengali consonants is called a *borgo* (বর্গ). As we shall see below, these groups are particularly important in poetic rhymes. There are five basic *borgos* in Bengali and four separate pronunciation groups, as shown in Table 1, where each consonant is displayed together with its pronunciation in the International Phonetic Alphabet (IPA). Many consonant sounds can be either unaspirated or aspirated (e.g., /t̪/vs /t̪^{h}/). The first five *borgos* are named according to their first character. In each *borgo*, the first consonant takes the least stress when pronounced and the last takes the highest stress. The first member is thus called less-stressed (*alpo-prāṇ*: অল্প প্রাণ), the second to forth members are called high-stressed (*mahā-prāṇ*: মহাপ্রাণ), and the fifth and last is a nasal (*nāsik*: নাসিক্য).

Following the classification of Sarkar (1986), Bengali has 16 canonical syllable patterns, but CV (consonant-vowel) syllables constitute 54% of the whole language (Dan 1992). Patterns such as CVC, V, VC, VV, CVV, CCV, and CCVC are also reasonably frequent. For more detailed recent overviews of Bengali phonetics, we refer the reader to, for example, Sircar and Nag (2014), Barman (2011) or Kar (2009), and just take the examples below of Bengali orthography — originally devised by Chatterji (1926) — to illustrate how it has deviated from the strictly phonemic orthography of Sanskrit.

- **Consonant clusters** are often pronounced as geminates irrespective of the second consonant. Thus: bAkya /bakko/, bakSha /bɔ/kkho, bismaYa /biʃʃɔê/.
- **Single grapheme for multiple phonemes:** The vowel [e] is pronounced as either /e/or /æ/. The ambiguity cannot be resolved by the phonological context alone as the etymology is often the underlying reason. For example: eka /æk/, but megha /megh/.

***Borgo* Name**	**Consonant Members**				
ক (k)-*borgo*	ক (k)	খ (k^h)	গ (g)	ঘ (g^h)	ঙ (ŋ)
চ (tʃ)-*borgo*	চ (tʃ)	ছ ($tʃ^h$)	জ (ʤ)	ঝ ($ʤ^h$)	ঞ (n)
ট (ʈ)-*borgo*	ট (ʈ)	ঠ ($ʈ^h$)	ড (ɖ)	ঠ ($ɖ^h$)	ণ (n)
ত (t̪)-*borgo*	ত (t̪)	থ ($t̪^h$)	দ (d̪)	ধ ($d̪^h$)	ন (n)
প (p)-*borgo*	প (p)	ফ (p^h)	ব (b)	ভ (b^h)	ম (m)
অন্তঃস্থ(internal)-sound	য (ʤ)	য় (e̯)	র (ɾ)	ল (l)	
উষ্ম(warm)-sound	শ (ʃ)	ষ (ʃ)	স (s)	হ (h)	
তাড়নজাত(scolding)-sound	ড় (ɽ)	ঢ় (ɽ)			
পরাশ্রয়ী(parasitic)-sound				ঃ (h)	ঁ (ŋ)

Table 1: Bengali *borgo*-phonetic groups

[a] is pronounced as /o/word medially or word finally in specific contexts: nagara /nɔgor/, bakra /bɔkro/.

- **Vowel harmony or vowel height assimilation:** [a] and [e] are pronounced as /o/ resp. /e/ if followed by a high vowel (/u/ or /i/): patha /pɔth/, but pathika /pothik/; ekaTA /ækʈa/, but ekaTu /ekʈu/.
- **Schwa deletion:** [a] is deleted from word final or medial open syllables under specific conditions dependent on phonotactic constraints and etymology. For example: AmarA /amra/, darbAra /dɔrbar/.

Metres and Rhythms in Bengali

Bengali poetry has three basic and common metres: *akṣara-vṛtta*, *mātrā-vṛtta*, and *svara-vṛtta*. The first two were inherited from Sanskrit, while the third is more genuinely Bengali. However, before Tagore popularized it, the *svara-vṛtta* was used mainly for nursery rhymes and not really recognised as a serious poetic metre.

The *mātrā-vṛtta* and *svara-vṛtta* metres are based on the length of the vowels. The *akṣara-vṛtta* metre is in contrast in Sanskrit based on the number of letters in a line (*akṣara* is the Sanskrit letter); however, in Bengali poetry the number of syllables are counted rather than the number of letters. The letters অ (a), ই (i) and উ (u) are counted as being of one unit (*mātrā*) each, that is, a short vowel (*mora*), while এ (e), ঐ (ai), ও (o), and ঔ (au) are counted as being two units each, that is, a long vowel (*macron*). Furthermore, at the end of a line a short vowel may be counted as a long one.

The concepts of open and closed syllables are also central to Sanskrit prosody and poetry: closed syllables are those ending with a vowel sound, while those ending without vowels are called open. In Bengali, a syllable is considered as being one or two units long depending on its position in a line, rather than on whether it is open or closed. If a line begins with a closed syllable, the syllable is counted as one unit, but if it occurs at the end of a line it is counted as two units. In the *mātrā-vṛtta* metre, the position of closed syllables does not matter; they are always counted as two units. In a similar fashion, in the *svara-vṛtta* each vowel (*svara*) is counted as one unit, regardless of whether the syllables are open or closed.

There are three types of rhymes in Sanskrit poetry, depending on whether the rhyme is on the first syllable of each line (*adiprāsa*), or on the second syllable (*dviteeyakshara prāsa*), or if it is the final syllable of the line which is rhyming (*antyaprāsa*). The most important rhyme for our purposes is *antyaprāsa*, which is known as *tail-rhyme* or *end-alliteration* in English, and as *anto-mil* in Bengali poetry.

There are many overviews and in-depth analyses of the metres and rhythms of Bengali poetry written in Bengali, but fairly few available in English. The reader is referred to Arif (2012), or the writings of Aurobindo (2004) that give a more poetic angle. Here, we will concentrate on poems written in *mātrā-vṛtta* metre with *anto-mil* rhyme, as these poems are relatively easy to understand and generate.

The Poetry Generation Model

The previous efforts on investigating computer poetic creativity vary widely in terms of the poetry generation approaches. Some have used document corpus-based models (Manurung 2004; Toivanen et al. 2012), while others have used constraint-programming based models (Toivanen, Järvisalo, and Toivonen 2013) or genetic programming based models (Manurung, Ritchie, and Thompson 2012).

In contrast, we choose a conversation follow-up model highly inspired by the Bengali movie *'Hirak Rajar Deshe'* ('Kingdom of Diamonds', 1980) by Oscar winning director Satyajit Ray (the son of Sukumar Ray, the poet whose writings form the basis of our rhyme understanding corpus, as further discussed below).

In Satyajit Ray's movie, the entire conversation was in rhythm. For example:

এরা যত বেশি পড়ে (1)

Ērā	*yata*	*bēśi*	*paṛē*
they	as much	more	read

'The more they read'

তত বেশি জানে (2)

Tata bēśi jānē
that more know
'The more they learn'

তত কম মানে (3)

Tata kama mānē
that less obey
'The less they obey'

For the present task, the follow-up model means that the system automatically generates a follow-up rhythmic line based on the user's one-line poetry input.

For example, if the given sentence is:

এই দুনিয়ার সকল ভাল (4)

Ē'i duniẏāra sakala bhāla
this world everything good
'All is well in the world'

the machine could generate a follow-up line such as:

আসল ভাল নকল ভাল (5)

Āsala bhāla nakala bhāla
best good fake good
'Real is good, even fake is also good'

There are two essential modules for effective follow-up poetry generation in Bengali: rhyme structure understanding of the given user input and matching rhyme generation. The development of those modules is discussed in turn in the next two sections.

Rhyme Understanding

The initial step involves understanding the rhyme in an input line given by the user. The actual rhyme understanding module consists of syllable identification followed by *borgo* identification and open/closed syllable identification. Firstly, however, it is necessary to collect a corpus in order to understand the rhythm and metre structures of Bengali poems.

Corpus Acquisition

To collect the corpus, several dedicated Bengali poem sites (called *Kobita* in Bengali)[1] were chosen. For the present task, we choose mainly poems written for children, as they mostly are written in *mātrā-vṛtta* metre and with *anto-mil* (tail) rhyme, which is relatively easy to start with for the task of automatic poetry generation. The poems chosen were mainly written by Sukumar Ray (1889–1923), as the rhyme structure of those poems is fairly easy to grasp. A few of Tagore's poems, in particular those written for children, were also collected. Corpus size statistics are reported in Table 2.

This corpus was used later on to train a classifier to predict follow-up rhyme syllables. Therefore, from the collected poems only those pairs of lines were extracted that had both *mātrā-vṛtta* metre and *anto-mil* rhythm.

[1] http://www.bangla-kobita.com/

Type of units	Number
Sentences	3567
Words	9336
Unique tokens	7245

Table 2: Bengali poem corpus size statistics

Syllabification

Syllabification processes depend directly on the pronunciation patterns of any language. In Bengali poetry, open and closed syllables have been used deliberately to continue or stop rhythmic matras (units), as described in the section above on Bengali poetry. These are important features for syllabification.

In order to implement a syllabification engine, we developed a grapheme to phoneme (G2P) converter following the methods discussed by Basu et al. (2009). The consonants and vowels IPA patterns were inherited from that work, while the orthographic and contextual rules were rebuilt. An open-source Bengali shallow parser based POS tagger[2] was used for the task.

With the help of this list, the syllabification engine marks every input word according to its *borgo*. If a word stars with a vowel, the system marks it as a 'v' group. Only the rules mentioned in the paper by Basu et al. have been included, whereas a few things that are not clearly described in the paper remain unattended, for example, some orthographic and exception rules. An example of syllabification output is given in Table 3, where the input is the first line of Sukumar Ray's poem 'Cloud Whims', '*Mēghēra khēẏāla*' (মেঘের খেয়াল).

Borgo Identification

For open syllabic words, identification of the *borgo* class for the final character is quite important. In case no rhythmic follow-up word is available for the last word in the given sentence, an alternative approach is to choose a word that ends with a consonant belonging to the same *borgo*. This helps in keeping the rhythm alive.

For example, in the following sequence (also from Sukumar Ray's poem 'Cloud Whims') the first line ends with ঠ(/ʈʰ/) and the final word of the second line ends with a member of the same *borgo*, namely ট (/ʈ/).

বুড়ো বুড়ো ধাড়ি মেঘ ঢিপি হয়ে উঠে (6)

Buṛō buṛō dhāṛi mēgha ḍhipi haẏē uṭhē
old old inveterate cloud mound becomes
'The very old inveterate cloud looks like a hill'

শুয়ে বসে সভা করে সারাদিন জুটে। (7)

Śuẏē ba'sē sabhā karē sārādina juṭē
laid sitting meeting all day fellows
'They were meeting all the day with the gathered friends.'

[2] http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php

Input	আকাশের	ময়দানে	বাতাসের	ভরে
	akasher	*maẏadane*	*bataser*	*vore*
	ākāśēra	*maẏadānē*	*bātāsēra*	*bharē*
English	In the sky with the air			
Syllables	*ākā-śē-ra*	*maẏa-dānē*	*bātā-sē-ra*	*bharē*
Syllable count	3	2	3	1
Open/Closed	o	c	o	c
Borgo	v	p	p	p

Table 3: Sample syllabification output

Rhyme Generation

The automatic rhyme generation engine consists of several parts. First, an SVM-based classifier predicts syllable sequence patterns. Then, a set of candidate output words are selected from preprocessed syllable-marked word lists. In order to preserve the rhythm in the generated sentence, a few other parameters are checked, such as *borgo* classes, *anto-mil*, and whether the syllables are open or closed. Finally, bigrams are used to prune the list of candidate words and weighted sentence aggregation used to generate the actual system output. These steps are described in detail in turn below.

Syllabic Sequence Prediction

A machine-learning classifier was trained for the syllabic rhyme sequence prediction. The Weka-based Support Vector Machine (SVM) implementation (Hall et al. 2009) was chosen as basis for the classifier The collected poetry corpus described above was used here for training and testing. The training corpus was split into rhythmic pairs of sentences, where the first line would represent the user-provided input whereas the second line would be the one that has to be generated by the system. The input features for the syllabic sequence prediction are: the syllable count sequence of the given line, open/closed syllable pattern sequence of the given line, and the *borgo* group marking sequence of the first given line. The output labels for the training and testing phases are the syllable counts of each word.

For simplicity only those pairs of sentences were chosen where the number of words are same in both the lines. The overall task has been designed as a sequence syllable count prediction, but there are tricky trade-offs for initial position and the last position. The common rhythmic pattern in Bengali poems is *anto-mil* (tail-rhyme), so it is necessary to take care of the last word's syllables separately. Therefore three different ML engines have been trained: One for the initial position, one for the final position, and one for other intermediate positions. Feature engineering has been kept the same for each design, whereas different settings have been adopted for the intermediate positions.

Word Selection

A relatively large word collection was used for the word selection task. The collection consists of the created poem corpus and an additional news corpus.[3] For rhythmic coherence, all words are kept in their inflected forms. In practice, stemming changes the syllable count of any word and may therefore affect the rhythm of the rhythmic sequence.

All word forms are pre-processed and labelled with their syllable counts using the G2P syllabification module. For the word selection, the following strategies have been incorporated serially in the same sequential order as they are described here, in order to narrow down the search space.

Syllable-wise: All words with similar syllabic patterns are extracted from the word list.

Closed Syllable / Open Syllable: Depending on the word in the previous line at the corresponding position, either open or closed syllabic words are chosen. The rest of the words are discarded.

Semantic Relevance: Semantic relevance is very essential to keep the generated rhyme meaningful. There is neither any WordNet publicly available for Bengali nor any relational semantic network like ConceptNet. Therefore the English ConceptNet (Havasi, Speer, and Alonso 2007) and an English-Bengali dictionary (Biśvās 2000) were used to measure the semantic relevance of the automatically chosen words.

Before the semantic relevance judgement, each Bengali word from the given input is stemmed using the morphological analyser, packaged with the Bengali shallow parser. After stemming, those words are translated to English by dictionary look-up. The translated English words are then checked in the ConceptNet and all the semantically related words are extracted. Now, if a selected word co-occurs with the given word in the ConceptNet extracted list, then it is considered as relevant. Otherwise it is discarded. For the ConceptNet

[3] `http://www.anandabazar.com/`

search, only nouns and verbs are considered. For example (same as in Table 3) if the given line is:

আকাশের ময়দানে বাতাসের ভরে (8)

Ākāśēra maẏadānē bātāsēra bharē
sky field air filled

'The sky is filled with the air from the fields'

The words that will be searched in ConceptNet are sky (আকাশ), field (ময়দান), and air (বাতাস). The extracted word list will then definitely contain words such as cloud (মেঘ), which was used by Sukumar Ray in the original poem (again '*Mēghēra khēẏāla*' or 'Cloud Whims'):

ছোট বড় সাদা কালো কত মেঘ চরে। (9)

Chōṭa baṛa sādā kālō kata mēgha carē
small large white black many clouds grazing

'Many large and small, black and white clouds are grazing.'

***Borgo*-wise:** *Borgo*-wise similarity is checked and only words ending in the same *borgo* classes are kept for the last position word. The other words are checked for first letter *borgo*-similarity, and the non-matching are discarded.

Anto-mil: For *anto-mil* or tail-rhyme matching, an edit distance (Levenshtein 1966) based measure has been adopted. If the Minimum Edit Distance is ≤ 2, then any word is considered as homophonic and kept. This strategy only works for the final word position. The remaining members are excluded.

Pruning and Grammaticality

The methods described so far are able to produce word-lists for each word member from the input. Appropriate pruning and natural language techniques are required to generate grammatically correct rhythm sequences from these word options.

N-gram (bigram) matching followed by aggregation is used for the final sentence generation. The n-grams have been generated using the same word collection as described above, that is, the poem corpus plus the news corpus. The system computes weights (*frequency/total number of unique n-grams in the corpus*) for each pair of n-grams. For example, suppose that the total number of generated word candidates for the first position word is n_1 and for the second position word it is n_2. Then $n_1 \cdot n_2$ valid comparisons have to be carried out. The possible candidates will be:

$$\sum_{i=0}^{n_1} w_i^1 \cdot \sum_{i=0}^{n_2} w_i^2 \quad (10)$$

Where the sums intend to represent the relevance of using one term after another to create a meaningful word sequence. Suppose the targeted sentence has m number of words. The process will then be continued for each successive bigram pair, for example, for

$$w^1 - w^2, w^2 - w^3, w^3 - w^4, w^4 - w^5, \ldots, w^{m-1} - w^m$$

Finally, the best possible combination is chosen by maximizing the total weighted path as a multiplication function (that is, by maximizing over the dot product of all the possible n-gram sequences). The process is illustrated in Figure 1.

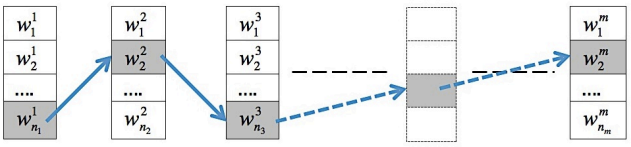


Figure 1: Word sequence selection by n-gram pruning

Experiments and Performance

The generated system has been evaluated in two ways: through a set of in-depth studies by three dedicated expert evaluators and in more free-form studies by ten randomly selected evaluators.

As discussed in the introduction, three major criteria for the quality assessment of automatic poetry generation have been used previously: poeticness, grammaticality, and meaningfulness (Manurung 2004). The same evaluation measures have been applied to the present task. The evaluation process is manual and each of the three dimensions is assessed on a 3-point scale:

- **Poeticness:**
  (3) Rhythmic
  (2) Partially Rhythmic
  (1) Not Rhythmic
- **Grammaticality:**
  (3) Grammatically Correct
  (2) Partially Grammatically Correct
  (1) Not Correct
- **Meaningfulness:**
  (3) Meaningful
  (2) Partially Meaningful
  (1) Not Meaningful

The evaluation results are reported in Table 4, where the scores assigned by three in-depth evaluators are reported separately, while the randomly selected evaluators have been grouped according to whether they should give short (not more than five words) input lines or whether they could give unrestricted length input. The whole assessment process is elaborated on below, including explanations for the scores given by the different evaluators.

Evaluators	Dedicated experts			Randomly chosen	
	#1	#2	#3	≤ 5 words	unrestricted
Poeticness	2.4	1.2	2.1	2.3	1.9
Grammaticality	1.7	1.0	1.4	1.8	0.6
Meaningfulness	1.5	0.9	1.1	1.6	0.8

Table 4: Evaluation of the Bengali poetry generator

In-Depth Evaluation

Three dedicated expert evaluators were chosen for an in-depth evaluation. One of them is a Bengali literature student, the second a Bengali journalist, and the third a technical undergraduate student. Each of them were asked to test the system performance on 100 input sentences, chosen by themselves.

Evaluator 1: Literature Student
The Bengali literature student was instructed to collect 100 simple poem lines from various poets, whose poems were not included in our training set. Through discussion with the evaluator, we decided to choose lines from Satyendranath Dutta's (1882–1922) poems since he is known for his rhyme sense and renowned as the 'wizard of rhymes' (ছন্দের যাদুকর) in Bengali literature. Also, his creatures are very easy to understand.

We started with the famous 'The Song of the Palanquin', *'Palkir Gan'* (পালকির গান). Following are some examples of the output the system produced. The second lines in the examples were generated by the system, while the first lines were given to the system as input.

পালকী চলে ! (11)
'Palanquin moves!'
দুলকি চালে
'Trot pace'

স্তব্ধ গাঁয়ে (12)
'Stunned village'
রুদ্ধ দ্বারে
'Cloggy doors'

The output in Example 11 is surprisingly good. Actually, the same line has been used as follow-up to this input line in one of the paragraphs of the original poem. The output in Example 12 is also good in terms of poeticness, but is less meaningful, while the first output is fabulous for all the evaluation criteria poeticness, meaningfulness and grammaticality. However, we obviously also got many bad output sequences.

Evaluator 2: Journalist
The journalist evaluator was requested to judge the system's performance on news line input and was instructed to chose short sentences with a prior assessment of having a possible poetic sequence. He chose lines from the Bartanam newspaper.[4] The best system output was the one in Example 13, where first line again is the input line and the second line has been generated by the system.

কে হবেন প্রধানমন্ত্রী ? (13)
'Who will be the prime minister?'
গদি নেওয়ার ষড়যন্ত্রী
'Conspirator for the throne'

However, most of the system output in the news domain was unsatisfactory. From discussions with the evaluator, it was eminent that it also is very difficult for humans to generate poetic sequences for any given line, so it is naturally quite difficult for a machine to do this, in particular if the lines are coming from a non-rhythmic news domain.

Evaluator 3: Technology Student
The technical undergraduate student was asked to chose lines from modern Bengali songs, and was instructed to chose smaller and simpler sentences. In the evaluation, she assigned a high score to poeticness, but lower scores to grammaticality and meaningfulness. Thus the system performed better than in the news domain, but inferior to the poetry domain. The best output produced by the system is shown in Example 14.

গভীরে যাও (14)
'Dive into the depth of your heart'
শুধরে নাও
'Rectify yourself'

Evaluation by Random Evaluators

Ten randomly selected evaluators (not connected to the research in any way) were asked to evaluate the system's performance on sentences given by themselves, with the only restriction given that they should provide simple examples with possible tail-rhymes.

The first five of them were instructed to limit their input to five words only. This is in order to understand system performance on longer vs shorter sentences. As a result, we found that system performance is good on all the three aspects on shorter sentences, but that it degrades drastically when longer sentences are given as input. As can be seen in Table 4, this is in particular the case for the dimension of grammaticality, and also true for meaningfulness, while the scores on poeticness are not that bad overall.

[4] `http://bartamanpatrika.com/`

Conclusion

This paper has reported some initial experiments on automatic generation of Bengali poems. Bengali is a morph-syntactically rich language which has inherited the characteristics and fundamentals of its poems from Sanskrit. Automatic rhyme generation for Bengali is therefore a relatively complex problem. The approach taken here is novel and based on interaction with the user who enters a line of poetry, which the system then aims to understand in order to generate a corresponding text line, adhering to the rules and metres of Bengali poetry and rhyming with the input.

This basic system has many drawbacks and limitations, especially in the understanding of wide varieties of rhythms and in terms of grammaticality. The rhyme generation utilises a Bengali syllabification engine and an SVM-based classifier for predicting the structure of the output sentence and for the candidate word generation, which is based on a notion of semantic relevance in terms of proximity mappings derived from ConceptNet translations. The final selection of the actual poetic words is presently done through bigram pruning and aggregation.

Using the notion of semantic relevance is a computationally cheap way to automatically create meaningful rhymes, although poetry written by humans obviously do not always contain semantically related words. However, this is initial work and using ConceptNet is a straight-forward approach; and even though conceptual similarity hardly is the ultimate way to measure word relevance for poems, it is probably one of the easiest ways. In the future, we would aim to involve further natural language generation techniques to create more meaningful poetry.

Acknowledgments

Many thanks go to the evaluators for all their efforts, assistance and comments. We would furthermore like to thank the anonymous reviewers for several comments that helped to substantially improve the paper.

We are grateful to the late Satyajit Ray (1921–1992) for directing the movie 'Kingdom of Diamonds' (*'Hirak Rajar Deshe'*) which originally inspired our approach.

A very special token of appreciation goes to the three Bengali poets who in the early years of the previous century wrote the verses that were used in the building, training and evaluation of our system: Sukumar Ray, Rabindranath Tagore and Satyendranath Dutta.

References

Arif, H. 2012. Prosody. In *Banglapedia: the National Encyclopedia of Bangladesh*. Asiatic Society of Bangladesh, 2 edition.

Aurobindo, S. 2004. *Letters on Poetry and Art*, volume 27 of *The complete works of Sri Aurobindo*. Pondicherry, India: Sri Aurobindo Ashram Publication.

Bailey, R. W. 1974. Computer-assisted poetry: The writing machine is for everybody. In Mitchell, J. L., ed., *Computers in the Humanities*. Edinburgh University Press. 283–295.

Barman, B. 2011. A contrastive analysis of English and Bangla phonemics. *Dhaka University Journal of Linguistics* 2(4):19–42.

Basu, J.; Basu, T.; Mitra, M.; and Mandal, S. 2009. Grapheme to phoneme (G2P) conversion for Bangla. In *Proceedings of the Oriental International Conference on Speech Database and Assessments COCOSDA*, 66–71. IEEE.

Biśvās, Ś. 2000. *Samsad Bengali-English dictionary*. Calcutta, India: Sahitya Samsad, 3 edition.

Chatterji, S. K. 1926. *The Origin and Development of the Bengali Language*. Calcutta University Press.

Colton, S.; Bundy, A.; and Walsh, T. 2000. Automatic invention of integer sequences. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, 558–563. AAAI.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-FACE poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.

Colton, S. 2005. Automated conjecture making in number theory using HR, Otter and Maple. *Journal of Symbolic Computation* 39(5):593–615.

Dan, M. 1992. *Some issues in metrical phonology of Bangla: The indigenous research tradition*. Phd Thesis, Deccan College, University of Poona, Pune (Poona), India.

Díaz-Agudo, B.; Gervás, P.; and González-Calero, P. A. 2002. Poetry generation in COLIBRI. In *Advances in Case-Based Reasoning*. Springer. 73–87.

Funkhouser, C. 2009. *Prehistoric Digital Poetry: An Archaeology of Forms, 1959–1995*. Modern and Contemporary Poetics. University of Alabama Press.

Gervás, P.; Pérez y Pérez, R.; Sosa, R.; and Lemaitre, C. 2007. On the fly collaborative story-telling: Revising contributions to match a shared partial story line. In *Proceedings of the 4th International Joint Workshop in Computational Creativity*, 13–20. Goldsmiths, University of London.

Gervás, P. 2000. WASP: Evaluation of different strategies for the automatic generation of Spanish verse. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, 93–100. AISB.

Gervás, P. 2001a. An expert system for the composition of formal Spanish poetry. *Knowledge-Based Systems* 14(3):181–188.

Gervás, P. 2001b. Generating poetry from a prose text: Creativity versus faithfulness. In *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in Arts and Science*, 93–99. AISB.

Gervás, P. 2002a. Exploring quantitative evaluations of the creativity of automatic poets. In *Proceedings of the 2nd Workshop on Creative Systems, Approaches to Creativity in Artificial Intelligence and Cognitive Science, the 15th European Conference on Artificial Intelligence*.

Gervás, P. 2002b. Linguistic creativity at different levels of decision in sentence production. In *Proceedings of the AISB 02 Symposium on AI and Creativity in Arts and Science*, 79–88. AISB.

Gross, O.; Toivonen, H.; Toivanen, J. M.; and Valitutti, A. 2012. Lexical creativity from word associations. In *Seventh International Conference on Knowledge, Information and Creativity Support Systems*, 35–42. IEEE.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter* 11(1):10–18.

Havasi, C.; Speer, R.; and Alonso, J. B. 2007. ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In *Proceedings of the 6th International Conference on Recent Advances in Natural Language Processing*.

Kar, S. 2009. *The syllable structure of Bangla in Optimality Theory and its application to the analysis of verbal inflectional paradigms in Distributed Morphology*. Phd Thesis, Neuphilologischen Fakultät, Universität Tübingen, Tübingen, Germany.

Kaufman, L. 2012. *The Kaufman Repertoire for Black and White: A Complete, Sound and User-friendly Chess Opening Repertoire*. Alkmaar, The Netherlands: New In Chess.

Levenshtein, V. I. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady* 10(8):707–710.

Levy, R. P. 2001. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the Workshop on Creative Systems, International Conference on Case-Based Reasoning*.

Manurung, R.; Ritchie, G.; and Thompson, H. 2012. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence* 24(1):43–64.

Manurung, H. M. 2004. *An Evolutionary Algorithm Approach to Poety Generation*. Phd Thesis, School of Informatics, University of Edinburgh, Edinburgh, UK.

Mishra, A. 2010. Modelling Aṣṭādhyāyī: An approach based on the methodology of ancillary disciplines (*Vedāṅga*). In *Sanskrit Computational Linguistics*. Springer. 239–258.

Ramakrishnan A, A., and Devi, S. L. 2010. An alternate approach towards meaningful lyric generation in Tamil. In *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*, 31–39. ACL.

Ramakrishnan A, A.; Kuppan, S.; and Devi, S. L. 2009. Automatic generation of Tamil lyrics for melodies. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, 40–46. ACL.

Ritchie, G. D. 2001. Assessing creativity. In *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in Arts and Science*, 3–11. AISB.

Ritchie, G. D. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Sarkar, P. 1986. Aspects of Bengali syllables. In *National Seminar on the Syllable in Phonetics and Phonology*. Hyderabad, India: Osmania University.

Sircar, S., and Nag, S. 2014. Akshara–syllable mappings in Bengali: a language-specific skill for reading. In Winskel, H., and Padakannaya, P., eds., *South and Southeast Asian psycholinguistics*. Cambridge University Press. 202–211.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the Third International Conference on Computational Creativity*, 175–179.

Toivanen, J. M.; Järvisalo, M.; and Toivonen, H. 2013. Harnessing constraint programming for poetry composition. In *Proceedings of the Fourth International Conference on Computational Creativity*, 160–167.

Toivanen, J. M.; Toivonen, H.; and Valitutti, A. 2013. Automatical composition of lyrical songs. In *Proceedings of the Fourth International Conference on Computational Creativity*, 87–91.

Veale, T. 2012. *Exploding the Creativity Myth: The computational foundations of linguistic creativity*. Bloomsbury Academic.

Coming Good and Breaking Bad:

Generating Transformative Character Arcs For Use in Compelling Stories

Tony Veale
School of Computer Science and Informatics
University College Dublin, Belfield D4, Ireland.
Tony.Veale@UCD.ie

Abstract

Stories move us *emotionally* by *physically* moving their protagonists, from place to place or from state to state. The most psychologically compelling stories are stories of *change*, in which characters learn and evolve as they fulfil their dreams or become what they most despise. Character-driven stories must do more than maneouver their protagonists as game pieces on a board, but move them along arcs that transform their inner qualities. This paper presents the *Flux Capacitor*, a generator of transformative character arcs that are both intuitive and dramatically interesting. These arcs – which define a conceptual start-point and end-point for a character in a narrative – may be translated into short story pitches or used as inputs to an existing story-generator. A corpus-based means of constructing novel arcs is presented, as are criteria for selecting and filtering arcs for well-formedness, plausibility and interestingness. Characters can thus, in this way, be computationally modeled as dynamic *blends* that unfold along a narrative trajectory.

Metamorphosis

As Gregor Samsa awoke one morning from uneasy dreams, he found himself transformed in his bed into a monstrous vermin. So starts Franz Kafka's novella of transformation, titled *Metamorphosis*, in which the author explores issues of otherness and guilt by exploiting a character's horrific (if unexplained) change into an insect.

Authors from Ovid to Kafka demonstrate the value of transformation – physical, spiritual and metaphorical – as a tool of character development, just as storytellers from Homer to Kubrick demonstrate the value of journeys as support-structures for narratives of becoming and change. Even narratives that are primarily plot-focused or action-centric can, many times, be succinctly summarized by listing key character transformations. Consider *Gladiator*, an Oscar-winning action film from 2000. The main villain of that piece, *Emperor Commodus*, summarizes the plot with three successive transformations: "The *general* who became a *slave*. The *slave* who became a *gladiator*. The *gladiator* who defied an *emperor*." Note how the third transformation is implicit, for the gladiator *Maximus* has transformed himself into a potential leader of Rome itself.

Kafka presents his driving transformation as a *fait accompli* in the very first line of his story, while in Ovid's *Metamorphoses*, characters are transformed by Gods into trees or animals with magical immediacy. Most narrative transformations occur gradually, however, with a story charting the course of a character's development from a start-state **S** to target-state **T**. In this respect the television drama *Breaking Bad* offers an exemplary model of the slow-burn transformation. We first meet the show's main character, Walter White, in his guise as a put-upon high-school chemistry teacher. "*Chemistry*", he tells us, "*is the study of change*." Though Walter has a brilliant mind, he lives a dull suburban life of quiet desperation, until a diagnosis of lung cancer provides a catalyst to look anew at his life's choices. Walter decides to use his chemistry skills to "cook" and sell the drug *Crystal Meth*, and recruits former student Jessie as a drug-savvy partner. In 62 episodes, the show charts the slow transformation of Walter from dedicated teacher to ruthless drug baron. As the show's writer/creator Vince Gilligan put it, "*I wanted to turn my lead character from Mr. Chips into Scarface*."

Walter's progress is neither smooth nor monotonic. He becomes an unstable, dynamic blend of his start and end states. Though he commits unspeakable crimes, he never entirely ceases to be a caring parent, husband or teacher. As viewers we witness a true conceptual integation of his two worlds: Walter brings the qualities of a drug baron to his family relationships, just as he brings the qualities of a husband and father-figure to his illicit business dealings. To fully appreciate this nuanced character transformation, we must understand it as more than a monotonic journey between two states: characters must unfold as evolving blends of the states that they move between, so they can exhibit emergent qualities that arise from no single state.

This paper presents a CC system – *The Flux Capacitor* – for generating hypothetical character arcs for use in story generation. The *Flux Capacitor* is not itself a story generation system, but a stand-alone system that suggests "what-if" arcs that may underpin interesting narratives. Though it is a trivial matter to randomly generate arcs between any two conceptual perspectives – say between *teacher* and *drug-baron*, or *terrorist* and *politican* – the

Flux Capacitor generates arcs that are well-formed, well-motivated, intuitive and of dramatic interest. It does so by using a rich knowledge-representation of our stereotypical perspectives on humans, knowing e.g. what qualities are exhibited by teachers or criminals. It uses corpus analysis both to acquire a stock of valid start- and end-states and to model the most natural direction of change. It further uses a robust model of conceptual blending to understand the emergent qualities that may arise during a transformation.

The *Flux Capacitor* builds on a body of related work which will be discussed in the next section. The means by which novel transformative arcs are formulated is then presented, before a model of property-level blending and proposition-level analogy/disanalogy is also described. *The Flux Capacitor* does more than generate a list of possible character arcs: it provides to a third-party story generator a conceptual rationale for each transformation, so a story-teller may properly appreciate the ramifications of a given arc. In effect this rationale is a *pitch* for a story. Before drawing our final conclusions, we describe how such a pitch can be constructed from a blending analysis.

Related Work and Ideas

What is a hero without a quest? And what is a quest that does not transform its hero in profound ways? The scholar Joseph Campbell has argued that our most steadfast myths persist because they each instantiate, in their own way, a profoundly affecting narrative structure that Campbell calls the *monomyth*. Campbell (1973) sees the monomyth as a productive schema for the generation of heroic stories that, at their root, follow this core pattern either literally or figuratively: "*A hero ventures forth from the world of common day into a region of supernatural wonder: fabulous forces are encountered and a decisive victory is won: the hero comes back from this mysterious adventure with the power to bestow boons on his fellow man*." Many ancient tales subconsciously instantiate this schema, while many modern stories – such as George Lucas's *Star Wars* – are consciously written so as to employ Campbell's monomyth schema as a narrative deep-structure.

A comparable schematic analysis of the heroic quest is provided by Propp's *Morphology of the Folk Tal*e (1968). Like Campbell, Propp identifies an inventory of recurring classes (of character and event) that make up a traditional Russian folk tale, though Propp's analysis can be applied to many different kinds of heroic tale. Transformative elements in Propp's inventory include *Receipt of Magical Agent*, which newly empowers a hero, *Transfiguration*, in which a hero is rewarded through change, and *Wedding*, through which a hero's social status is elevated. Propp also anticipates that a truly transformed hero may not be recognized on returning home (*Unrecognized Arrival*) and may have to undergo a test of identity (*Recognition*). The basic morphemes of Propp's model can be used either to analyze or to generate stories, in the latter case by using a variant of Fritz Zwicky's *Morphological Analysis* (1969). Propp's morphemes have thus been used in the service of automated game design (Fairclough and Cunningham, 2004) as well as creative story generation (Gervás, 2013).

Campbell's monomyth and Propp's morphology can each be subsumed under a more abstract mental structure, the *Source-Path-Goal (SPG)* schema analyzed by Johnson (1987). Johnson argues that any purposeful action along a path – from going to the shops to undertaking a quest – activates an instance of the SPG schema in the mind. In cinema the SPG is most obviously activated by "road movies", in which (to quote the marketing campaign for *Dances With Wo*lves), a hero goes *"in search of America and finds himself*". Such movies use the SPG to align the literal with the figurative, so that a hero starts from a state that is both geographic and psychological, and reaches an end-point that is similarly dual-natured. The SPG schema is also evident in comic-book tales in which an everyman is transformed into a superheroic form that permits some driving goal (revenge, justice) to be achieved. Forceville (2006) has additionally used the SPG to uncover the transformative-quest structure of less overtly heroic film genres, such as documentaries and autobiographical films.

Storytelling is a purposeful activity with a beginning (*Source*), middle (*Path*) and end (*Goal*) that typically shapes the events of a narrative into a purposeful activity on the part of one or more characters. Computer systems that generate stories – as described in e.g. Meehan (1981), Turner (1994), Perez y Perez & Sharples (2001), Riedl & Young (2004) and Gervás (2013) – are thus, implicitly, automated instantiators of the Source-Path-Goal schema. This is especially so of story systems, like that of Riedl & Young, that employ an explicitly *plan-based* approach to generation. These authors use a planner that is anchored in a model of the beliefs and internal states of the story's characters, so as to construct narrative plans that call for believable, well-motivated actions from these characters. The use of a planner also ensures that these actions create the appearance of an intentional SPG path that is viewed as plausible and coherent by the story's audience.

Outside the realm of myths and fairy-tales, the deepest transformations are to the beliefs and internal states of a character, though such profound changes may be reflected in outward appearances too, such as via a change of garb, residence, place of work, or choice of tools. Consider the case of a prostitute who becomes a nun, or the altogether rarer case of a nun who *breaks bad* in the other direction. Such transformations are dramatically interesting because they create oppositions at the levels of properties and of propositions. Though frame-level symmetries are present, since each kind of person follows a particular vocation in a particular place of work while wearing a particular kind of clothing, the specific frame-fillers are very different. We can imagine a tabloid headline screaming "*Nun burns habit, buys thong*" or "*Nun flees convent, joins bordello*." Analogies and disanalogies between the start- and end-states of a transformation provide fodder for the evolving blends that need to be constructed to ferry a protagonist between these two states in a narrative.

Conceptual blending is a knowledge-hungry process *par excellence* (see Fauconnier and Turner, 1998, 2002).

However, Veale (2012a) presents a computational variant of conceptual blending, called the *conceptual mash-up*, that is robust and scalable. Propositional knowledge is milked from various Web sources – such as query completions from Web search engines – and, using corpus evidence, this knowledge is mapped to more than one concepts. Veale (2012b) also presents a robust method for mining stereotypical properties from Web similes, such as "*as chaste as a nun*" and "*as sleazy as a prostitute*". Used here, these representations allow the *Flux Capacitor* to analyze the blending potential of a transformative arc, and so construct a conceptual rationale as to why a given arc has the potential to underpin an interesting narrative.

Opposites Attract

At its most reductive, a transformative character arc is an unlabeled directed edge **S→T** that takes a character from a conceptual starting-state **S** to a conceptual end-point **T**, where **S** and **T** are different lexicalized perspectives on a character (such as e.g. **S**=*activist* and **T**=*terrorist*). To be a truly transformative arc, as opposed to an arbitrarily random pairing of **S** and **T** states, an arc should induce a dramatic change of qualities. Superficially, this change may be reflected in a reversal of affective polarity from **S** to **T**. Thus, if **S** is viewed as a positive state overall, such as *activist*, *saint* or *defender*, and **T** is predominantly seen as a negative state, such as *terrorist*, *prostitute* or *tyrant*, then a character will *break bad* by following this arc. Conversely, if **S** is most often seen as a negative state, and **T** is typically seen as a positive state, then a character will *come good* by following this arc. Naturally, our overall affective view of a concept will be a function of our property-level perception of all its stereotypical qualities. If **S** typically evokes a preponderance of positive qualities then it will be viewed as a positive state overall. Likewise, if **S** typically evokes a preponderance of negative qualities then it will be viewed as a negative state overall. A means of mapping from property-level representations to overall +/- affective polarity scores is presented in Veale (2011).

Stories thrive on conflict and surprise, and surprising transformations arise when the pairing of **S** and **T** gives rise to a clash of opposing properties. Consider again the case of the *prostitute* (=**S**) who becomes a *nun* (=**T**). The transformation **S→T** at the conceptual-level implies the property-level oppositions *dirty*↔*pure, immoral*↔*moral*, *promiscuous*↔*chaste* and *sleazy*↔*respected*, affording an opportunity for a truly dramatic Proppian transfiguration. Generalizing, we say that a character arc **S→T** implies a direct opposition at the property-level if **S** and **T** each exhibit properties that can produce antonymous pairs. We thus use WordNet (Fellbaum 1998) as a comprehensive source of antonymy relationships (such as *pure*↔*dirty*), which we apply to any putative arc **S→T** to determine whether the arc involves a dramatic conflict of properties.

This property-level analysis allows *The Flux Capacitor* to identify nuanced transformations that allow a character to come good *while also* breaking bad. Consider the arc *beggar→king*. A character following this arc may come good in many ways, by going from *lowly→lordly*, *poor→lofty*, *broke→wealthy, impoverished→privileged* and *ragged→regal*. Yet such an arc may induce negative effects too, changing a character from *humble→arrogant*, *humble→haughty* and *humble→unapproachable*. Perhaps a beggar that becomes a king may come to rue his change of station, while a king that becomes a beggar may derive some small comfort from his fall from grace?

Yet **S** and **T** need not conflict directly at the property-level to yield an opposition-rich transformation. The clash of properties may be *indirect*, if **S** relates to a concept **S'** in the same way that **T** relates to **T'**, and if a clash of opposing properties can be observed between **S'** and **T'**. For instance, *scientists* and *priests* do not directly oppose one another, but a property-level clash can be found in the stereotypical representations of *science* and *religion*, since science is stereotypically rational while religion is often seen as irrational. Since scientists practice science while priests practice religion, a character that goes from being a scientist to being a priest will, in a leap of faith, reject *rational* science and embrace *irrational* religion instead.

A gifted storyteller can surely make an transformation, no matter how random or illogical, seem interesting. Such is the art of improvizational comedy, after all. However, rather than abdicate its responsibility for making an arc interesting to a subsequent story-telling component, the *Flux Capacitor* applies it own filtering criteria to find the arcs it considers to have dramatic potential. An arc **S→T** is generated only if **S** and **T** possess opposing qualities, or if **S** and **T** are indirectly opposed by virtue of being analogously related to a concept pair **S'** and **T'** that do. We now turn to how **S** and **T** are found in the first place.

Charging the Capacitor

We often speak of children in terms of what they may one day become, but speak of adults in terms of what they have *already* become. Some concepts are more naturally thought of as start-states in a transformation, while others are more naturally viewed as end-states. Beyond the clear cut cases, most concepts sit on a continuum of suitability for use on either side of a transformation. To determine the suitability of a given concept **C** as either a start state or an end state, we can simply look to a large text corpus. The frequency of the 2-gram "**C**+*s become*" in a corpus such as the Google n-grams (Brants and Franz, 2006) will indicate how often **C** is viewed as a start-state, while the frequency of the 2-gram "*become* **C**+*s*" will indicate **C**'s suitability as an end-state. Since the n-gram frequency of "*become terrorists*" (7180) is almost 7 times greater than the frequency of "*terrorists become*" (1166), *terrorist* is far more suited to the role of end-point than to start-point.

The *Flux Capacitor* limits its choice of start-states to any stereotype **S** for which the Google n-grams contains the bigram "***S***+*s become*". Similarly, it limits its choice of end-states to any stereotype **T** for which Google provides the bigram "*become* ***T***+*s*". Within these constraints, the Google n-grams suggests 1,213 person-concepts to use as start-states, and 1,529 to use as their ultimate end-states.

The Google n-grams contains a small number (< 500) of well-established transformations between person-types that can be found via the pattern "**S**-*turned*-**T**". Examples include *friend*-turned-*foe*, *bodybuilder*-turned-*actor* and *actor*-turned-*politician*. Though some turns have dramatic value (like *bully*-turned-*Buddhist*), most are well-trodden paths with little to offer a creative system. Nonetheless, the Google n-grams are a valuable source of inspiration for the generation of novel transformations that combine complementary ideas. For the n-grams can tell us whether two ideas have a history of working well together, either in harmony or as part of an antagonistic double-act.

Consider the 3-gram pattern "**X**+s *and* **Y**+s", which matches all instances of coordinated bare plurals in the Google n-grams. Examples include "*angels and demons*", "*nuns and prostitutes*" and "*scientists and priests*". While these attested coordinations often bring together opposing concepts, they are concepts drawn from the same domains or semantic fields, and thus seem *fitted* to each other. So while a transformation linking two such conflicting states may strike one as a surprising turn of events, it will also likely strike one as a *fitting* turn of events. By mining the Google 3-grams for instances of this pattern that connect a valid start-state to a valid end-state, where these states also exhibit either a direct or indirect conflict of qualities, the *Flux Capacitor* harvests a large collection of potential state-pairs for its own transformative character arcs. The question of which state can best serve as a start-state, and which should serve as the end-state, is decided afterwards.

Coordinations are a rich source of explicit constrasts between conceptual states, but other n-grams are an even richer source of *implicit* contrasts. Consider the 3-gram "army of dreamers". The typical member of an army is a *soldier*, not a *dreamer*, as borne out by the system's own propositional world-knowledge. This 3-gram thus implies a clash of soldiers and dreamers, which in turn implies the property-level conflicts *disciplined*↔*undisciplined* and *fit*↔*lethargic*. Generalizing, we mine all Google 3-grams that match the pattern "<**group**> *of* <**person**>+s", such as "*church of heretics*", "*army of cowards*" and "*religion of sinners*", to identify any cases where the stated member (*sinner*, *coward*, etc) contrasts with a known stereotypical member of the group. A large pool of contrasting concept pairs is mined in this way from the Google n-grams, to be used to form each side of a transformative character arc.

But what trajectory should each transformation follow? Which concept will serve as the start-point **S** of an arc, and which as its end-point **T**? We infer the most natural direction for an arc by again looking to corpus data. For a pair of contrasting concepts **X** and **Y**, we calculate a score for the arc **X**→**Y** as the sum of the n-gram frequencies for "**X**+*s become*" and "*become* **Y**+s". Likewise, we calculate the score for the arc **Y**→**X** as the sum of the n-gram frequencies for "**Y**+*s become*" and "*become* **X**+s". We then choose the arc/direction with the greatest score. Consider, for example, the pair *militant* and *politician*, which share, in the world-view of the *Flux Capacitor*, this implicit contrast: militants launch *celebrated* rebellions, whilst politicians launch *hated* wars. Corpus data suggests that *politician* is more suited to be the end-state of an arc than its start-state, perhaps because politicians must be elected, and election is an obvious goal-state in the SPG schema. In contrast, *militant* is slightly more comfortable in the role of start-state than end-state, no doubt because militants fight so as to initiate some future change. Thus, the arc *militant*→*politician* is favored over its inverse, *politician*→*militant*, and so only the former is generated.

Blended States

In character-led stories, key transformations often unfold gradually through a build-up of incremental changes. So as characters follow their trajectory along an arc that takes them ever closer to their final state, they will exhibit more of the qualities we stereotypically associate with the endpoint of their arc and fewer of the properties we associate with their starting point. In effect, a changing character becomes a dynamic *blend* of the starting-point and end-point concepts that define its narrative trajectory.

The theory of conceptual integration networks, also known as *conceptual blending* (see Fauconnier & Turner, 1998, 2002), offers a principle-driven framework for the interpretation of any blend, while Veale (1997) further explores the workings of character blends that gradually unfold during a narrative. A character blend – a character that moves between two states and thus assumes a mix of the properties and behaviors associated with each – can be modeled computationally at the level of properties and of propositions. To model the former, we explore the space of complex properties that integrate nuances from each of the inputs, while to model the latter we draw on Markman and Gentner's (1993) theory of *alignable differences*.

Consider a proposition-level blend in the shocking case of our *nun-turned-prostitute*. The alignable differences in this example concern the propositions associated with nuns and with prostitutes that can be aligned by virtue of positing exactly the same relationship for each subject, but with different values for their objects. For instance, nuns work and reside in convents or cloisters, under the supervision of a mother superior, while prostitutes work and reside in bordellos under the supervision of madams and pimps. So as this transformation is effected, convents and cloisters will give way to bordellos, while mother superiors will lose out to pimps and madams, just as wimples and habits will transition into an altogether racier style of dress. It is a simple matter to connect propositions with alignable differences such as these, to produce a structural blend that is part analogy and part disanalogy.

The *Flux Capacitor* is also sensitive to the reversals of status and power that accompany a given transformation. By attending to the relationships that link a subject A to an object B, and the relationships that reciprocally link B as a subject to A as an object, it learns how to recognize

situations where a protagonist's social inter-relationships are dramatically reversed in a blend. Thus, for instance, it observes a fundamental tension between the verbs *obey* and *control*, between *ruling* and *being led*, and between *governing* and *electing*. In the case of a *king-turned-slave* then, it perceives an interesting reversal of power, where a once-mighty king goes from being served by respectful followers to being led by haughty and arrogant rulers, just as he may go from appointing fawning servants to being managed by dominant and exalted masters. The scale of each reversal is emphasized by highlighting the most pointed contrasts between the blended states; thus, it also suggests that our deposed king goes from being served by *honorable* knights to being led by *depraved* rulers. While these new rulers need not be depraved, it heightens the dramatic potential of the blend to assume that they are.

At the property-level, we strive to understood how a property *A* associated with a start-state **S**, and a property *B* associated with an end-state **T**, might yield an emergent property *AB* that arises from a character's transformation from **S** into **T**. Might our *nun-turned-prostitute* retain a residual sense of *piety*, even if such piety were to be unjustified or even immoral? The Google 2-grams inform us that the phrase "*immoral piety*" denotes an attested state (with a Web frequency of at least 49). Since nuns are typically pious and so practice piety, while prostitutes are typically seen as immoral, *immoral piety* denotes the kind of nuanced state that may arise as one state gives way to the other. The Google n-grams also suggest, in this vein, that a nun-turned-prostitute might be a *moral prostitute*, a *compassionate prostitute*, a *religious prostiute* or, at least, a *spiritual prositute*, one that commits *pure* or *virtuous sins* despite practicing a *sleazy morality* and a *dirty faith*. Likewise, when intellectuals become zealots, attested 2-grams that bridge both states include "*inspired rant*", "*misguided superiority*", "*uncompromising critique*", "*extreme logic*", "*intellectual obsession*", "*scholarly zeal*" and even "*educated stupidity*".

The Google n-grams attest to the validity of a great many complex states that can be surprising and revealing. By seeking out nuanced states that bridge the properties of the conflicting concepts in a character arc, the *Flux Capacitor* can tap in to the vast, collective imagination of readers and writers as exercised for other, past narratives.

Hold The Presses

These blend interpretations serve to advertize the merits of a given character transformation: the richer the blend, in terms of aligned propositions and nuanced properties, the richer the narrative it should yield when turned over to a dedicated story-generation system. In many ways then, these blend interpretations are the computational version of a Hollywood story *pitch*, in which a screenwriter *sells* his or her vision of a story to the studio that will make it. Like a Hollywood studio, which can only afford to make a small number of films per year, a story-generation system will need some narratological basis to judge which stories ideas to further refine and which to reject outright.

The *Flux Capacitor* is not a story-generation system, but a creator of high-concept story ideas. Yet to better *sell* these ideas, it uses natural-language generation techniques to convert its blend analyses into simple pitches. Consider the following pitch, in which each mapping in the blend for *nun→prostitute* has been realized as its own sentence:

Nun condemns chastity, wallows in wickedness
Nun criticizes convents, bounces into brothels
Nun chucks crucifixes, gropes for garters
Nun fatigued by fidelity, veers toward vices
Nun hates habits, stockpiles stilettos
Nun mistreated by mother superiors, pulled to pimps
Nun skips out of spectacles, loves latex
Nun vents about veils, crazy for corsets
Nun vents about virginity, seduced by shamelessness
Nun whines about wimples, grabs garters
Nun goes from being managed by abbesses and mother superiors to being controlled by pimps
Nun goes from carrying beads to carrying infections
Does strict chastity struggle with wild promiscuity?
How long can outer purity suppress inner filth?
Nun goes from being unflinchingly faithful to being increasingly unfaithful
Nun goes from living in cloisters and convents to working in brothels and bawdy houses
Can inner morality be transformed into naked sin?
Nun goes from practicing chastity to practicing vices
How long can a superficial respectability suppress pervasive sleaze?
Nun goes from wearing habits and crucifixes to wearing corsets and fishnets
Nun goes from wearing veils and spectacles to wearing latex and stilettos
Nun goes from wearing wimples to wearing hotpants

Note the simple structure of each sentence in the pitch. Wherever possible, a tabloid-headline style is employed, using alliteration – as in *condemns chastity, wallows in wickedness* – to make each stage of a transformation seem more compelling. Such devices, though simple, embody a strategy that psychologists call the *Keats heuristic*, for the use of even the most rudimentary rhymes has been empirically shown to heighten the perceived truthfulness of a statement (see McGlone and Tofighbakhsh, 2000).

Conversely, character transformations can also be used to craft rhetorical questions and figurative allusions for automated poetry. The *Stereotrope* system of Veale (2013) thus generates rhetorical questions such as "*how does a selfish wolf become a devoted zealot?*", "*how does a devoted zealot become a selfish bully?*" and "*how does a mindless zealot become a considerate lover?*" to allude to unknown protagonists whose identify must ultimately be determined by the reader.

Transformative Possibilities

The *Flux Capacitor* uses the corpus-based techniques of the previous sections to construct 63,016 unique character transformation arcs, using a combination of the Google n-grams, a large database of stereotypical properties, and a propositional model of world-knowledge. Each arc links character states that conflict either directly or indirectly, where each gives rise to its own blending interpretation.

Some arcs simply demand too much from an audience. Novel character arcs may be provocative, but they should rarely be jarring. Arcs that strain credulity, or require an element of cod science to work at all, are best avoided. While it is not possible to predict every faultline along which a narrative may rupture, it is worth considering the most obvious problem-cases here, as these allow us to draw broad generalizations about the quality of our arcs.

The first problem-case concerns gender. Though there exist famous *and* dramatically successful exceptions to this rule, such as Virginia Wolff's *Orlando*, characters rarely change their gender during a transformation. Of the valid start/end states used by the *Flux Capacitor*, 84 are manually annotated as *male*, such as *pope* and *hunk*, while 72 are annotated as *female*, such as *geisha* and *nun*. All other states are assumed to be compatible with both male and female characters. In all, 9,915 of the 63,016 arcs that are generated involve one or more gender-marked states. Of these, only 7% involve a problematic mix of genders (e.g. *pope→mother*). Though a creative story-teller might make lemonade from these lemons (e.g. as in the tale of Pope Joan, who passed as a man until made pregnant), the *Flux Capacitor* simply filters these arcs from its output.

The second problem-case concerns age. Once again, though Hollywood may occasionally find a cod-science reason to reverse time's arrow, characters rarely transform into people younger than themselves. Not wishing to paint a story-teller into a corner, where it must appeal to a dust-blown plot device such as time travel, body swapping or family curses to get out, the *Flux Capacitor* aims to avoid generating such arcs altogether. So of its valid start/end states, 52 are manually tagged for age to reflect our strong stereotypical expectations. Elders such as *grandmother*, *pensioner* and *archbishop* are assigned a timepoint of 60 years, while youths such as *student*, *rookie* and *newcomer* are given a timepoint of 18. Younger states, such as *baby*, *toddler*, *child*, *kid*, *preteen* and *schoolgirl*, are assigned lower time-points still, while those states unmarked for age are all assumed to have a default timepoint of 30. In all, 7,892 arcs are generated for which one or more states is explicitly marked for age. Now, if our corpus-based approach to determining the trajectory of an arc is valid, we should expect most of these 7,892 arcs to flow in the expected *younger→older* direction. In fact, 76% of arcs do flow in the right direction. The remaining 24% are not simply discarded however. Rather, these arcs are inverted, turning e.g. *mentor→student* into *student→mentor*.

The ultimate test of a character transformation is the quality of the narrative that can be constructed around it. We cannot evaluate the quality of these narratives until they have been woven by a subsequent story-generation system, acting as a user of the *Flux Capacitor*'s outputs. Nonetheless, the diversity of the *Flux Capacitor*'s outputs – 63,016 well-formed arcs, bridging 1,213 start-states to 1,529 end-states in interesting ways that pair concepts that conflict *and* which also exhibit corpus-attested affinities – is a reason to be optimistic about the quality of the many as-yet-unwritten stories that may employ these arcs.

Back to the Future

Georges Braque, who co-developed *Cubism* with Pablo Picasso, was less than impressed with the arc of Picasso's career, noting late in life that "*Pablo used to be a good painter, but now he's* just *a genius*." If character arcs induce change, such changes are just as likely to remove a desirable quality as add it. For Braque, to go from noted painter to certified genius was to follow a downward arc, for Picasso was now to be feted more for his politics, his lifestyle and his women than for any of his painterly gifts. Braque's view of Picasso's career is witty because it runs against expectation: to become a genius is often seen as the highest of achievements and not a vulgar booby prize. As we strive to make the *Flux Capacitor* generate arcs that seem interesting yet plausible, we must remember that it is not just a transformation *per se* that can be original, but the manner in which we choose to interpret it, not to mention the way we ultimately use it in a story.

Creativity requires more than generative capability, and a generative system is *merely generative* if it can perform neither deep interpretation nor critical assessment nor insightful filtering of its own outputs. Though the *Flux Capacitor* is just one part of a story-generation pipeline, it is not just a *mere* generator of character arcs. It operates in a large space of possible transformations, sampling this space carefully to identify those transformations that change a character in dramatically interesting ways into something that is at once both incongruous and fitting.

The property transfers that accompany a transformation may serve as causes or as effects. That is, some property shifts initiate a change while others naturally follow on as consequences of these root causes. Consider the case of a *king-turned-slave*, in which the *Flux Capacitor* identifies the following wealth of property conflicts and shifts:

worshipped→contemptible, revered→contemptible, lofty →inferior, lofty→subservient, lofty→submissive, anointed→cursed, powerful→powerless, powerful→ contemptible, powerful→frightened, powerful→scared, powerful→inferior, magisterial→powerless, learned→ illiterate, learned→uneducated, commanding→ cowering, commanding→subservient, commanding→ passive, commanding→powerless, commanding→

submissive, rich→powerless, rich→malnourished, rich →miserable, merry→miserable, merry→unfortunate, crusading→frightened, august→contemptible, celebrated →contemptible, honored→contemptible, regal→ powerless, spoiled→whipped, spoiled→abused, spoiled →overworked, spoiled→exhausted, spoiled→ malnourished, spoiled→overburdened, spoiled→ exploited, comfortable→miserable, contented→unhappy, contented→miserable, delighted→unhappy, leading→ submissive, leading→subservient, ruling→submissive, ruling→subservient, lordly→inferior, pampered→ whipped, pampered→abused, pampered→overworked, pampered→exhausted, pampered→malnourished, pampered→overburdened, pampered→exploited, prestigious→inferior, prestigious→subservient, prestigious→submissive, reigning→submissive, reigning →subservient, royal→inferior, royal→subservient, exalted→inferior, exalted→subservient, deified→ powerless, beloved→cursed, beloved→miserable, beloved→contemptible, beloved→condemned, magnificent→powerless, magnificent→miserable, magnificent→contemptible, honorable→contemptible, great→powerless, dominant→subservient, dominant→ dependent, dominant→inferior, dominant→submissive, mighty→powerless, mighty→low-level, mighty→ contemptible, mighty→scared, fortunate→unfortunate, fortunate→cursed, fortunate→unhappy, fortunate→ miserable, consecrated→cursed, worthy→miserable, adored→contemptible, happy→unhappy, happy→miserable, happy→unfortunate, venerated→contemptible, grand →powerless

Dramatic changes are very often precipitated by external actions, and some states – expressed as past-participles – are easily imagined as both the primary cause and direct effect of a transformation. Thus, the property *cursed* may serve as both cause and effect of the dramatic humbling of a king, when perhaps cursed by a witch, demon or other entity as suggested by attested n-grams (e.g. "*cursed by a witch*"). Further n-gram analysis will also suggest that one who is *cursed* may also be be *condemned* and *abused*, while one who is *abused* is more likely to be *hungry* and *dependent*. Or perhaps our king is first *defeated*, since the Google 3-grams suggest *defeat* leads one to become *powerless*, that being *powerless* leads to being *oppressed*, and that *oppression* leads one to being *tortured*, *miserable* and *unhappy*. The next stage of the *Flux Capacitor*'s development will thus focus on imposing a plausible causal ordering on the properties that undergo change in a transformation, to provide more conceptual insight to any story-generation system that exploits its character arcs.

A story-generation system may then use a Proppian or Campbellian analysis to impose narrative structure on any such character arc. For a transformed character effectively undertakes a journey, whether or not this journey takes place entirely within one's mind or social circumstances. By better understanding how the arrow of causality may impose a narrative ordering on the property-changes in a story, a system can better impose the morphology of a folk-tale or a monomyth on any generated character arc. This system may ask which property changes conform to what Propp deemed a *Transfiguration*, and which can best underpin the role of a *Magical Agent* in a story? Does a character return, or attempt to return, from the end-state of a transformation, and which actions or events can make such a *Return* possible? What property changes make a character difficult to *Recognize* post-facto, and which initial properties of a character continue to shine through?

We do not see the *Flux Capacitor* as a disinterested sub-contractor in the story-telling process, but an active collaborator that works hand-in-glove with a full story-generator to help weave surprising yet plausible stories. As it thus evolves from being a simple provider of arcs to being a co-creator of stories in its own right, we expect that its usefulness as a sub-contractor to existing story-generation systems will yield insights into the additional features and functionalities it should eventually provide.

Out of the Mouths of Bots

To showcase the utility of the *Flux Capacitor* as a sub-contractor in the generation of creative outputs, we use the system as a key generative module in the operation of a creative *Twitterbot*. Twitterbots, like *bots* in general, are typically simple generative systems that autonomously perform useful, well-defined (if provocative) services. A Twitterbot is an automated generator of tweets, short micro-blog messages that are distributed via the social media platform Twitter. Most twitterbots, like most bots, are far from creative, and exploit *mere generation* to send superficially well-formed texts into the twittersphere, so in most cases, the conceit behind a particular twitterbot is more interesting than the content generated by the bot.

Twitter is the ideal midwife for pushing the products of true computational creativity – such as metaphors, jokes, aphorisms and story pitches – into the world. A new *twitterbot* named ***MetaphorIsMyBusiness*** (handle: ***@MetaphorMagnet***) thus employs the *Flux Capacitor* to generate a novel, well-formed, creative metaphor or story pitch every hour or so. As such, *@MetaphorMagnet*'s outputs are the product of a complex reasoning process that combines a large knowledge-base of stereotypical norms with real usage data from the Google n-grams. Though encouraged by the quality of the bot's outputs, we continue to expand its expressive range, to give the twitterbot its own unique voice and identifiable aesthetic. Outputs such as "*What is an accountant but a timid visionary? What is a visionary but a bold accountant?*" show how *@MetaphorMagnet* frames the conceits of the *Flux Capacitor* as though-provoking metaphors, to lend the bot a distinctly hard-boiled persona. Ongoing work with the bot aims to further develop this sardonic voice.

There are many practical advantages to packaging creative generation systems as Web services, but there are just as many advantages to packaging these services as twitterbots. For one, the panoply of mostly random bots on Twitter that make little or no use of world knowledge or of true computational creativity – such as the playfully subversive *@metaphorminute* bot – provide a competitive baseline against which to evaluate the creativity and value of the insights that are pushed out into the world by theory-driven and knowledge-driven twitterbots like *@MetaphorMagnet*. For another, the willingness of human Twitter users to *follow* such accounts regardless of their provenance, and to *favorite* or *retweet* the best outputs from these accounts, provides an empirical framework for estimating (and promoting) the quality of the back-end Web services in each case. Finally, such bots may reap some social value in their own right, as sources of occasional insight, wit or profundity, or even of useful metaphors or story ideas that are subsequently valued, adopted, and re-worked by human speakers.

References

Thorsten Brants and Alex Franz. (2006). Web 1T 5-gram database, Version 1. *Linguistic Data Consortium*.

Joseph Campbell. (1973). *The Hero With A Thousand Faces*. Princeton University Press.

Chris Fairclough and Pádraig Cunningham. (2004). AI Structuralist Storytelling in Computer Games. In *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*.

Gilles Fauconnier and Mark Turner. (1998). Conceptual Integration Networks. *Cognitive Science,* 22(2):133–187.

Gilles Fauconnier and Mark Turner. (2002). *The Way We Think. Conceptual Blending and the Mind's Hidden Complexities.* New York: Basic Books.

Christiane Fellbaum (ed.). (2008). *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Charles Forceville. (2006). The source-path-goal schema in the autobiographical journey documentary: McElwee, Van der Keuken, Cole. *The New Review of Film and Television Studies* 4:3, 241-261.

Pablo Gervás. (2013). Propp's Morphology of the Folk Tale as a Grammar for Generation. In *Proceedings of the 2013 Workshop on Computational Models of Narrative*, Dagstuhl, Germany.

Mark Johnson. (1987). *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*. University of Chicago.

Arthur Markman and Dedre Gentner. (1993). Splitting the differences: A structural alignment view of similarity. *Journal of Memory and Language* 32(4):517–535.

Matthew McGlone and Jessica Tofighbakhsh. (2000). Birds of a feather flock conjointly (?): rhyme as reason in aphorisms. *Psychological Science* **11** (5): 424–428.

James Meehan. (1981). *TALE-SPIN*. In Roger Schank and C. K. Riesbeck (eds.), *Inside Computer Understanding: Five Programs plus Miniatures*. Hillsdale, NJ: Lawrence Erlbaum.

Rafael Pérez y Pérez and Mike Sharples. (2001). *MEXICA: A computer model of a cognitive account of creative writing*. The Journal of Experimental and Theoretical Artificial Intelligence, 13: 119-139.

Vladimir Propp. (1968). *Morphology Of The Folk Tale* (second edition). University of Texas Press.

Mark Riedl and Michael Young. (2004). An intent-driven planner for multi-agent story generation. In Proc. of 3rd International Joint Conference on Autonomous Agents and Multi-agent Systems, 186-193.

Scott R. Turner. (1994). *The Creative Process: A Computer Model of Storytelling*, Hillsdale, NJ: Lawrence Erlbaum.

Tony Veale. (1997). Creativity as pastiche: A computational treatment of metaphoric blends, with special reference to cinematic "borrowing". In *Proc. of Mind II: Computational Models of Creative Cognition*, Dublin, Ireland.

Tony Veale. (2011). The Agile Cliché: Using Flexible Stereotypes as Building Blocks in the Construction of an Affective Lexicon. *In: Oltramari, A., Vossen, P., Qin, L. & Hovy, E. (eds.) New Trends of Research in Ontologies and Lexical Resources. Springer: Theory and Applications of Nat. Lang. Processing.*

Tony Veale. (2012a). From Conceptual Mash-ups to "Bad-Ass" Blends: A Robust Computational Model of Conceptual Blending. *In Proceedings of ICCC 2012, the 3rd International Conference on Computational Creativity*. Dublin, Ireland.

Tony Veale. (2012b). *Exploding the Creativity Myth: The Computational Foundations of Linguistic Creativity*. Bloomsbury.

Tony Veale. (2013). Less Rhyme, More Reason: Knowledge-based Poetry Generation with Feeling, Insight and Wit. In Proc. of the 4th International Conference on Computational Creativity, Sydney, Australia.

Fritz Zwicky. (1969). *Discovery, Invention, Research: Through the Morphological Approach*. Toronto: Macmillan.

A Model of Runaway Evolution of Creative Domains

Oliver Bown
Design Lab, University of Sydney,
NSW, 2006, Australia
oliver.bown@sydney.edu.au

Abstract

Creative domains such as art and music have distinct properties, not only in terms of the structure of the artefacts produced by in terms of their cultural dynamics and relation to adaptive functions. A number of theories have examined the possibility of functionless cultural domains emerging through a runaway evolutionary process. This includes models in which engaging in creative domains is actually counterproductive at the individual level, but is sustained as a behaviour through an evolutionary mechanism. I present a multi-agent model that examines such an evolutionary mechanism, derived from these theories.

Introduction

The study of computational creativity involves both general theory and domain-specific theoretical and experimental studies. Domains such as music, visual art and humour have very different properties owing mainly to the ontological and structural nature of the artefacts produced. But we also know that these domains have different socio-cultural natures. For example, Hargreaves and North (1999) and Huron (2006), discuss social functions and contextual factors that appear to be specific to music, and may not have any relevance to art or humour (although they could). A major contribution to computational creativity therefore involves the computational modelling of specific domains, as in the classic examples described in Miranda, Kirby, and Todd (2003), and more abstract notions of creative domain dynamics, as studied by Saunders and Gero (2001) and Sosa and Gero (2003), drawing on the theoretical formulation of Csikszentmihalyi (1990). The specific analysis of creative domains – their origins, dynamics and relations to individual motivations – makes a critical contribution to computational creativity by framing how we should understand the evaluation of automated creative agents acting in those domains. This paper follows the latter work but looks at the more fundamental evolutionary question of the emergence of creative domains, i.e., how humans came to exhibit behaviour in specific realms such as art and music, either through genetic or cultural evolutionary processes.

The approach used here follows the epistemological method, established in multi-agent modelling fields such as artificial life (Di Paolo, Noble, and Bullock, 2000) and computational social science (Conte et al., 2012), of attempting to reveal novel mechanisms through the study of the emergent qualitative outcomes of local interactions in computer simulations.

The model presented in this paper is based on theories of the evolution of music and takes the form of a minimal abstract model of biological evolution. However, it does not directly look at modelling music, but at a proposed model of underlying social interactions that would allow a runaway evolutionary process to take place. Theoretically this is grounded in the ideas of cultural evolution provided by Boyd and Richerson (1985) and Laland, Odling-Smee, and Feldman (2000). In the language of Laland, Odling-Smee, and Feldman (2000), the model is an experimental study of the 'construction of cultural niches' which remains generic for the sake of simplicity, but could be later developed into a specific model of the construction of a music niche, or applied comparatively to different creative activities. A niche is defined here as a site of fitness acquisition for an individual. Niches can be pre-existing, as in the use of trees for birds, or constructed, as in the alteration of an ecosystem by a beaver building a dam. The model can be interpreted as a general model of runaway evolution of creative domains.

In my conclusion, I discuss the applicability of this model, and more generally this type of modelling, to developing a richer understanding of creative domains that may inform computational creativity. This and similar models provide candidate properties of creative domains that directly inform the way we view the analysis and evaluation of individual creative systems within specific domain contexts.

'Runaway' Theories of the Origins of Musical Behaviour

The origins of music are mysterious and highly contested. In *The Descent of Man* (1883), Darwin introduced the principle of sexual selection and suggested that various aspects of human appearance and behaviour, including music, may be sexually selected. The theory of sexual selection states, in modern genetic terms, that since reproductive achievement is key to the perseverance of genetic lineages, then genetic adaptations that increase ones attractiveness to potential mates will prosper. The theory of sexual selection was developed considerably by Fisher (1915), who proposed that

a runaway selection of arbitrary traits could occur if male traits and female preferences coevolved (since females typically have the greater investment in reproduction they are typically the choosier sex). The question of whether sexually selected traits can be fully arbitrary has been the subject of much debate. As part of a general principle that underlies the contemporary study of 'honest signalling theory', Zahavi (1975) proposed that female preference is likely to be guided towards traits that are actually an external (visible or audible) indicator of some positive quality. Thus when male traits and female preferences coevolve, it is those pairings that lead to stronger fitter males that persevere. For example, the quality of a bowerbird's nest indicates the ability of the bowerbird in foraging.

More recently, Miller (2000) has revived the argument that music, amongst other aspects of human appearance and behaviour, is sexually selected. Miller presents musical ability as an indicator trait of general intelligence and health. The theory continues to attract attention but competes with a number of other theories about the origins of musical behaviour. Two strong competing theories are that music serves some cooperative function (Brown, 2007), and that music has no function at all, instead being a cultural innovation that exploits human aesthetic preferences (Pinker, 1998). Both runaway sexual selection and this cultural exploitation theory fit well with an apparent lack of function in music. Although evidence does exist to support social functions in music that would support the cooperative view, this view has also struggled to gain traction due to uncertainty surrounding plausible mechanisms for the evolution of altruism (Fisher, 1958). The sexual selection view has also been criticised because of a lack of typically sexually dimorphic traits in humans with respect to music, and the prevalence of music in situations that appear to have nothing to do with courting, such as at funerals and heavy-metal concerts (Huron, 2001).

However, runaway evolutionary processes are not limited to sexual selection. Zahavi's (1975) examples of honest signalling, for example, extend to other coevolutionary situations. Boyd and Richerson (1985) propose a runaway cultural evolutionary process based on a set of heuristics describing how individuals adopt cultural traits, based on frequency and status. They hypothesise that people are more likely to adopt a cultural trait the more other people adopt that trait, and the higher the status of the people are. They also propose that minimal discrimination is applied to the choice of traits to adopt, on the basis that false positive assumptions are more acceptable than false negative assumptions. In this way potentially arbitrary traits exhibited by high status individuals can easily and rapidly become adopted. Blackmore (1999) develops similar principles through the theory of memetics, and suggests that various aspects of culture, even language, might be understood as having emerged as 'parasites', exploiting human behaviour to become established. These views align with Pinker's view of music as a functionless cultural innovation. Such theories also raise the possibility of a coevolution between genes and culture, which has been explored by a number of theorists, most notably Laland, Odling-Smee, and Feldman (2000). Their extensive theoretical and empirical review suggest that sexual selection and Boyd and Richerson's runaway cultural evolution are just instances of a more general tendency for runaway evolutionary processes to occur between environments and organisms, and that there may be other ways in which runaway evolution could occur in cultural systems. Here the term 'environment' includes culture, and culture is viewed as a site with great potential to exhibit runaway evolutionary processes.

A Model of Runaway Evolution

Little research has been done into how specific cultural forms such as music might be explained by runaway evolution. In this paper, I present a model that provides a very simple mechanism whereby runaway selection of arbitrary cultural domains can become established.

The model is predicated on the broad question underpinning runaway evolutionary processes: under what circumstances will populations of individuals evolve to exhibit traits or engage in behaviour that has no net advantage? Models such as those of runaway sexual selection present such circumstances and show how they are viable. Whilst peacock tails are a burden to peacocks as far as flying or escaping predators are concerned, they give the individual peacock with the better tail a reproductive advantage and thus a net fitness gain. The peacock's tail is understood in terms of the niche created by the peahen's evolved sexual preference, and vice versa. By analogy, in the present case, the goal is to examine examples of cultural behaviours where a similar emergent cultural niche could be established. In our case, we choose to examine a scenario that is not underpinned by sexual selection, but by economics. Primate social organisation is sufficiently complex to lend to the idea that human evolution has been guided by very simple but significant forms of economic interaction. In particular, simple forms of transferrable wealth might have had the capacity to influence fitness dynamics, stimulating the emergence of new cultural niches through positive feedback. Transferrable and cumulative wealth has the capacity to influence evolutionary fitness by allowing one person to effectively take fitness from another person, and, on a macroscopic scale, for societies to develop systems by which to organise their collective wealth, in effect providing some top-down determination of fitness. Under such circumstances, the nature of that social system would have a significant influence on an individual's choice of fitness strategies and this might ultimately have an influence on culturally evolved behaviour, and possibly even a genetic influence. Note that transferrable wealth could mean something such as rights to land that is not achieved technologically, but merely requires a simple concept of ownership or title, although in the present case wealth is also considered cumulative, which might entail something being harvested, or simple things such as clothing being made. Given their simplicity, these factors plausibly predate the creative domains under consideration.

But what has this got to do with creative domains such as art and music? A number of recent studies have looked at how creative success is organised at a social level, suggesting that there is inherent positive feedback in the way that we

allocate reward for creative achievements. Salganik, Dodds, and Watts (2006), for example, show that music ratings are directly influenced by one's perception of how others rated the music, not just in the long term but at the moment of making the evaluation. The result is a winner-takes-all outcome, where a piece of music that is rated highly by others is more likely to be highly rated in the future, as long as people are aware of the already-high regard given to the work. Rather than directly appraising creative works in terms of their content, they are appraised as social artefacts, subject to social processes that transcend the creative content itself. If this is true, then one potential effect of individuals engaging in creative domains is to create winner-takes-all redistributions of some social entity, most broadly described as *prestige*, that may be assumed to relate in some way to wealth.

Accepting the assumption that any given creative domain has no other fitness-enhancing function, then in evolutionary terms it can be understood as a time and effort commitment that needs to be explained. The present model looks to reduce such a scenario to its simplest abstraction and consider the evolutionary effects (whether generic or cultural). In particular, it asks whether it is possible that the creative domain acts to reinforce itself over time, thus providing a evolutionary explanation in the form of niche construction. For this to be demonstrated, a population must be shown to transition from not engaging in the creative behaviour to engaging in it. This occurs when those who engage in the creative behaviour are more successful than those who do not engage over evolutionary time. The model presented here looks at how this can happen over evolutionary time, despite the net average benefit for engaging in creative domains being lower than for avoiding them.

Model Design [1]

The model has a very specific purpose, which is to show how an arbitrary activity can emerge amongst a population of rational selfish agents. Underlying the model, a simple economic system is implemented in which wealth is tied to evolutionary fitness. Agents with higher wealth have a greater chance of survival and are therefore driven by natural selection to maximise wealth. The purpose of the model is to demonstrate evolutionary scenarios in which emergent social conditions favour acting in an apparently irrational way, by engaging in an arbitrary functionless behaviour: a 'game'. The functionless behaviour in turn provides the conditions for runaway evolution.

Note that evolution here can refer to the evolution of genes or of culturally (vertically) acquired traits, interchangeably. Thus the model works as either a biological or a cultural evolutionary model. For the purpose of this paper I refer to genes in the model, but these can be replaced by 'memes' that are vertically transmitted.

The model consists of a fixed population of N agents. Evolutionary competition is implemented through tournament selection. Each agent has the following genetic variables:

- Tendency to play the game (G_i): the probability that an agent will chose to play the game in a given round. At each time step, each agent is identified either as a gamer or a non-gamer;
- Competence (C_i): the game is predominantly random, but there is a bias towards agents with a higher competence;
- Taxation Vote (T_i): all agents vote on a level of taxation that non-gamers should pay into the game, the tax at each round is the average of these T_i.

Each agent also has a wealth variable, W, which is modified through transactions as described in the sequence below.

The following sequence is run at each time step:

1. All agents accumulate a fixed 'pay', p.
2. A globally imposed non-gamer 'tax', t is calculated as the average of all agents' Taxation Votes, T_i.
3. All agents are asked if they wish to play the game in the current round, resulting in a number n of gamers. The tendency to play the game, G_i, is treated as a probability that determines this choice.
4. All gaming agents pay a fixed cost, c, whilst all non-gaming agents pay the non-gamer tax, t. Non-gamer agents also receive the fixed non-gamer bonus, b.
5. The game winner is determined as follows: two different agents are randomly chosen from the list of gamers. The agent with the greatest competence, C_i, out of these two candidates wins. In the case of equal ability to cheat, a random agent is the winner. The winner receives all of the bids, $n \times c$, and all of the tax, $(N - n) \times t$.
6. A fixed number m of reproductive tournaments are run as follows: two different agents are randomly chosen from the population. The agent with the greatest wealth is the winner. In the case of equal wealth, a random agent is the winner. The loser is replaced by a child (mutated copy) of the winner. The parent gives a fixed proportion, w, of its wealth to its child.
7. All agents' wealth is depreciated by a wealth depreciation coefficient, d ($0 \leq d \leq 1$). Each agent's wealth is scaled by this number.

Children's G_i, C_i and T_i values, the genetic variables, are copies of the parent with a Gaussian mutation with a standard deviation of 0.001. G_i and T_i values are constrained between 0 and 1. C_i values are only constrained with a lower bound of 0. The parent gives a fixed proportion, w, of its wealth to its child. Unless otherwise stated, initial values for all agents are $G_i = 0$, $C_i = 0.5$ and $T_i = 0$.

The model variables used in the studies detailed in this paper used the values specified in Table 1.

Starting from an initial value of zero, an increase in the mean tendency to play the game, $\bar{G}$ is then interpreted as a scenario in which game-playing behaviour has become established in the population. The model is designed to reveal

[1] All code for the software model can be found at https://www.dropbox.com/s/48oy1v32lx0utp0/LotteryMain.java. The variables described in this paper differ from those in the code, which are based on the scenario of a lottery game.

Var	Description	Value
p	Pay for all agents at each time step	1
c	Cost of bid paid by gamers	1
b	Bonus paid to non-gamers	1
m	Reproduction tournaments per iteration	10
w	Proportion of wealth paid to children	0.2
d	Depreciation of wealth at each time step	0-0.999

Table 1: Values used in experiments. All values are fixed except the experimental variable d.

the conditions that are required for this to arise. $\bar{G}$ is subject to dynamic selection pressures and can also drift, if no strong selection is observed. Through propagation through a population the range of drifting G values can appear to have low variance, so low variance is not considered sufficient to indicate strong selection. Constraint of the variable to a specific range over a long period of time and multiple runs is used: if $\bar{G}$ sits consistently above 0.8, it is concluded that a game behaviour has emerged in the population.

We assume that individuals are equally able to generate transferrable wealth at a fixed rate, p, per time step. For the game, players put a unit of their wealth, c, into a pot and one individual, chosen at random, wins the entire pot. In addition, we assume that game-playing has a fixed time cost. This is implemented as a further payment, b, to non-gamers. The relative values of p, c, and b therefore define a space of possible model parameters with possible outputs with regard to how G evolves.

Results

The wealth depreciation coefficient (d) was compared across 4 values, 0, 0.9, 0.99 and 0.999. In the first case, wealth is transitory, acquired at the beginning of each time-step, then either spent in the game or kept, and then used to compete in tournaments. For values of d approaching 1, wealth becomes increasingly cumulative. This has two implications: firstly, wealth reaches higher levels, since with a constant income the stable state wealth value is greater. Greater wealth takes longer to accumulate and means that individual gains are ultimately less relevant. Secondly, the gains of short-term successes stick around longer and are more likely to transform into reproductive success. These can also be transferred to children.

Figure 1 shows model outcomes for the values of d, 0.9, 0.99 and 0.999. Each graph shows the average of the 'tendency to play the game' genetic variable, $\bar{G}$, in the population over time, with 20 runs of the model superimposed on each graph. $\bar{G}$ tends toward its upper bound in models with $d = 0.999$, whereas it does not drift far away from zero in models with low d ($d = 0$ and $d = 0.9$). Even for $d = 0.999$ there is the potential for $\bar{G}$ to drift down as well as up, indicating that population-wide game behaviour under favourable circumstances is not as strong an evolutionary stable-state as game-avoidance under unfavourable circumstances. These results show that the durable, transferrable forms of wealth discovered by humans create a situation conducive to the formation of game-playing.

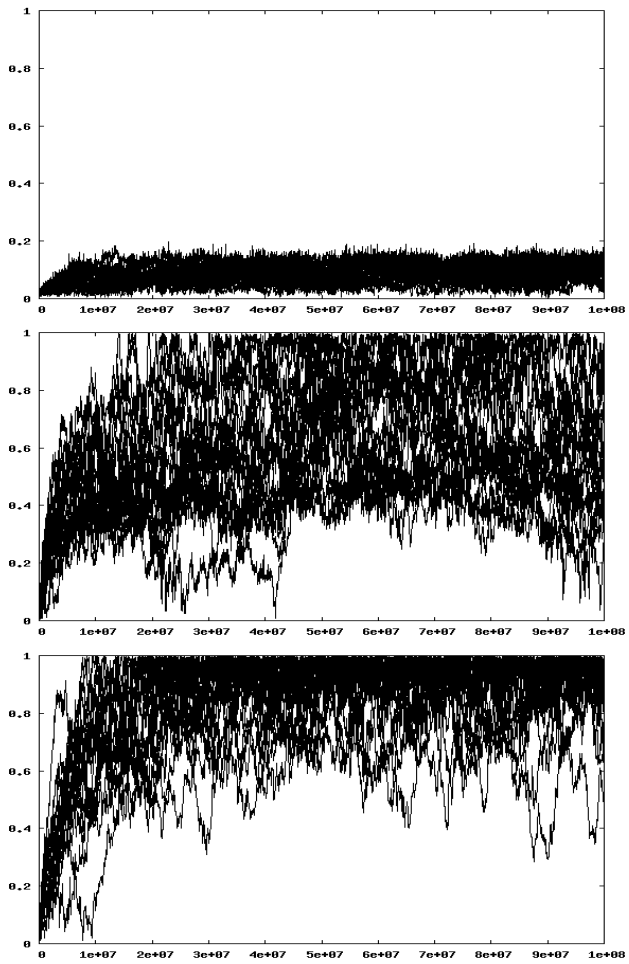


Figure 1: Evolutionary runs with wealth depreciation coefficient, d, values of (from top to bottom) 0.9, 0.99 and 0.999. Each graph shows the mean 'tendency to play the game' genetic variable, $\bar{G}$, evolving over 100 million time-steps, repeated over 20 runs of the model. The taxation vote is allowed to evolve genetically.

Figure 2 shows a typical instance of the model for $d = 0.999$ and evolvable taxation vote T, with $\bar{G}$ in red and the mean Taxation Vote genetic variable, $\bar{T}$ in green. Both values are attracted towards their upper bound of 1, with $\bar{T}$ more inclined to drift. It may be a reasonable assumption that these variables are positively mutually reinforcing, though this has yet to be tested.

In order to understand the specific economic pressures on individuals, a simplified study was conducted with the taxation vote set to a fixed value. To further clarify the model, the accumulated tax was not passed to the game winner, as described above, but was instead discarded. This makes it easier to measure the average expected incomes of individuals in the non-gaming and gaming categories, since average incomes are no longer frequency-dependent (as compared to the standard model where tax channels wealth from non-gamers to the winning gamer). In this simplified model,

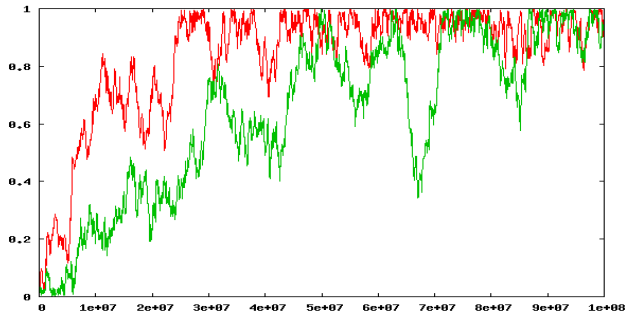


Figure 2: An example evolutionary run with wealth depreciation coefficient, d, of 0.999. The graph shows the mean 'tendency to play the game' genetic variable, $\bar{G}$, in RED, and the mean taxation vote genetic variable, $\bar{T}$, in GREEN, evolving over 100 million time-steps for a typical run.

non-gamers gain $(p + b - T)$ units of wealth at each time step. Gamers do not gain benefits b or pay tax T. Since the game is zero-sum their average income is simply p. Non-gamers all receive the average income, whereas gamers' real incomes are skewed according to the outcome of individual games.

Figure 3 shows the emergence of game playing (situations where $\bar{G}$ tends towards 1) for different values of T, under these conditions. For $T = 0.4$ game playing begins to emerge. The transition from non-game to game takes the form of a sudden phase shift with an erratic onset, and no transitions occur in the opposite direction, implying that game-playing is evolutionarily stable in the population once established. With $T = 0.6$ game playing consistently emerges. In the latter case, the average non-gamer income is $(p + b - T) = (2 + 1 - 0.6) = 2.4$ whereas the average gamer income is $p = 2$. Therefore even when the non-gamer group is fitter on average, the gamer group comes to dominate. This shows a minimum requirement for game playing to emerge. By comparison, the graph at the bottom of Figure 1 shows that this result is robust if T is allowed to vary genetically, even when initial values for G and T are zero.

Figure 4 shows the mean competence genetic variable, $\bar{C}$ increasing steadily without limit for the same run as the graph in Figure 2. $\bar{C}$ exhibited this increase consistently across all runs, even with $d = 0$. By the model design there can be no circumstances under which lower C is advantageous, and always the occasional accidental game that selects in favour of higher C. The purpose of modelling C is not to show that it increases, which is inevitable and obvious, but to show that it has no impact on the emergence of the game behaviour, despite undermining its 'fairness'. We can say that random success is sufficient for the game to emerge, and may enable the initial adoption of the behaviour, but that it is not strictly required. What matters is that the game is robust once established, and creates a stable scenario in which C is driven to evolve. In this model, C is just a numerical variable that is driven to evolve indefinitely towards higher values, but in its place more complex models could explore the potential for the game-playing niche to drive a runaway

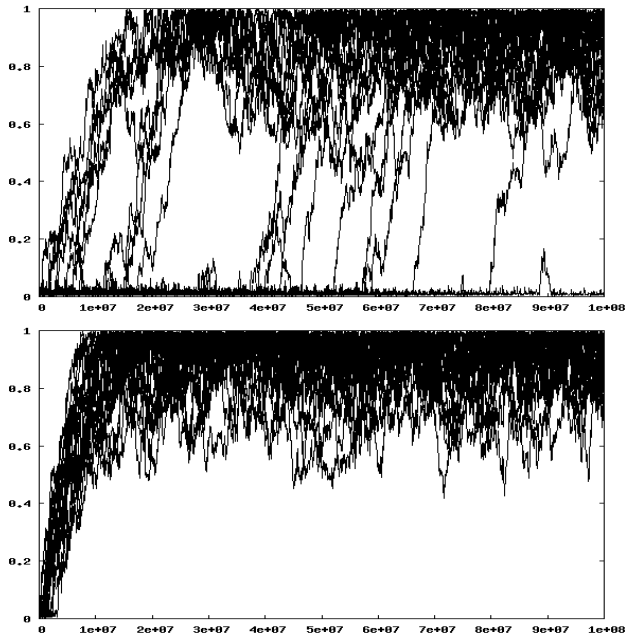


Figure 3: Evolutionary runs with wealth depreciation coefficient, d, values of (from top to bottom) fixed 0.999. Each graph shows the mean 'tendency to play the game' genetic variable, $\bar{G}$, evolving over 100 million time-steps, repeated over 20 runs of the model. In this case the taxation vote is fixed at 0.4 (top) and 0.6 (bottom). Furthermore, in these instances taxes are *not* passed onto the game winner but are simply discarded.

arms-race of game-playing skill, with each winner passing on the greatest skill traits to the next generations.

Discussion

Summary of Results

To summarise the key results, the model shows how a population can evolve an apparently economically irrational behaviour that drives inequality. A greater durability of wealth increases the tendency for game playing to occur, even if the net benefit to the average individual is lower. The emergence of evolutionarily stable game playing behaviour creates a selective pressure driving the constant and rapid increase in game playing ability, but as the population evolves together towards greater competence, the game itself is sustained. As discussed, the properties of this system resemble a set of hypothesised properties of creative domains, satisfying a niche construction view of their emergence.

The results therefore reveal a hypothesised emergent cultural niche which, too all extents and purposes, is functionless, but provides a site for individual fitness acquisition (albeit achieved by lottery) by individuals, and drives a runaway competitive coevolution amongst the population of greater competency in this domain.

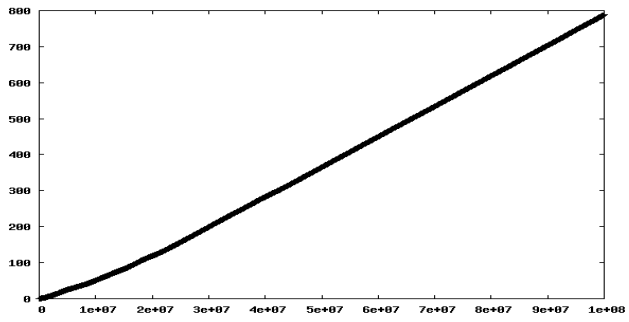


Figure 4: A simulation run ($d = 0.999$) showing the mean competence genetic variable, $\bar{C}$, evolving over time. In all cases, including $d = 0$, $\bar{C}$ increased without limit.

On Randomness

The choice to base the model on a lottery-like game was not discussed in the theoretical background, but is also grounded in a well-founded evolutionary concept. Given the evidence for winner-takes-all processes in human artistic domains, the possibility that randomness is a significant part of the process is actually something that should be seriously considered. A possible role of randomness in structuring social systems, proposed by Wilson (1994), supports a functional role for randomness.

Along with heredity and meritocracy, Wilson (1994) shows that chance can and does play a role in the construction of socially structured systems. The clearest and most striking example of this is the determination of gender, a stochastic process occurring in development, that leads to a prominent social distinction, underpinned by physiological divergence. Looking at the abstract properties of our biological system of gender, Wilson (1994) argues that there may be any number of other behavioural traits determined through a similar process: genetically determined phenotypic variations derived from a common genotype, allocated stochastically. They are, by this definition, not environmentally determined, and are therefore strictly chance allocations, not local adaptations. It is through a stochastic process that a given distribution of possible behaviours emerges, just as in the case of gender, where we end up with a roughly 50-50 split.

Wilson proposes variation along a boldness-shyness personality scale as a candidate example. Assume that boldness and shyness are both proven to be optimal behavioural strategies in different social contexts (in the context of art we could map these onto traits such as creativity and conformity). Typically we think of phenotypic plasticity as the only approach to arriving at good context-dependent behavioural strategies such as these. A plasticity-based view of these traits is that an individual would learn from cues in their environment to be either shy or bold. An equally plausible explanation, Wilson argues, is that the trait is randomly assigned by a stochastic developmental process. Assuming that, to some extent, individuals can find roles that suit their phenotypes (*i.e.*, there are places in the social system where both shy and bold individuals can thrive better than the other), and that an appropriate range of roles is available, then all individuals can emerge well-adapted. Thus a social structure that demands a mix of traits can coevolve with this kind of stochastic allocation of traits. The principle of self-organisation can explain the resulting assignment of roles.

This explanation is also satisfying because the genetic mechanism for stochastically switching between two evolved behavioural variations is arguably simpler than the psychological mechanism required to work out which behaviour strategy is successful in a given, novel context. In addition, the precise source of randomness might be at a number of different stages other than in the genetics. For example, boldness-shyness development could be triggered by events that are effectively random, *i.e.*, there is nothing in the content of the trigger that conveys relevant information about the environment. In the case of creative domains, as suggested by the present model, creative success could be allocated randomly, with the effect that those creatively successful individuals act to reinforce the existence of the creative domain for future generations. This is only to say that random allocation of creative success may be sufficient for the creative domain to work. In reality, creative success may also depend on non-random processes, as with our competence variable.

An important clarification of this principle is that it is not necessary for every individual to do equally well out of the situation for it to be evolutionarily viable: a principle well established by sociobiologists, as in the respective reproductive fitness of different individual ants in a colony. Instead, the process can produce clear inequalities. This parallels the principle of kin selection; kin-directed altruism is able to evolve in proportion to the degree of relatedness between kin, based on the fact that altruism between close kin is as good a way for genes to persevere as individual selfishness. Kin-selection is widely believed to be the most robust mechanism by which cooperative behaviour emerges in nature (Maynard Smith and Szathmáry, 1995).

Conclusion

The model of runaway evolution presented in this paper simply provides a mechanism whereby a pattern of behaviour resembling human creative domains can emerge. The provision of a mechanism does not in any way help prove the theory that music and other creative domains emerged through runaway evolution, but enables predictions derived from the mechanism provided.

The simulation model can be tested against studies of the nature of creative success over multiple generations, taking into account the relationship between creative success, core economic motivations, overall fitness and other contextual factors. In particular, the model predicts that the motivation to engage in creative domains is irrational in the short term but evolutionarily stable in the long term. We can test this by looking at the immediate payoff to art practitioners of varying levels of success. The model predicts that this payoff would be poor in the short term, but that this apparently irrational behaviour could be explained by a process of rein-

forcement occurring at the social level, whereby creatively successful individuals effectively assert the status quo.

Such factors provide a wider context for thinking about the evaluation of artificial creative systems. Evaluation as presently conducted on an individual case-by-case basis (system by system or output by output) may need to be revised to take into account a more complex understanding of the relationship between long-term creative dynamics and short term creative success. Rather than building one virtual Mozart or virtual Picasso, we may need to deploy millions of them in virtual communities in order to truly understand creative success.

References

Blackmore, S. J. 1999. *The Meme Machine*. New York: OUP.

Boyd, R., and Richerson, P. J. 1985. *Culture and the Evolutionary Process*. Chicago, IL, US: University of Chicago Press.

Brown, S. 2007. Contagious heterophony: A new theory about the origins of music. *MusicæScientiæ* 11(1):3–26.

Conte, R.; Gilbert, N.; Bonelli, G.; Cioffi-Revilla, C.; Deffuant, G.; Kertesz, J.; Loreto, V.; Moat, S.; Nadal, J.-P.; Sanchez, A.; et al. 2012. Manifesto of computational social science. *The European Physical Journal Special Topics* 214(1):325–346.

Csikszentmihalyi, M. 1990. The domain of creativity. In Runco, M., and Albert, R. S., eds., *Theories of Creativity*. Sage Publications. 190—212.

Darwin, C. 1883. *The Descent of Man and Selection in Relation to Sex*. New York, USA: Appleton and Co.

Di Paolo, E.; Noble, J.; and Bullock, S. 2000. Simulation models as opaque thought experiments. In A.Bedau, M.; McCaskill, J. S.; Packard, N. H.; and Rasmussen, S., eds., *Articial Life VII: Proceedings of the Seventh International Conference on Articial Life*, 497–506. Cambridge, MA: MIT Press.

Fisher, R. A. 1915. The evolution of sexual preference. *Eugenics Review* 7:184–192.

Fisher, R. A. 1958. *The Genetical Theory of Natural Section*. London: Dover.

Hargreaves, D. J., and North, A. C. 1999. The functions of music in everyday life: Redefining the social in music psychology. *Psychology of Music* 27(1):71–83.

Huron, D. 2001. Is music an evolutionary adaptation? *Annals of the New York Academy of Sciences* 930(1):43–61.

Huron, D. 2006. *Sweet Anticipation*. MIT Press.

Laland, K. N.; Odling-Smee, J.; and Feldman, M. W. 2000. Niche construction, biological evolution, and cultural change. *Behavioral and brain sciences* 23(01):131–146.

Maynard Smith, J., and Szathmáry, E. 1995. *The Major Transitions in Evolution*. New York: Oxford University Press.

Miller, G. 2000. Evolution of human music through sexual selection. In Wallin, N. L.; Merker, B.; and Brown, S., eds., *The Origins of Music*. Cambridge, MA, USA: MIT Press. 329–360.

Miranda, E. R.; Kirby, S.; and Todd, P. M. 2003. On computational models of the evolution of music: From the origins of musical taste to the emergence of grammars. *Contemporary Music Review* 22(3):91–111.

Pinker, S. 1998. *How the Mind Works*. London, UK: Allen Lane The Penguin Press.

Salganik, M. J.; Dodds, P. S.; and Watts, D. J. 2006. Experimental study of inequality and unpredictability in an artificial cultural market. *science* 311(5762):854–856.

Saunders, R., and Gero, J. S. 2001. Artificial creativity: Emergent notions of creativity in artificial societies of curious agents. In *Proceedings of the Second Iteration Conference*.

Sosa, R., and Gero, J. S. 2003. Design and change: A model of situated creativity. In Bento, C.; Cardoso, A.; and Gero, J., eds., *Approaches to Creativity in Artificial Intelligence and Cognitive Science, IJCAI03*, 25–34.

Wilson, D. 1994. Adaptive genetic variation and human evolutionary psychology. *Ethology and Sociobiology* 15:219–235.

Zahavi, A. 1975. Mate selection—a selection for a handicap. *Journal of theoretical Biology* 53(1):205–214.

Computational Creativity: A Philosophical Approach, and an Approach to Philosophy

Stephen McGregor, Geraint Wiggins and Matthew Purver
School of Electronic Engineering and Computer Science
Queen Mary University of London
s.e.mcgregor@qmul.ac.uk, geraint.wiggins@qmul.ac.uk, m.purver@qmul.ac.uk

Abstract

This paper seeks to situate computational creativity in relation to philosophy and in particular philosophy of mind. The goal is to investigate issues relevant to both how computational creativity can be used to explore philosophical questions and how philosophical positions, whether they are accepted as accurate or not, can be used as a tool for evaluating computational creativity. First, the possibility of symbol manipulating machines acting as creative agents will be examined in terms of its ramifications for historic and contemporary theories of mind. Next a philosophically motivated mechanism for evaluating creative systems will be proposed, based on the idea that an intimation of dualism, with its inherent mental representations, is a thing that typical observers seek when evaluating creativity. Two computational frameworks that might adequately satisfy this evaluative mechanism will then be described, though the implementation of such systems in a creative context is left for future work. Finally, the kind of audience required for the type of evaluation proposed will be briefly discussed.

Introduction

In quotidian interactions, either on a personal or social level, computers are such familiar devices that their operations are taken for granted as having the same kind of relatively universal grounding that humans engaging in interpersonal exchanges of information employ. When computers become either the platform for or the object of philosophical enquiries, though, it becomes necessary to talk about them as information processing systems or as symbol manipulating machines (per Newell and Simon, 1990): in this sense, the operations which computers perform must be seen as transpiring in an abstract space, defined by a system of information grounded somehow relative to an observer. This quality of computation immediately introduces a problematic element of subjectivity to the assessment of a purely informational system's ability to generate meaning, and an ambiguity arises over whether such a system can really autonomously produce output which has been invested with semantic content.

It is due to precisely this key feature of computational systems, their dependence on an observer for operational coherence, that computers have become an element in various philosophical discussions, often in the form of *reductiones ad absurdem*, exercises aimed at problematising both reductionist and internalist accounts of mental phenomena. Putnam (1988) in particular has argued for the computational significance of the internal states of a rock, while Searle (1990) constructed his famous Chinese room argument to demonstrate the absence of intentionality in machines which merely manipulate symbols, a stance subsequently used as a platform for questioning the very basis of cognitive experience. In these examples, computers come out as the foils for arguments about the intractable difficulty of defining or even talking about human consciousness. Rather than treating computers as the theoretical objects of thought experiments, this paper will argue, as Sloman (1978) did several years ago, that computers should be considered essential tools for doing good philosophy, and that in particular the question of whether computers can be autonomously creative is philosophically valid.

This paper's first objective is to place the field of computational creativity within the context of the philosophy of mind, and in particular to consider how the field might be used as a vehicle for empirically exploring the problem of dualism which has been characteristically at the centre of questions regarding the mind and consciousness in modern Western philosophy. To this end, a strong counterargument to the traditional mode of dualism, which argues that the mind and physical matter occupy two mutually irreducible spaces, can be found in considering ways in which symbol manipulating machines might be able to autonomously produce informational artefacts that are new and valuable and that furthermore bear some sort of meaning relevant to the way in which the creative system itself operates. If a computational system can produce new, valuable artefacts in a way that is deemed suitably creative, and yet these systems are themselves reducible to manipulations of symbols grounded in the workings of a physical machine, there seems to be no case to make for the idea that the act of generating new meaning in the world transpires in some intangible mental domain.

The second objective of this paper is to propose a new mechanism for evaluating creative systems, motivated by insights into the way that humans view themselves. Taking the intransigence of the mind/body problem as a starting point, it will be suggested that it is precisely the kind of represen-

tational internal states that dualists have attributed to the immaterial space of the mind that should be sought in the operations of creative agents. While a positive assessment of the creativity of an informational system would clearly negate the premise of a mental space separate from physical reality, it is argued herein that precisely this negation serves as a good basis for using the mere impression of such states in a system as an ersatz device for evaluating the real presence of creative behaviour. To this end, two topical computational frameworks, vector space models and deep belief networks, will be put forward as candidates for future work in various domains of computational creativity, with the view that these approaches to computation have the potential to build conceptual structures which might be considered by some observers as corresponding to the type of mental representations attributed to humans.

Computational Creativity and the Demise of Dualism

Descartes' (1911) theory of a mind/matter divide, and the notion of internal mental representations which in particular have characterised the type of introspective reports of the mental space described by philosophers of this bent, have been at the centre of the development of modern Western philosophy, with subsequent canonical philosophers routinely name checking Descartes. The dualism inherent in the mind/matter world view has, however, fallen so severely into disrepute with latter day theorists of mind that a cognitive scientist recently felt comfortable in asserting that in the field today, "even the word 'Cartesian' is often used as a term of abuse," (Rowlands, 2010, p 12). Indeed, in their immensely public debate over the nature of consciousness, Dennett and Searle (1995) resort to mutual accusations of existential partitioning, with both thinkers avowing their own faithfulness to what they perceive as the fundamental, indeed, explicitly monist type of data on which an analysis of existence should be based and upon which any theory of consciousness must supervene.

So the great feuds of contemporary philosophers have been characterised not by a debate over the extent of the merits and faults of dualism, but rather by quarrelling about the precise way in which this dead idea should be autopsied. Whether from the material perspective of reductionist science or from the subjective vantage point of emergent intentionality, the idea that the mind inhabits some physically irresolvable realm has been rejected. This rejection has done little, however, to mitigate the deep issues which characterise the problem of cognition. Furthermore, where strong dualism has been largely vanquished from the philosophical vanguard, it seems even more clear that blunt behaviourism has been thoroughly rebuffed: the idea that cognition can be discussed in terms of simply observed bodily reactions is considered philosophically infeasible (see Boden, 2006, for an overview). The mind evidently experiences the world not as simulating data, but as an array of semantically loaded entities that interact on various levels and according to various rules. The consequent problem of what constitutes perceptual cognition has been characterised as "the binding problem", by which the mind must somehow perform the trick of corralling multifarious sensory stimuli into a unified experience of reality consisting of discernible, describable things which exist on various levels of abstraction.

With this in mind, certain radical views are open to misinterpretation as harbingers of a Cartesian resurrection. For instance, Chalmers (1996) describes a nuanced functionalism by which an agent is conscious by merit of the processes that it performs on a certain level of abstraction regardless of the physical mechanisms of those processes, and Pattee (2008) posits that language and physics should be viewed as two intertwined but mutually irreducible phenomena. Humans are somehow engaging in the act of meaning, in the sense that Wittgenstein indicated when he wrote that "only the act of meaning can anticipate reality" (Wittgenstein, 1967, p 76): it is the characteristically human ability to see a world full of meaningful things rather than just a world full of data. It is not clear, however, how the binding problem is solved, and how the multifarious world is transformed from material input into expressions which are likewise fundamentally material through a cognitive process which is somehow perceptive and expressive. The human ability to perpetually perform this trick is the subject of the dispute between Searle and Dennett, and is the object of what Chalmers has characterised as the "hard problem" of consciousness. The answers to these questions remain arguably as opaque as they were in Descartes' world.

It would seem that computational creativity should, in principle, be the darling of any effort to empirically vanquish any remnant of dualism: to show that a physically grounded symbol manipulating machine is capable of participating as an agent in meaning-making interactions could only illustrate the fallacy of the supposition that such things occur in some kind of non-material space. Wiggins (2012) has recently argued that creativity is in fact the substrate of consciousness, with the capacity for an agent to imagine the world as being different than it is serving as the basis for cognitive action in an environment. In this scenario, an information theoretical process corresponds to Wittgenstein's act of meaning, with statistical computations of perceptual data emerging as semantically gravid expectations of what will happen in an environment. Creativity itself becomes precisely Wittgenstein's act of producing new meaning, of building new ways of perceiving and anticipating the world on different levels of abstraction. Notwithstanding the resilient arguments from Searle (1990) that purely informational symbol manipulating systems cannot have intentionality at the root of their machinations, it would seem that just the ability for an algorithmic machine to be creative would at least prove that the basis of consciousness can be in the material world of physics.

So the argument here is not that, in being creative, in participating in the act of meaning, computers have some chance of becoming conscious. Even with this caveat, though, there is an inherent causal ambiguity in the stance that computers can be creative: it is not clear that the idea that machines can autonomously generate meaningful output *a priori* is necessarily sound. In fact, short of imposing emergent phenomenological properties on hardware, ac-

ceptance that a computer can be creative implies a *de facto* rejection of dualism on the grounds that the machine cannot imaginably be partly located in some immaterial mental space. A tautology emerges by which a positive result for computational creativity is dependent on precisely the reductionist premise that it will hopefully be used to prove.

Rejection of dualism and of the corollary representations which inhabit a placeless mental space, on the other hand, do not necessarily entail an acceptance of the idea that computers can participate in the same kind of creative meaning-making as humans. Indeed, a notable trend in contemporary cognitive science is a move away from the idea that symbolic approaches to the mind can have anything to do with cognition at all, as characterised by the work of Noë (2004), Rowlands (2010), and Chemero (2009). This unfolding movement in the theory of mind traces its roots back to the enactivism of Varela, Thompson, and Rosch (1991) and to the ecologically situated psychology of Gibson (1979): these traditions seek to embed the thinking organism in a physical environment from which the processes underlying consciousness cannot be isolated. In terms of building creative machines, this bodily, environmental approach seems to indicate something more like robotics than the traditional conception of computational creativity as involving the algorithmic construction and traversal of abstract informational state spaces per Boden (1990).

Hence, if computational creativity is to be used as a tool for talking empirically about philosophical questions, the burden of proof shifts onto demonstrating somehow or another that information processing systems can behave creatively in the first place. If this can be done, then it seems likely that an analysis of the specific types of systems which generate creative output might yield some interesting philosophical insights into the nature of cognition. But as will be illustrated in the next section, the evaluation of computational creativity is not by any means a straightforward issue.

Evaluating Symbol Manipulating Systems

The problem of evaluation is a significant aspect of Boden's (1990) classic treatment of computational creativity, where it is argued that in order for computer generated artefacts to be considered as creative output, the program that generated them must likewise be judged as somehow creative in its procedures. In Wiggins' (2006a) subsequent formalisation of Boden's model, the creative agent itself is bestowed with an evaluative function which it uses to assess its own output, effectively building a sense of creative value into the agent's procedure. Ritchie (2001) has likewise formally described the operation of creative systems in terms of an "inspiring set" of known good artefacts of a certain type: this set becomes both the basis for the way the system will structure its own output and the index beyond which the system must extend itself in order to be considered creative, in a process which involves sequences of self-evaluation moving from a basic set of possible items, through a consideration of the inspiring set, to the output of artefacts which are hopefully both new and valuable.

In more recent work, Ritchie (2007) considers the merits of the view that the creativity of a system should only be considered in terms of its output. Part of Ritchie's reasoning is that human creators are generally only judged on the basis of what they do, not how they do it. On the surface this might seem to be in line with Wiggins' definition of computational creativity in terms of "behaviour exhibited by natural and artificial systems, which would be deemed creative if exhibited by humans," (Wiggins, 2006b, p 210). In fact, though, it would be a mistake to take "behaviour" here in the Skinnerian sense of observable responses to stimuli; what is really in question in terms of behaviour is the way in which the agent goes about making the artefact. And finally, Gervás (2010) has proposed a model for creative output that involves cycles of production and reflection on the work in progress. This is again ostensibly in a similar vein to Ritchie's chain of evaluation of different stages in an overall creative process, but Gervás, in support of the significance of procedure, actually specifically suggests that it is perhaps misguided to try to build systems to appear operationally like creative humans, reasoning that there are a multitude of engineering solutions for a given objective, and blind imitation is rarely the best approach.

From these stances a range of approaches to evaluation emerge, aligned along two main axes: on the one hand, there is the problem of whether or not the system should be considered in terms of its internal workings, and on the other hand, there is the question of whether or not the system should attempt to be humanlike. But establishing what exactly counts as a creative process in the first place has proven extremely difficult. Where human creators are easily forgiven for keeping their methodologies secret – where, indeed, the mysteriousness of creativity is enshrined in humans through terms such as "genius" – such vagary is deemed unacceptable in a computer. The problem at least partially lies in the question of precisely where the act of meaning occurs: can a computational system really make meaning, or is it the observer who gives meaning to output which is merely the result of informational shuffling? In particular, a problem arises in terms of defining what counts as internal with regards to an information processing system. Given that the operation of a symbol manipulating machine is based on an interpretation of symbols which is fundamentally relative to a subjective observer (Putnam, 1988), the idea of a computational system being anything other than observable seems to fall apart, in which case everything that the actual system does can only be construed as output. If this is the case there is at least an argument to be made from a philosophical perspective for Ritchie's (2007) view that changes in the system's process must themselves be viewed as output in order to be assessed.

One practical approach to resolving these issues of evaluation has been formalised by Colton, Charnley, and Pease (2011), who, through their FACE model, propose a four step process for generating creative artefacts, or, in their terminology, expressions of concepts. Crucially, this process involves the establishment of framing information that potentially contextualises or justifies corresponding generative acts. The FACE model is complemented by the IDEA model, a framework specifically designed for the evaluation of creativity, both in terms of artefacts and actions. In an

explication of the theories behind these models, Pease and Colton (2011a) are motivated by an appreciation of the tensions that arise between creators and observers in the course of creative generation and evaluation, and seek to place the generation of new meaning in this dynamic relationship. By grounding the context of meaningful expression in public information, the hope is that the problem of trying to conceive of mechanical systems with internal states might be resolved.

The FACE model has been implemented by Colton, Goodwin, and Veale (2012): the output of the system developed by these authors offers, in conjunction with new poetry, a narrative alleging motivations on the part of the system in the course of poetic production. Furthermore, this narrative is grounded in an analysis of sentiments and concepts found in an external source, namely, in newspaper articles from a chosen date. This reflexive procedural commentary is specifically motivated by the view that observers do take into account creative process when evaluating an artefact, a stance which is also expressed by Colton (2008) in earlier theoretical work. By ascribing a phenomenology, however implausible, of intentions and emotions to the computational agent, the system generates a secondary level of artifice wherein the artefact is the result of some process of conceptualisation, representation, and execution. The hope is that humans will associate a capacity for creativity with the impression of intentionality.

What seems to be happening here is the simulation of precisely those properties of internal mental states that, as discussed in the previous section, have been attacked by contemporary cognitive scientists and philosophers of mind. Despite this, the stance taken in this paper is that this type of simulation is, broadly, the correct approach to take towards the evaluation of creativity—an evaluative act which, looking at it from the other end of the equation, might as easily be described as persuasion on the part of the agent. However, the stance here is also that mere mimicry of phenomenology is not ultimately a compelling argument for the creativity of a system. Rather, what is needed is a system that legitimately instantiates mechanisms with some similar properties to those that result in the appearance of mental states in cognitive agents.

In their zeal for non-representationalist, anti-dualist theories of mind, the contemporary mode of environmentally oriented approaches to cognition have arguably overshot the philosophical mark: not only do they reject the Cartesian stance; their rejection is so thorough that they neglect to properly consider why the mind/body divide has preoccupied Western thinkers for so long in the first place. But the appeal of the idea of an inner life of the mind is powerful on a collective level, running so deep in society that it has been instantiated in the form of intellectual property law, whereby authorship is ascribed on the basis of an ill defined "creative spark," (Fei, 1991). Indeed, in a legal sense, and therefore also to some degree on the scale of society, ownership of expressions is construed in terms of the distinction "between creation and discovery," (ibid). Elsewhere, McGregor (2014) has proposed that intellectual property law itself might be considered as one viable mechanism for the evaluation of creativity, and that something in the creative process or artefact might be offered up to appease the law's requirement for a distinguishing creative aspect. This is a problematic stance for the prevalent model of computational creativity, which, again per Boden (1990), involves a combinatorial exploration of a well defined state space, where the artefacts of such an exploration must be construed as discovered rather than generated. If the computational agent is to be presented as creative on a social level, then, it would seem the only course of action is to somehow trick the public into thinking of certain informational manipulations as being somehow inherently mental.

The idea of trickery isn't totally new to the field. In particular, where the theoretical work of Colton (2008), like that of Gervás (2010), plainly states that the computational agent should be straightforward about its own nature, the practical implementation of Colton, Goodwin, and Veale (2012) develops an agent that sets about selling its own product with an appeal to intentionality which might almost be described as deceptive. Similarly, albeit in a different domain, Leymarie and Tresset (2012) have designed an ingeniously conceived robotic portrait artist that is programmed to simulate behaviours the programmers have determined sitters and onlookers expect to witness in human artists: the robot enacts a roving quality to its video-camera eye, accompanied by built in pauses which create the illusion that the device is contemplating its work. The deception here, though, is transparent, and is committed with the good faith of honest artistry: it is unlikely that many observers believe these processes, which in the cases in question involves prefigured semantic networks and sentiment analysis or else an encoded parroting of creative behaviour, actually build up any kind of intentionality prior to the production of the output. Even a philosophically disengaged observer should not be expected to accept that phenomenology and intentionality can arise simply through the application of preconceived frequentist methods of data interpretation, or through the robotic rehearsal of a choreographed sequence of stereotypical gestures.

So it is proposed here that observers look for familiar processes when analysing creativity, but that this familiarity should be on the level of the impression of developed internal mental states rather than just superficial expressions; it is further proposed that the right approach to building creative agents is therefore to construct systems which remit the appearance of some kind of internal representations which are developed and manipulated in the course of searching for new, interesting artefacts in any given domain. The claim will be that, in such systems, while the base level of artefacts output for a target domain may be considered simply discovered within the search space chosen by the agent, creativity happens in terms of shaping the search space in the first place, not in terms of the subsequent traversal of that space, an idea which lines up nicely with Boden's (1990) notion of high level transformational creativity. Of course, this attempt to move creativity up a level, so to speak, suggests a secondary search space for new search spaces. Ritchie (2007) touched on this idea when he suggested that the creative process itself should be considered an abstract artefact

of the system, but what emerges is an infinite regression of spaces of spaces which immediately calls to mind the parallel homunculus problem in the philosophy of mind. This well travelled argument against representationalist theories of mind questions the basis for a secondary observation of mental representations by some internal observer – a homunculus, so to speak – an evidently necessary and likewise confounding condition for mind/matter dualism (see Dennett, 1991).

And this is precisely the point: entertaining this approach to the evaluation of computational creativity, namely, the consideration of an agent as being composed of a recursive hierarchy of creative search spaces, results in the same kind of untenable scenarios which characterise the dualist world view. The Cartesian outlook begs the question of who or what is perceiving internal mental states, and, more pointedly, suggests that these internal observations must likewise yield to some form of dualism, setting off a concatenation of ever deepening layers of internal states with no explanation of how this chain could terminate. In the same way, suggesting that a system becomes truly creative when it actually changes the parameters for discovering new and valuable artefacts necessitates a secondary search space with some sort of overview of the primary space from which it might seek the appropriate transformations; this secondary space, however, immediately becomes subject to the same criteria for transformation as the primary space, and an infinite regression rapidly develops. By this untenable mechanism of an infinite hierarchy of spaces in a finite system, a deeper operational correspondence between dualist theories of mind and transformational theories of computational creativity emerges. When the external impression of phenomenology is constructed by merely using information processing systems to analyse input and then combine indexical terms into the semblances of intentions and emotions, the impression of creativity can only be ephemeral. When a system actually reveals that it is operating in a way which establishes the kind of conceptual structures and recursive levels of abstraction associated with what is popularly, if erroneously, considered to be the dualist nature of cognition, on the other hand, the system has much more of a chance of being considered autonomously creative. The question, then, is what exactly qualifies as a semblance of dualist operation in a symbol manipulating system.

Implementing Representations

The next section of this paper will briefly consider two emerging computational models in terms of their potential as operational frameworks for computationally creative systems. Both vector space models and deep belief networks have been developed for the purpose of computing with high-level conceptual structures, and each system has been at least somewhat successful in its applications to specific informational domains. The question addressed here is whether the operation of either of these systems is such that they might be considered to produce the same kind of structures which observers imagine correspond to the mental representations attributed to the immaterial minds of humans under a dualist world view.

The hope is that these systems might show some promise in generating computationally graspable conceptual structures which can play a part in the act of meaning: more than just arrangements of data, these conceptual entities would stand for processes to be performed on data, abstract actions in the symbolic world of the computer, realised only through observation. The development of these systems for creative, generative purposes, however, is left for the future.

Vector Space Models

Initially developed as a mechanism for document indexing (Salton, Wong, and Yang, 1975), vector space models are built of high dimensional spaces whose dimensions correspond to the relational terms associated with a linguistic object: the object is described on the basis of the frequency with which each of the dimensional terms occurs in its context, and thus can be represented by a vector in the space. The idea is that similarity between two objects represented in such a space can be interpreted from the degree of the cosine angle between their corresponding vectors. In more recent work, vector space models have been applied to more basic problems of meaningfulness through distributional models of language, where words are represented in terms of their context and in particular through vectors representing either the frequency or the probability with which they occurred in the context of other words. This approach has been used to attack problems such as word disambiguation (Schütze, 1998) and compositional semantics (Mitchell and Lapata, 2008; Coecke, Sadrzadeh, and Clark, 2011).

The compositional approach in particular has revealed the utility of the mathematical nature of the vector space models. As illustrated in Grefenstette and Sadrzadeh's (2011) implementation of Coecke, Sadrzadeh, and Clark's (2011) framework, the properties of these kind of high dimensional representations allow for the composition of new representations through the use of Kronecker products, a technique which, by virtue of its non-commutativity, produces different spaces even for different combinations of the same words—a desirable outcome, given that word order can make a significant contribution to meaning in a sentence. This feature of vector space models allows for the construction of increasingly complex spaces as words are incrementally built into phrases and then sentences. The result is a system containing a vocabulary, so to speak, of highly modular compositional elements: the spaces of words can be easily concatenated into larger meaningful elements on the level of sentences, which become spaces themselves through the mathematical operations which can be performed on these types of structures.

In terms of computational creativity, what emerges from the perhaps somewhat complicated mathematics of vector space models is a mechanism for possibly representing what Davidson has described as "meanings as entities," (Davidson, 2001, p 116): the raw data of language become objects that can interact in ways that might produce valuable, surprising new semantic combinations. This approach to the composition of conceptual structures abstracts the problem of semantics away from the level of data processing, and likewise away from ungainly interventions of word associa-

tions and semantic ontologies that leave an observer wondering if the real creativity hasn't been imposed on the system through a preconceived framework. Instead, by generating and manipulating representations with operations that seem far removed from the logic of mental states or the syntax of the source language, a vector space is effectively promoted to the same level as the meaning-rich kind of encounter that humans have with the world and seems to thereby manifest some of the same mysteriousness associated with that way of being. Rather than relying on an externally grounded observation to give a system of symbols meaning, the objects that populate vector spaces can interact in ways native to their abstract mathematical domain, and in so doing instantiate entities that at least can be construed as conceptual representations analogous to the internal imagery of the Cartesian mental space.

While the use of vector space models for creative purposes remains unexplored, the indication from the work done in text analysis gives grounds for proposing that this could be a good method for likewise compositionally building linguistic artefacts which meet the constraints of a creative search space. And, importantly in terms of the subject of this paper, there seems to be good reason to hope that these conceptual structures might stand a chance of convincing a sceptical observer that a system employing them creatively could be utilising something similar to the types of internal representations which have been associated with the human use of language and the human mode of thought, per the likes of Chomsky and Halle (1968) and Fodor and Pylyshyn (1988).

Certain other systems have, in fact, taken a generative approach to vector spaces. The latent Dirichlet allocation model (Blei, Ng, and Jordan, 2003) is in particular a topic modelling technique that discovers topics within a range of documents and then builds a probability distribution for words across these topics. Latent Dirichlet allocation is generative in the sense that it picks potential words based on a probability distribution over a topic: the distribution of topics across a potential documents suggests likelihoods for the words which might occur in that document, albeit without the word ordering critical to a meaningful use of language. This is not necessarily an ideal strategy for modelling creative behaviour, however, as, in addition to the absence of compositionality, generative models tend to predict output that is highly likely but, conversely, not very surprising. In the context of generating meaningful and unexpected new language, the compositional approach discussed above seems to hold more promise for finding the semantically loaded output expected from a creative agent.

Deep Belief Networks

Where vector space models have proved particularly powerful for language, deep belief networks have been used effectively for work in the domains of both computational linguistics and computer vision. Deep belief networks were proposed by Hinton, Osindero, and Teh (2006) as high parameter frameworks that would learn to identify handwritten numerals by developing a model for generating the same artefacts. In this case the generative quality of deep belief networks do specifically point to a creative application, in that the network actually learns to match new, noisy percepts with semantically tagged representations by actually learning to produce those representations in an initial stage of development. Across its many levels of processing, the network purportedly develops different layers of feature detection, and these features – for instance, lines, contours, or, eventually, at a high level, concepts – arguably convey the impression of the internal states corresponding to the mental perception of properties in the world.

The idea is that densely connected networks consisting of a large number of artificial neurons rising over several diminishing layers in a pyramid type structure can be efficiently and effectively trained if they are constructed with the right kind of architecture. The keys to this architecture are a special mechanism at the low level related to Ackley, Hinton, and Sejnowski's (1985) earlier work on Boltzmann Machines (another type of neural net that utilises a stochastic mechanism), as well as the simplicity with which the connecting weights between neurons are updated. With their highly interconnected structure, deep belief networks might be seen as the next phase in the historic cycle of interest in connectionist approaches to computing; the new element in this latest manifestation is the stacking of several operational layers where parameters are established in a layer by layer fashion.

The operational key to deep belief networks is the idea that, by allowing a single neuron on a higher level to represent the clusters of neurons which feed into it from the level below, an exponential reduction of computational space can be realised (Bengio, 2009). In this way, these networks establish elevating levels of abstraction that might be construed as internal representations. Indeed, in precisely this sense, deep belief networks seem to relate to the idea of the act of meaning by which potentially diffuse visual data are resolved into higher level percepts with some semantic value. The argument put forward here is that this on the one hand instantiates the approach to cognition through the creative reconstruction of anticipated events in the world endorsed by Wiggins (2012), and, on the other hand, creates structures which might be recognised by an observer as something similar to internal mental states.

Going back to some of the original literature from the first wave of neural networks, the structure of the human brain was clearly a primary motivation in the effort to compute using weighted networks of nodes (McCulloch and Pitts, 1990). Deep belief networks have inherited this property, and have taken inspiration from another aspect of neuroscience: the multiple layers in a deep belief network specifically resemble the hierarchical structure of the visual cortex in the human brain (Bengio, 2009). Indeed, Serre et al. (2005) have done work towards isolating the ways in which different levels of the primate visual cortex build up different aspects of representations of raw visual stimuli, ultimately resulting in the high level perceptions of parametrically bound entities which seeing, thinking agents experience in the world. In the same way, deep belief networks seek to use increasingly complex clusterings of input data to form higher levels of representation within their architec-

ture. Coupled with the fact that these systems are fundamentally generational, such networks seem like an excellent candidate for consideration as visually creative agents with a convincing impression of internal representations, and probably warrant exploration in other domains, as well.

Conclusion

Dualism was born of a simple thought experiment: Descartes (1911) imagined himself plagued by a demon fixated on deceiving him, and in response strove to strip away from his experience of reality everything which could possibly be considered illusory. He was left with the certainty of his own irreducible mental existence, but maintained that this existence must also be involved in some sort of likewise irreducible physical reality. Since even before Descartes' time, various similar imaginative exercises have characterised the development of Western philosophy, from Plato's (1892) cave to Wittgenstein's (1967) beetle. Notable recent thought experiments seeking insight into the mind have included Putnam's (1996) twin earth, Davidson's (1987) swampman, and Chalmers' (1996) philosophical zombies—and, as mentioned earlier, the computer has played a part in some other recent enquiries, though generally as a device for demonstrating the absurdity of certain views of cognition that can be reduced to mere data shifting.

The purpose of pointing out this tradition of thought experiments is to highlight the role which the peculiar act of introspection has played in the development of modern Western philosophy. The preoccupation with intentionality and phenomenology have grown out of an intellectual culture of examining the self, and the willingness which humans have to accept the creativity and indeed the very meaningfulness of the expressions of other humans seems to stem from the recognition of a similarly calibrated other-self. What has been proposed in this paper is that the external alienation of an encounter with a computational system can be replaced with a look into the exposed operations of the system, and, in this exposure, there may be some hope of acceptance that the symbol manipulating machine is behaving in a way which is creative, in the operational sense of behaviour described by Wiggins (2006b).

The idea that information processing systems should be investigated for indications of familiar processes in order to be considered creative is not new. Gervás (2010) has argued that hardware which operates in a highly parallel manner should be taken more seriously as a candidate for instantiating creative agency, as this type of procedure to some degree mirrors the evident dispersion of activity in the human brain. Perhaps even more fundamentally, Pease, Winterstein, and Colton (2001) call for a criterion of procedural complexity intended to measure the extent of the creative search space and the difficulty of the agent's traversal of this space. It is not clear, however, how such mechanisms don't become just another aspect of the agent's output, adjunct to the creative artefact itself. What is called for in this paper is a probing of the machine – an extrospection, so to speak – for the representational type of processes that society at large seems to deem, in the tradition of Descartes, should count as cognitive and potentially creative. It is for the observer to seek out and identify the structures which form these representations rather than for the system to simply present them either though a statement of intentionality or an exposure of process.

What form these representional structures would take remains to be defined, though two possible candidates have been proposed here. A further area for enquiry is the question of what kind of observer would be able to recognise these structures in the first place: is some combination of expertise in philosophy and computer science necessary in order for a computationally creative agent to be recognised as such? Ideas along these lines have been proposed by Pease and Colton (2011b) and Boden (2014), all of whom suggest that computational creativity may be best judged by an audience with a degree of knowledge about how computers work. On the one hand, the idea of expert criticism informing the public as to the value of creativity has long been common in various domains such as art, literature, and film, and some degree of expertise is probably necessary to achieve recognition of the relatively complex frameworks discussed earlier in this paper. On the other hand, relying on computer scientists for assurances of the legitimacy of creative agents risks further alienating an audience already confronted with a very new and different mode of creation, and, indeed, of creator.

So even the proposal for a solution to the problems laid out in this paper seems to open the door on another potential debate. Such is the nature of philosophy. Nonetheless, this paper has sought to show that computational creativity as a field is an appropriate platform for engaging in discussions about not only aesthetics but also cognition and theories of mind, and has at least presented an avenue for further philosophical investigation.

Acknowledgements

This research has been supported by EPSRC grant EP/L50483X/1.

References

Ackley, D. H.; Hinton, G. E.; and Sejnowski, T. J. 1985. A learning algorithm for Boltzmann machines. *Cognitive Science* 9(1):147–169.

Bengio, Y. 2009. Learning deep architecture for AI. *Machine Learning* 2(1):1–127.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Boden, M. A. 1990. *The Creative Mind: Myths and Mechanisms*. London: Weidenfeld and Nicolson.

Boden, M. A. 2006. *Mind as Machine: A History of Cognitive Science*. Oxford: Clarendon.

Boden, M. A. 2014. Skills and the appreciation of computer art. In *Proceedings of AISB14CC*.

Chalmers, D. J. 1996. *The Conscious Mind*. Oxford University Press.

Chemero, A. 2009. *Radical Embodied Cognitive Science*. Cambridge, MA: The MIT Press.

Chomsky, N., and Halle, M. 1968. *The Sound Pattern of English*. New York: Harper and Row.

Coecke, B.; Sadrzadeh, M.; and Clark, S. 2011. Mathematical foundations for a compositional distributed model of meaning. *Linguistic Analysis* 36(1-4):345–384.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA models. In *Proceedings of the International Conference on Computational Creativity*.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-FACE poetry generation. *Proceedings of the Third International Conference on Computational Creativity* 95–102.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *Proceedings of the AAAI Spring Symposium on Creative Intelligent Systems*.

Davidson, D. 1987. Knowing one's own mind. In *Proceedings and Addresses of the American Philosophical Association*, volume 60, 441–458.

Davidson, D. 2001. Truth and meaning. In Martinich, A. P., ed., *The Philosophy of Language*. 114–124.

Dennett, D. C., and Searle, J. R. 1995. 'The Mystery of Consciousness': An Exchange. *The New York Review of Books* 42(20).

Dennett, D. C. 1991. *Consciousness Explained*. London: The Penguin Press.

Descartes, R. 1911. *The Philosophical Works of Descartes*. Cambridge University Press. Translated by Elizabeth S. Haldane.

1991. Feist Publications Inc. v. Rural Tel. Service Co., 499 U.S. 330.

Fodor, J. A., and Pylyshyn, Z. W. 1988. Connectionism and cognitive architecture: A critical analysis. *Cognition* 28(1-2):3–71.

Gervás, P. 2010. Engineering linguistic creativity: Bird flight and jet planes. In *Proceedings of the NAACL HLT 2010 Second Workshop on Computational Approaches to Linguistic Creativity*, 23–30.

Gibson, J. J. 1979. *The Ecological Approach to Visual Perception*. Boston: Houghton Miffline.

Grefenstette, E., and Sadrzadeh, M. 2011. Experimental support for a categorical compositional distributional model of meaning. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Hinton, G. E.; Osindero, S.; and Teh, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural Computation* 18(7):1527–1554.

Leymarie, F. F., and Tresset, P. 2012. Robot drawing and human engagement. In *Proceedings of the 5th International Workshop on Human-Friendly Robotics*.

McCulloch, W. S., and Pitts, W. H. 1990. A logical calculus of the ideas immanent in nervous activity. In Boden, M. A., ed., *The Philosophy of Artificial Intelligence*. Oxford University Press.

McGregor, S. 2014. Considering the law as an evaluative mechanism for computational creativity. In *Proceedings of AISB14CC*.

Mitchell, J., and Lapata, M. 2008. Vector-based models of semantic composition. In *Proceedings of ACL-08:HLT*, 236–244.

Newell, A., and Simon, H. A. 1990. Computer science as empirical enquiry: Symbols and search. In Boden, M. A., ed., *The Philosophy of Artificial Intelligence*. Oxford University Press. 105–132.

Noë, A. 2004. *Action in Perception*. Cambridge, MA: The MIT Press.

Pattee, H. H. 2008. Physical and functional conditions for symbols, codes, and languages. *Biosemiotics* 1(2):147–168.

Pease, A., and Colton, S. 2011a. Computaional creativity theory: Inspirations behind the FACE and IDEA models. In *Proceedings of the International Conference on Computational Creativity*.

Pease, A., and Colton, S. 2011b. On impact and evaluation in computational creativity: A discussion of the turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and Philosophy*.

Pease, A.; Winterstein, D.; and Colton, S. 2001. Evaluating machine creativity. In *Proceedings of ICCBR-2001*.

Plato. 1892. *The Republic*. Oxford University Press.

Putnam, H. 1988. *Representations and Reality*. MIT Press.

Putnam, H. 1996. The meaning of "meaning". In Pessin, A., and Goldberg, S., eds., *The Twin Earth Chronicles: Twenty Years of Reflections on Hilary Putnam's "The Meaning of 'Meaning"'*. Armonk, NY: M.E. Sharpe. 3–52.

Ritchie, G. 2001. Assessing creativity. In *Proceedings of the AISB Symposium on AI and Creativity in Arts and Science*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Rowlands, M. 2010. *The New Science of the Mind*. Cambridge, MA: The MIT Press.

Salton, G.; Wong, A.; and Yang, C. S. 1975. A vector space model for automatic indexing. In *Proceedings of the 12th ACM SIGIR Conference*, 137–150.

Schütze, H. 1998. Automatic word sense discrimination. *Computational Linguistics* 24(1):97–123.

Searle, J. R. 1990. Minds, brains, and programs. In Boden, M. A., ed., *The Philosophy of Artificial Intelligence*. Oxford University Press. 67–88.

Serre, T.; Kouh, M.; Cadieu, C.; Knoblich, U.; Kreiman, G.; and Poggio, T. 2005. A theory of object recognition: Computations and circuits in the feedforward path of the ventral stream in primate visual cortex. Technical report, MIT Computer Science and Artificial Intelligence Laboratory.

Sloman, A. 1978. *The Computer Revolution in Philosophy: Philosophy Science and Models of Mind*. The Harvester Press.

Varela, F. J.; Thompson, E.; and Rosch, E. 1991. *The Embodied Mind*. Cambridge, MA: The MIT Press.

Wiggins, G. A. 2006a. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19:449–458.

Wiggins, G. A. 2006b. Searching for computational creativity. *New Generation Computing* 24:209–222.

Wiggins, G. A. 2012. The mind's chorus: Creativity before consciousness. *Cognitive Computing* (4):306–319.

Wittgenstein, L. 1967. *Philosophical Investigations*. Oxford: Basil Blackwell, 3rd edition. trans. G. E. M. Anscombe.

Is it Time for Computational Creativity to Grow Up and start being Irresponsible?

Colin G. Johnson
School of Computing
University of Kent
Canterbury, Kent, UK
C.G.Johnson@kent.ac.uk

Abstract

A recent definition of computational creativity has emphasised that computational creativity systems should "take on certain responsibilities" for generating creative behaviour. This paper examines the notion of responsibilities in that definition, and looks at a number of aspects of the creative act and its context that might play a role in that responsibility, with an emphasis on artistic and musical creativity. This problematises the seemingly simple distinction between systems that have responsibilities for creative activity and those which support or provide tools for creativity. The paper concludes with a discussion of an alternative approach to the subject, which argues that the responsibility for creative action is typically diffused through a complex human/computer system, and that a "systems thinking" approach to locating computational creativity might ask better questions than one that tries to pin creative responsibility to a particular agent.

Introduction

A recent paper by Colton and Wiggins (2012, p21) gives a succinct definition of computational creativity as "the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative. Compared to earlier attempts to define this area, this definition is notable because it does not define computational creativity with regard to human creativity.

By contrast, earlier definitions have been grounded in comparisons with human creative behaviour. For example, Ritchie (2007, p69) grounds his list of criteria for attributing creativity to a computer program thus: "A central assumption here is that any formal definition of creativity must be based on its ordinary usage; that is, it must be natural and it must be based on human behaviour.". Furthermore, an earlier overview by Colton, de Mántaras and Stock (2009, p11) begins with the statement that, "At its heart, computational creativity is the study of building software that exhibits behavior that would be deemed creative in humans.".

In this paper I will explore a specific phrase in the definition— "taking on particular responsibilities" which is the main difference with previous definitions. I would like to explore where these "particular responsibilities" might sit in the creative process, and how the use of computers might change our idea of where that responsibility might sit. In particular, my focus will be on artistic and musical creativity, though there may be implications for other creative areas.

Who/what is "responsible" for a particular creative artistic act? We can argue that there are a number of things that share this responsibility (here we frame these in the context of a human artist):

- The artist themselves, their actions and patterns of behaviour.
- The artist's motivation to create the work.
- The background knowledge that the artist has acquired through life, which reflects their general cultural background and specific things that they have encountered or learned.
- The context in which they are making the work.
- The materials that they are using to make the work. In particular, the "resistance", "grit" and "grain" offered by some materials, which can provide new material that can be serendipitously exploited by the artist.

It is commonly seen as the first of these as the action that takes on "responsibility" for the artistic creation. However, when we try to pin down why this is so, we might start by arguing that had the artist not decided to carry out that particular behaviour, to decide not to create that particular work, then the work would not exist. But, the same argument can be applied to the other items on the list: had the artist not had the relevant background knowledge, or had the material worked in a different way, and so on, the work would not have been capable of being created. We can, of course, take this argument to ludicrous extremes—part of the responsibility for the art being the artists own existence, etc.

Indeed, this is not just an intellectual exercise; determining the responsibility (or credit) for a creative act is important for legal arguments concerning intellectual property rights. McGregor (2014) has recently argued that the legal arguments around creativity might provide a framework for considering computational creativity; along similar lines, Koza (2010) has argued for the use of patentability as a criterion to determine when an AI system is creating artefacts that require "human-competitive" levels of intelligence.

We might stop at a "proximate" cause as being the primary point at which the responsibility lies. But, what is the proximate cause? We might argue that the immediate actions of making the art are the artist's behaviours, in putting pencil to paper in a particular way. But, even at such a proximate level, we can see that that activity interacts closely with the artist's motivation, and that during the time-span of creating even a single, simple piece of work there might be a complex interaction between motivation and action.

So, where is the computer in all of this? I have argued elsewhere (Johnson, 2012) that computational creativity research has focused too much on the role of the individual creator, favouring the view of the creative "romantic hero" over forms of creativity that are based on collaboration or the mediation of interaction. In this paper I would like to argue further that the nature of computer-grounded artistic creativity makes assigning this responsibility even harder than it would be for traditional artforms.

The remainder of this paper splits into three sections. The first is concerned with the role of materials, and in particular whether computational artistic and musical materials present a particular challenge for making the distinction between passive tools/materials and active agents to which creative responsibility can be ascribed. The second is an examination of context and background, and considers, through examples of search-based art and semantic mass, whether these can be considered to have any responsibility for the creative action. Finally, a concluding section examines whether a better way to examine this is through a "systems thinking" view, rather than a view based on the notion of responsibility.

Materials

What is an artistic material, or an artistic tool, and how does it differ from something that plays a collaborative role in artistic creation? For traditional artworks, the distinction is clear. For example, an artist uses a tool such as a pencil to create their art, a musician uses an instrument to create a piece of music. Part of an artistic training is to learn to "master" such tools; to learn how to realise artistic intents through the coordinated use of perception, thought and the manipulation of tools. But, even at the level of simple physical tools, there is some level of interaction between the tool and artistic creation—part of the study of a particular artistic medium is learning its constraints, and learning how to make adaptations when a particular intentional action does not realise the intended aim.

Certain computational artistic media and tools make this distinction between passive tools and media that are manipulated by an artist or musician, and participants that take an active (creative?) role in the artistic creation. Rowe (1993) has discussed a continuum of interactive computer music systems, ranging from simple action-response systems where a performer makes a physical gesture and generates a consistent sound, to systems which listen and make sound as an autonomous and equal participant in a musical interaction. An example of the latter might be the *Voyager* system (Lewis, 2000); these ideas have been taken further by Paine (2002).

We would probably consider something to the latter end of the continuum to sit comfortably within definitions of computational creativity such as that discussed at the beginning of this paper. Whilst a system of that kind might always perform within an interactive context with other (human or computer) performers, it is "responsible" for holding up its own end in the music being produced, and "creating" music that is sensitive to the current situation. There are multiple "responsible" agents involved, and nothing playing the role of "mere tool".

But, when we look at systems towards the middle of that continuum, the allocation of creative responsibility becomes murkier. For example, consider the LIES system by Sanfilippo (2012). This consists of a number of acoustic feedback loops, which initially create sounds by creating positive feedback cycles that can start from tiny fluctuations in the performance environment. These are modified by a large number of digital filters and feedback networks. The performer interacts with this system by adjusting the parameters of the various filters and intensity of the feedback system.

What is *responsible* for the final creative output in this system? The interaction between human and machine is complex and at times incomprehensible to the human; the performance mode is one where the human sometimes tries to control the sound being generated to bring it into line with a desired sound (sometimes successfully, sometimes not), sometimes just letting the sound unfold without interference, and sometimes to explore the effect of parameter changes with, depending on context, a greater or lesser understanding of the likely effects. Certainly, the system generates a decent amount of the creative material here, with the human sometimes (importantly—not all of the time) being unable to shape the system's outputs in any comprehensible way.

The systems view of this creativity is articulated well by the creator of the system: "...the human and machine are considered as inseparable: two autonomous entities which, unavoidably, will influence each other, creating a unique meta-system made up of these two elements. The human and the machine establish a dialectics, a *talking through the other*, with no attempts of subordination, creating a performance which is the result of their cooperation, where, thus, the performer creates *together* with the machine." (Sanfilippo, 2012).

Is this a computational creativity system? All of the sounds are coming from the system (but, the same is true for a piano). The human would not be able to make the work without the machine (but, the same is true of an artist without a pencil or whatever). Nonetheless, the computer/electronic system seems to be playing a stronger creative role in this interaction than that. Perhaps part of this is that the human is sometimes reacting to the outputs of the computer system as much as they are trying to shape it.

Contexts and Background Knowledge

In the list towards the beginning of the paper, we identified the background knowledge of the creator, and the context in which they were working in, as other things that could form part of what is responsible for a particular creative action or outcome.

Where is this background knowledge in a computational creativity system? In many cases it has been included as a part of the basic architecture of the system: for example, in Cohen's AARON system (Cohen, 1995), its figurative works are generated from parameterised algorithms that describe the basic figurative structures that are used to create the work. Other work draws on internet search algorithms as a way of accessing a background of knowledge (Johnson, 2013).

Can a way of accessing information, enabled by a technology, become part of the creative responsibility that a computer system provides to a creative activity or outcome? To what extent does the choice to use a complex, unpredictable computational technique in creating a work of art mean that that artwork has had a creative contribution from the computational system?

Let us consider a specific example. The image in Figure 1 is created by using the well-known Google image search functionality to search for images related to the word "secure" (filtered for images of a certain colour palette). If I choose to exhibit this as an artwork, where does the responsibility for the creative decisions sit? With me, alone? But I have hardly done anything! With the Google information retrieval system? With the people who have provided images for the system?

One role that computer systems—not just individual computers, but networked collections of computers with an associated infrastructure of information gathering and information retrieval—might play is to facilitate whole new areas of creativity. For example, the existence of vast online collections of images, together with technology of evergrowing sophistication to search and group such images by their meanings, facilitates a way of creating artworks that we might describe as *semantic mass*, where large collections of related information are gathered together and displayed.

Consider an example such as Jennifer Mills's work *What's in a Name?* (Figure 2), consisting of a large number of postcard-size paintings, each of which represents a person with the name "Jennifer Mills", gained from a search on Facebook. Is this work an artist's reflection on the ready ability to track down all of these people using the computer system, or is this a piece of collaborative creativity between the artist and that system? Even if it is not, does the system bear any "responsibility" for the artwork—any more than the paintbrush used to create the work?

Manovich (2002) has made related observations, that a technology can, by facilitating a change in the speed or scale of a process, create something which an observer might see as a genuinely new system. This can be seen by contrasting the Mills piece with comedian Dave Gorman's pre-web-search project *Are You Dave Gorman?*, where he tracked down a large number of people with the same name has him (Gorman, 2001). The Gorman project is focused on the labour of making the connections; the Mills piece on its effortlessness.

A number of artists and musicians have chosen to deliberately divest themselves of the responsibility of making creative choices in their art. Perhaps the best known of these is John Cage, who created musical/theatrical works based on chance processes or on transcoding (Manovich, 2002) non-musical objects. An example of the latter is *Atlas Eclipitcalis*, where star-charts were transcribed onto music staff paper, with stars representing notes, and the resulting music performed. By refusing the composer's traditional "responsibility" to decide (at a detailed level) where the notes go on a page, where has "responsibility" for the artwork gone? Perhaps an argument can be made that the responsibility has been abstracted to a higher level—that the details of the notes "don't matter", but the choice of star maps, rather than any any other printed material, is where the creator has chosen to vest his responsibility. A version of this argument has been made by Xenakis (1992), presenting a form of music in which the composer manipulates large-scale parameters of generative algorithms, rather than details.

There is a connection too, to the ideas of Goldsmith (2011), who has discussed the idea of "ostensive creativity", i.e. a means of being creative by "pointing at" material in the world, or organising it in a way that makes us see it afresh. Internet search based art can be seen as a form of this. But, who is doing the pointing?

Again, we are drawn back to a system view of the idea of creative responsibility. All of these components have some bearing on the final creative activity, and it is their interactions that lead to creativity happening, rather than one playing a responsible role and the others a supporting role.

Conclusions

At first it seems easy to distinguish between a system that is a tool that can be used in the aid of creative action, and one that takes on the responsibility for the creative act itself. However, when we look at complex, resistant artistic materials, systems containing complex interactions between humans and computers, and the kinds of human creativity and relational creativity that depend irreducibly on computers or networks of computers, then the distinction between a responsible creative agent, a creativity support system, and the more complex kind of tool become rather blurred.

It is easy to understand why the idea of responsibility finds its way into a definition of computational creativity. There is always a sneaky suspicion in a system involving interaction between humans and computers that all of the creativity is "coming from" the human (even when that human demonstrates surprise at the output from the system!). There is also the desire to distinguish creative systems from "mere tools". It is fairly clear that this can be done, up to a point, but the point at which tools slip over from being passive to being an active player in the creative process is a rather vague one.

Indeed, it is precisely because computers can be used to build complex, interactive, indeterminate systems that this distinction starts to become more problematic. Indeed, it is perhaps naive to assume that even in traditional non-computational artistic and musical creativity that a simple distinction can be drawn between individuals responsible for their creative action and the tools and concepts that they make use of. After all, reams of pages are written that attempt to explain why a particular artistic action was done by

Figure 1: Google image search result for the word “secure”.

Figure 2: Extract from *What’s in a name?*, Jennifer Mills, 2009-11.

contextualising it in the political, economic and social situation in which it is created.

One alternative approach would be to apply a "systems thinking" approach (Churchman, 1968) to this question. This approach would argue that it is futile to try and assign a particular component of the "art creating system" the definitive responsibility for producing the art work. Instead, there is a complex system of interacting agents and properties that lead to the work being realised (or not!) in the form that it ends up. By doing this, we are not throwing our hands in the air and saying that nothing can be said about how the work is produced. Instead, we are arguing that there is a complex system of interactions which in itself needs to be studied. Indeed, Csikszentmihalyi (1988) has explored a similar approach to explaining human creativity.

Perhaps we can modify the Colton/Wiggins definition in the following way: "the philosophy, science and engineering of computational systems which, by playing a role in an interactive system, contribute to that system producing behaviours that unbiased observers would deem to be creative". Note that the systems have to "play a role" in the system; this opens up the possibility of many different possible roles.

This would seem to bring many activities that are currently seen as part of computational creativity squarely into the definition. For example, Veale (2011, 2013) has discussed the idea of creativity as a service, i.e. the provision of computational components that can are designed to be part of a larger creative system, glued together using web services frameworks.

The main point, however, is not to contribute to a pedantic (if sometimes enlightening) debate on definitions, but to shift the emphasis of computational creativity research. Rather than trying to identify the single actor in a complex, interactive system that is "responsible" for the creativity, instead we should recognise that this responsibility is diffuse and part of the behaviour of a complex human/computer system. That then leads onto much more interesting questions about how such systems gives rise to creativity, how components can be engineered for such systems, and how interactions in such systems can be managed, rather than searching for the single romantic hero who is the fount of all creativity in the system.

References

Churchman, C. W. 1968. *The Systems Approach*. New York: Dell Publishing Co.

Cohen, H. 1995. The further adventures of AARON, painter. *Stanford Electronic Humanities Review* 4(2).

Colton, S., et al. 2009. Computational creativity: Coming of age. *AI Magazine* 30(3):11–14.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In De Raedt, L., et al., eds., *ECAI2012 (Proceeedings of the European Conference on Artificial Intelligence)*, 21–26. IOS Press.

Csikszentmihalyi, M. 1988. Society, culture, and person: A systems view of creativity. In Sternberg, R., ed., *The Nature of Creativity: Contemporary Psychological Perspectives*, 325–329. Cambridge: Cambridge University Press.

Goldsmith, K. 2011. *Uncreative Writing: Managing Language in the Digital Age*. Columbia University Press.

Gorman, D. 2001. *Are You Dave Gorman?* Ebury Press.

Johnson, C. G. 2012. The creative computer as romantic hero? computational creativity systems and creative personæ. In Mayer, M. L., et al., eds., *Proceedings of the International Conference on Computational Creativity*, 57–61.

Johnson, C. G. 2013. Artistic and musical application of internet search technologies: Prospects and a case study. *Digital Creativity* 24(4):342–266.

Koza, J. R. 2010. Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines* 11:251–284.

Lewis, G. E. 2000. Too many notes: Complexity and culture in voyager. *Leonardo Music Journal* 10:33–39.

Manovich, L. 2002. *The Language of New Media*. MIT Press.

McGregor, S. 2014. Considering the law as an evaluative mechanism for computational creativity. In al Rifaie, M. M.; Gow, J.; and McGregor, S., eds., *Proceedings of the Symposium on Computational Creativity at the 50th AISB Convention*. Available at http://www.aisb50.org/.

Paine, G. 2002. Interactivity, where to from here? *Organised Sound* 7(3):295–304.

Ritchie, G. D. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Rowe, R. 1993. *Interactive Music Systems: Machine Listening and Composing*. MIT Press.

Sanfilippo, D. 2012. Lies (distance/incidence) 1.0: a human-machine interaction performance. In Polotti, P., et al., eds., *Proceedings of the 19th Colloquium on Music Informatics*, 198–199.

Veale, T. 2011. Creative language retrieval: a robust hybrid of information retrieval and linguistic creativity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 278–287.

Veale, T. 2013. A service-oriented architecture for computational creativity. *Journal of Computing Science and Engineering* 7(3):159–167.

Xenakis, I. 1992. *Formalised Music*. Pendragon Press. Second edition in English translation, first edition in French 1963.

Towards *Dr Inventor*: A Tool for Promoting Scientific Creativity

D.P. O'Donoghue[1], H Saggion[2], F. Dong[3], D. Hurley[1], Y. Abgaz[1], X. Zheng[3], O. Corcho[4], J.J. Zhang[5], J-M Careil[6], B. Mahdian[7], X. Zhao[8]

1 National University of Ireland Maynooth, Ireland.
2 Universitat Pompeu Fabra, Barcelona, Spain.
3 University of Bedfordshire, UK;
4 Universidad Politécnica de Madrid, Spain
5 Bournemouth University, UK
6 Intellixir, Manosque, France
7 ImageMetry, Prague, Czech Republic
8 Ansmart, Wembley, UK

Abstract

We propose an analogy-based model to promote creative scientific reasoning among its users. *Dr Inventor* aims to find *novel* and potentially *useful* creative analogies between academic documents, presenting them to users as potential research questions to be explored and investigated. These novel comparisons will thereby drive its users' creative reasoning. Dr Inventor is aimed at promoting *Big-C Creativity* and the *H-creativity* associated with true scientific creativity.

Introduction

Reasoning with analogical comparisons is highly flexible and powerful, playing a significant role in the creativity of scientific and other disciplines (Koestler, 1964; Boden, 2009). The role played by various analogies in both helping and (implicitly) hindering scientific progress is discussed by Brown (2003). Dunbar and Blanchette (2001) found that analogies were used extensively by working scientists as part of their day-to-day reasoning, playing significant roles in processes from explanation to hypothesis formation.

This paper discusses some initial work on an analogy-based model (called *Dr Inventor*), which will offer computational creativity as a web service to its users who are practising scientists. *Dr Inventor* is focused on helping research scientists by discovering creative analogical comparisons between academic documents and related sources for their consideration. So Dr Inventor will act as a creativity assistant, while its cognitively inspired architecture also offers one possible model of people thinking creatively.

The Web has become a ubiquitous source of publications, source code, data, research websites, wiki and blogs. These form the *Research Objects* (Belhajjame *et al*, 2012), used by Dr Inventor – a tool for the discovery and presentation of creative analogies between research objects. Dr Inventor is targeted on the *Big-C Creativity* (Gardner, 1993) sought by practising scientists. Indeed, the aspirations of Dr Inventor include supporting analogy driven H-creativity (Boden, 1992; 2009).

Analogies compare a *source* to a *target* problem highlight some latent similarity between them. A creative analogy uses a novel source to bring new and creative possibilities to light. Dr Inventor aims to discover novel analogies between academic resources, bringing unnoticed possibilities out of the shadows. Cognitive studies have shown that exposure to even a single analogical comparison can induce significant differences in peoples response to a given problem (Gick and Holyoak, 1980; Thibodeau and Boroditsky, 2011). This paper is focused on identifying *novelty* and *quality* (Boden, 1992) - essential qualities of creativity:

Baydin's (2012) model generated creative analogs for a given target. CrossBee (Juršič *et al,* 2012) looked for bridging concepts between documents from two given domains of interest. Kilaza (O'Donoghue and Keane, 2012) generated creative analogies but it relied on hand-coded data. Dr Inventor will offer a more complete model of creative analogising and blending (Fauconnier and Turner, 1998 Veale, O'Donoghue and Keane, 2000), addressing a broad range of the aspects of creativity.

Dr Inventor Overview

Dr Inventor will include a multi-phase model of analogy encompasing *representation, retrieval, mapping* and *validation.* It may become the first web-based system that supports the exploration of scientific creativity via a computational approach – offering creativity as a web service to its users, i.e. researchers. Dr Inventor is built upon the vision that technologies have a great potential to enhance the broader discipline of scientific creativity. It will build on technologies, such as information extraction, document summarization, semantic web and visual analytics to exploit the great potential in supplementing human ingenuity.

Dr Inventor will become researchers' personal research assistant by reporting to the researchers on a wide variety of relevant concepts through machine-powered search and visualization. It will assess an input research document through comparison with recognized research approaches and suggest new research ideas to the users in an autonomous manner. Dr Inventor will, to a degree, replicate one mode of human creativity to combine diverse information resources and generate new concepts with unexpected features. The new concepts may come from radical transformations inspired by other semantically distant but analogically similar concepts.

Dr Inventor will be based on computational models of analogical reasoning and conceptual blending. Computational models can arguably offer greater creative ability than human reasoners for at least three specific reasons. Firstly, "problem fixation" frequently acts to limit people's ability to think creatively (Lopez *et al*, 2011). Secondly,

people often fail to notice analogies when they are present (Gick and Holyoak, 1980). Thirdly, people often discard useful distant analogies once they have been discovered (Lopez *et al*, 2011). People tend to rate distant analogies as less useful - even if they produce better results. People also suffer from memory limitations, selective thinking, perception limitations, biases, *etc*. A computational model may help to address some of these limitations.

The work of Dr Inventor will synergistically explore techniques for information extraction, document summarization and semantic identification to support the analysis of research objects and the generation of new ontologies for scientific creativity. Interactive visual analytics will be applied to support a user centred creative process. The outcome will be evaluated through appropriately developed evaluation metrics, baselines and benchmarks.

Dr Inventor will focus its evaluation on a specific scientific domain (i.e. computer graphics) exploring *Research Objects (RO)* from various sources. These will include: free research papers on the Web, research websites, Wikipedia, Internet forums, the home pages of many research institutes and groups, as well as individual researchers. In addition, research sites and social networks such as CiteSeer^X, ResearchGate and Google Scholar offer large numbers of freely accessible research papers; research source code is available from GitHub, SourceForge, etc.; and data can also be downloaded for research in computer graphics and image processing, e.g. from Flickr and from benchmarking archives. Secondly, it will use scholarly open-access journals. Finally, it will use online professional digital libraries for top-class research publications. Patents will also be considered within the scope of analysis by Dr Inventor.

The Dr Inventor Model

Research objects will be represented by *skeletons* to allow further processing. A *Research Object Skeleton (ROS)* represents the key concepts and relationships extracted from each RO. Retrieving and representing these ROS is the first task for the Dr Inventor model. The main challenges of the Dr Inventor project are now described in turn.

Information Extraction, Summarization, and RO Skeleton Generation Information Extraction (IE) and Text Summarization (TS) (Poibeau *et al.*, 2013) are two key technologies for transforming document content into concise, manageable semantic representations for use by our creativity model. Dr Inventor's IE aims to find not only general scientific concepts and relations such as: authors, institutions, research objectives, methods, citations, results, conclusions, developments, hypothesis postulation, hypothesis rejection, comparisons, etc. but also domain specific computer graphic concepts/relations such as algorithms, 3D modelling, rendering techniques, *etc*. Initial investigations have identified difficulties in extracting text from papers in PDF format. Issues include: '*ff*' and '*fi*' being represented as single characters, word-flow problems particularly in multi-column documents, representation of mathematical expressions, footnotes and page numbers appearing within the text. PDFX (Constantin *et al.*, 2013) will be used to assist in the text extraction process.

The inventory of entities to be extracted from different data-sources will be modelled in a domain ontology developed for Dr Inventor (see next Section). The most important methods to be used for IE are based on machine learning both supervised and semi-supervised. Indeed, in order for our methods to be applicable to different domains, techniques which are able to learn conceptualizations from raw text and propose new concepts are needed (Saggion, 2013), in this way IE will closely interact with ontology learning so as to expand scientific ontologies with specialized domain information. The GATE (http:///gate.ac.uk) system provides us with the basic infrastructure for developing and integrating basic and advanced IE components. Our current IE system is composed of modules for entity recognition (Ronzano et al. 2014) based on support vector machines (Li et al, 2009) and a rule-based approach for relation extraction based on dependency parsing output (Bohnet, 2010).

Summarization research in Dr Inventor is focusing on adaptation of summarization to scientific data by developing content relevance measures that take into account among other the scientific article rhetorical structure. We are producing an annotated data using an annotation schema based on work by (Liakata et al., 2010). Summaries will be used both as textual surrogates to allow scrutiny for scientist and as content briefers to identify main semantic information in the input. The work is being based on available generic summarization technology being adapted to the scientific domain (Saggion, 2008). Methods to produce these generic summaries are currently based on statistical techniques; however adaptation will be required to target the rich information present in scientific documents - eg Qazvinian *et al.*, (2010). To generate the ROS we need to extract sentence components such as the nouns and verbs, and the structure joining them. For example, from the sentence *"This paper in contrast, proposes a surface-oriented FFD"*, we extract the grammatical subject of the sentence: *paper*, the grammatical object: *FFD* and the relationship holding between them: *propose*. In addition to propositions, information regarding the structure of the article is also available (e.g., the fact that the proposition is extracted from a *purpose* rhetorical zone in the article).

Semantic Technologies & Ontology We will use existing semantic technologies to build up concepts and to identify the relationships between them. Domain ontologies will be built through the learning from a wide variety of research objects, including: documents, datasets, scripts, *etc*. Domain ontologies will also be used and connected to an upper-level ontology network, which will be developed in Dr Inventor as well, reusing existing ontologies covering scientific discourse, document structures, bibliographies and citations (e.g., DoCO, BIBO, EXPO, SPAR etc.) (Belhaj-

jame *et al*, 2012). The extracted information related to authors, co-authors, affiliations, impact factors, h-indices, etc., will be used to facilitate the retrieval and ranking of RO's but it will not be required in the analogy based model. We will also focus on knowledge extraction from user-defined tags associated to research objects and their aggregated objects, following on current work in ontology learning from folksonomies. In addition, extending existing work on social recommendation of research objects, we will be able to discover implicit relationships between different pieces of work that were originally not considered by the author in a basic literature exploration activity that can increment creativity in research. Such an ontology network will be designed to allow the representation of scientific discourse for scientific creativity.

With respect to ontology matching, we want to make use of existing techniques (Shvaiko and Euzenat, 2013) in the context of applying structured similarity evaluations between the aggregations of objects that are represented by research objects. In this context, knowledge extracted from documents and other artefacts should be seen as a skeleton set of information that summarizes key ideas, which allows researchers to explore the content of existing RO's in the process of their evaluation and of the generation of scientific innovation. This will contribute to the similarity measure for comparing research object skeletons for the creativity process. Finally, ontologies will also be used to provide personalized recommendations of scientific RO's, using different sets of recommendation techniques.

Retrieval Model Retrieval will combine several techniques to identify homomorphic skeletons. A vector space model will enable quick, inexpensive comparison between skeletons, using numeric qualities representing the topology of each skeleton. This will also account for the inferences we expect to find in creative source domains.

Analogy/Blending Model Dr Inventor's comparison model will identify and extend detailed similarities between ROS. It will typically search for a source to reinterpret a given target problem, but can also select its own targets. Dr Inventor's final structure may be best seen as a conceptual blending (Fauconnier and Turner, 1998) model. It accepts as input two ROS, a generic space represents ontological and other commonalities while the output space represents the new creative concept (blend). (Space doesn't permit proper treatment of the similarities and differences between analogy and conceptual blending).

Dr Inventor presents many challenges to similarity based discovery, such as; identifying a compelling source ROS, balancing structural and semantic factors in the mapping phase and performing quality assurance on the resulting inferences. Choosing the correct interpretation(s) of each domain to find an appropriate mapping will also be crucial.

The analogy-based model envisages the re-description of any given target using a pre-stored collection of sources with which to re-interpret that problem. This requires a rich memory of background knowledge to seek creative interpretations of the targeted problem through an extensive analogical comparison to a wide range of objects. In this context, Dr Inventor aims at exploring the potential of web resource to promote scientific creativity. From the previous example sentence in the IE section, we have the graph `[paper]→(propose)→[ffd]` where [] is a concept node and () denotes a relation connecting concept nodes.

Visual Analytics In Dr Inventor, visual analytics will serve to visualize the analogical reasoning and conceptual blending processes. Graphs visualization is a natural choice for the visualization of the ROS, which can also be supported by other means of visualizations. This could involve a large number of skeletons with a considerable level of uncertainty originated from similarity measures between the ROSs. Also, to allow effective handling of large scale visualization, we will investigate aggregation techniques such as binning, abstraction, hierarchical clustering to create effective aggregation of data at different levels of details.

User interaction with a creative system is an interesting research issue. The interaction techniques are categorized as select, explore, reconfigure, encode, abstract/elaborate, filter, and connect. An important task of user interaction is to help user navigation of the data. To this end, the interaction will follow the recommendation of *"overviews first and details-on-demands"* by working together with the data aggregation. Also, techniques that support zoom in within local areas, focus+context and coordinate views will help users to interactively explore comparisons without losing the perception towards the overall data structure.

Web-Based Creativity Service Dr Inventor will present a web-based system for exploring scientific creativity. It will offer a front-end web interface and a back-end mechanism addressing data transfer, access and federation, resource management, *etc*. The backend crawler will constantly gather research objects from the web extending the ROS repository. Information extraction and subsequent activities will be applied, as previously discussed.

At the front end, a web-based interface will be built to provide interface to allow interactive browse, search and visualization of the analogies of ROSs from the repository that contains analogically matched skeletons to inspire user creativity; to provide interface to assess an input RO; to provide interface for a creativity inspiration engine that allows scientific creativity promotion in highly interactive ways. The system is expected to be linked to a social network service (e.g. LinkedIn, Facebook or Twitter) to enhance the interaction and to explore common interest between the researchers. Finally, APIs will be developed to support further development.

Evaluation Among the remaining significant challenges will be evaluation of Dr Inventor, assessing its impact on the creativity of its user groups. This will rely heavily on access to a group of domain (computer graphics) experts for assessment and evaluation. Just as important to Dr In-

ventor is the development of a set of benchmarks and metrics for evaluating progress of this project.

Conclusion

Models of analogical reasoning are presenting new horizons for intelligently processing information, unearthing creative possibilities in new and surprising ways. Using analogy-based models upon academic resources is a broad and open-ended challenge, requiring advances in areas like document analysis, representation, ontology, analogy & blending, visualization *etc*. Dr Inventor aspires to Big-C Creativity (Gardner, 1993) hoping to support the transformational creativity (Boden, 1992) associated with significant scientific progress. Boden (1998) identifies two major "bottlenecks" for transformational creativity. Firstly, the domain expertise required for mapping the conceptual spaces to be transformed and secondly, valuing the results produced by a transformationally creative system. We believe that both challenges will be addressed by the combined efforts of the different activities in Dr Inventor, leading to a powerful tool that will invigorate the research communities opening up new and exciting possibilities.

A number of high-level issues arise related to Dr Inventor. Firstly, is documented information sufficiently complete to allow fruitful comparisons to be drawn between research papers, collections of papers or other sources? Can Dr Inventor adequately identify creative analogies from such sources? Will users be sufficiently receptive to accept creative inspiration from Dr Inventor? How can we maximize the impact from each component of Dr Inventor to produce comparisons with the greatest effect on its users? These and many other challenges await.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement 611383.

References

Belhajjame, K.; Corcho, O.; Garijo, D.; Zhao, J.; Missier, P.; Newman, D.R.; Palma, R.; Bechhofer, S.; Garcia-Cuesta, E.; Gómez-Pérez, J.M.; Klyne, G.; Page, K.; *et al* 2012 *Goble CA: Workflow-Centric Research Objects*. Proc. ESWC2012 Workshop on the Future of Scholarly Communication in the Semantic Web (SePublica2012), Greece.

Boden, M.A. 1992. *The Creative Mind*. London: Abacus.

Boden, M.A. 1998. *Creativity and artificial intelligence*, Artificial Intelligence, 103, 347-356.

Boden, M.A. 2009. *Computer Models of Creativity*, AI Magazine, 23-34.

Bohnet, B. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. Proc. COLING, pp 89-97.

Brown, T.L. 2003. *Making Truth: Metaphor in Science*, University of Illinois Press.

Constantin, A.; Pettifer, S.; Voronkov, A. 2013. *PDFX: Fully-automated PDF-to-XML Conversion of Scientific Literature*. ACM Symp. Document Engineering, 177-180.

Fauconnier, G. and Turner, M. *Conceptual Integration Networks*, Cognitive Science 22(2), 133–187, 1998.

Gardner, H. 1993. *Creating Minds*, Basic Books, NY.

Gick, L. M. and Holyoak, J. M., 1980. *Analogical problem solving,* Cognitive Psychology, 12, pp 306-355.

Juršič, M.; Cestnik, B.; Urbančič, T.; and Lavrač, N. 2012. *Finding Bridging Concepts with CrossBee*, International Conference on Computational Creativity, Ireland, 33-40.

Koestler, A. 1964. *The Act of Creation*, Penguin, NY.

Lopez, R.; Lindsey, J.S.; and Smith, S.M. 2011. *Characterizing the effect of domain distance in design-by-analogy*, ASME IDETC-Design Theory and Methodology Conf.

Li, Y., Bontcheva, K. and Cunningham, H. 2009: Adapting SVM for Data Sparseness and Imbalance: A Case Study on Information Extraction. Natural Language Engineering, Vol. 15, pp. 241-271, Cambridge University Press (2009)

Liakata M.; Teufel S.; Siddharthan A.; Batchelor C. 2010. Corpora for conceptualisation and zoning of scientific papers. 7th Conf. Intl. Language Resources and Evaluation.

O'Donoghue, D.P. and Keane, M.T. 2012. A Creative Analogy Machine: Results and Challenges, *4th International Conference on Computational Creativity,* Ireland, 17-24.

Poibeau, T.; Saggion, H.; Piskorski, J.; and Yangarber, R. 2013. *Multi-source, Multilingual Information Extraction and Summarization++,* Theory and Applications of Natural Language Processing, Springer.

Thibodeau, P.H. and Lera Boroditsky L. 2011. *Metaphors We Think With*, PlosOne, 6(2) 1-11.

Qazvinian, V.; Radev, D.R.; and Özgür, A. 2010. *Citation Summarization Through Keyphrase Extraction,* COLING 2010, 895-903.

Ronzano, F.; Casamayor, G.; and Saggion, H. 2014 Semantify CEUR-WS Proc.: towards the automatic generation of highly descriptive scholarly publishing Linked Datasets. Proc. ESWC-14 Challenge on Semantic Publishing.

Saggion, H. 2008. SUMMA: A Robust and Adaptable Summarization Tool. Traitement Automatique des Langues 49(2). pp103-125.

Saggion, H. 2013. Unsupervised Learning Summarization Templates from Concise Summaries. Proc. North American Chapter of Assoc. Computational Linguistics, 270-279.

Veale, T.; O'Donoghue, D.; Keane, M. 2000. *Computation and Blending*, Cognitive Linguistics, 11, 3/4, 253-281.

Shvaiko, P.; Euzenat, J. 2013 Ontology Matching: State of the Art and Future Challenges. IEEE Transactions on Knowledge and Data Engineering 25(1), 158–176.

Combining Representational Domains for Computational Creativity

Agnese Augello, Ignazio Infantino, Giovanni Pilato, Riccardo Rizzo, Filippo Vella
Istituto di Calcolo e Reti ad Alte Prestazioni
Consiglio Nazionale delle Ricerche
sede di Palermo, ed. 11
e-mail: name.surname@cnr.it

Abstract

The paper describes a combinatorial creativity module embedded in a cognitive architecture. The proposed module is based on the focus of attention model proposed by and is implemented using Self Organising Map (SOM) neural networks.

Introduction

Creativity is mainly perceived as a high level cognitive characteristic, which should always be referring to a conceptual space, whether it is conceived to explore or to transform such space (Boden 2009). One of the components of creativity is an associative memory capable of restoring an incomplete sensory input stimulus by adjusting focus of attention.

A cognitive model for creativity based on the ability of adjusting focus of attention has been proposed in (Gabora 2002). According to this model a variable focus of attention, while pointing the basic idea, also collects other concepts that are parts of the stream of thought. The focus of attention can be considered as a basic idea, a *framework* that drives the creative process which is connected to the analytical mode of thought.

At the same time another basic component of the cognitive model proposed by Gabora is the associative memory. By means of associations between different concepts and completion mechanisms, new and surprising results can emerge (Bogart and Pasquier 2013). This kind of creative process can be bound to the process that Boden calls combinatorial creativity which is related to making unusual combinations, consciously or unconsciously generated, of familiar ideas (Boden 2009).

The Arcimboldo painting can be a good example for clarifying what we intend.The painting of a human figure presumes a very precise *framework* that is constituted by figure details, as nose, eyes, lips, rules and relative positioning and all the other details that made a human figure. The attention focus is what we use to "navigate" on the *framework*, what is pointing at the details of the figure, that can be substituted with elements belonging to another domain (as in the painting in fig. 1) exploiting the associative memory. We believe that, during the creative process of imagining the painting, the attention is relaxed and other images, searched in another domain, come in mind and take the place of the original parts of the human figure.

We consider completion operation in a very large meaning. The basic point of the combinatorial creativity is to mix together parts coming from different sources. In this sense completion is a way to enrich a *framework* with new items in order to obtain new combinations.

In our opinion it is possible to have robust fusion algorithms and completion through the combination of various models of neural networks: an example of such an approach is described in (Thagard and Stewart 2011) that allows emphasising associations useful to generate creative ideas by simple vector convolution. The importance of associative mechanisms is also underlined by neurobiological models of creativity, many of which are based on the simultaneous activation and communication between brain regions that are generally not strongly connected (Heilman, Nadeau, and Beversdorf 2003).

In this paper we illustrate an approach aimed at supporting the execution of an artificial digital painter (Augello et al. 2013b) (Augello et al. 2013a). The approach is exploited by the Long Term Memory (LTM) module of the cognitive architecture presented in (Augello et al. 2013b) and reported in fig. 2. The proposed approach is based on a multilayer mechanism that implements an associative memory based on Self Organizing Maps (SOMs) (Kohonen, Schroeder, and Huang 2001) and it is capable to properly mix elements belonging to different domains.

Figure 1: A detail from Spring (1563), an Arcimboldo painting (Image from Wikipedia).

Architecture

In (Augello et al. 2013b) we defined the mechanisms to support creativity in a cognitive framework. In this work we use the same architecture (see fig. 2) but we adopt a new version

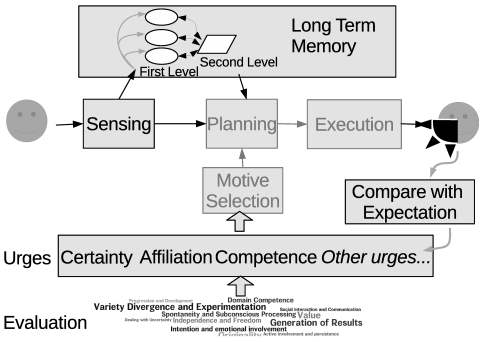


Figure 2: The general cognitive framework used for the proposed system. Light grey blocks are neglected in this implementation.

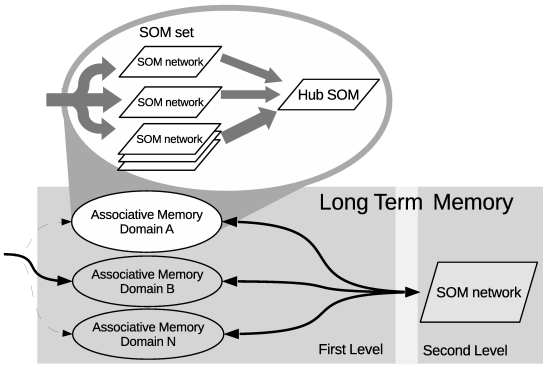


Figure 3: The overall schema of the proposed architecture for the Long Term Memory module (LTM).

of LTM (Long Term Memory) that implements an associative mechanism described in details below.

As said before, one of the basic components is an associative memory capable of restoring an incomplete sensory input stimulus.

Completion is guided by context: when we interpret fuzzy or confused handwritten characters, we use associations with memorised handwritten characters, then we complete or rebuild the input, so that the most common association are made using objects of the same context. Objects coming from the same domain are probably represented by the same features and share the same concept space that was described in Gärdenfors (Gärdenfors 2004). Associations can also involve objects from different contexts in a more "creative" way. In this case the original context is discarded and objects come from different domains.

According to these considerations we have built a multilayer mechanism that allows to connect memory locations related to a single domain. We have also built another layer that is used to connect memory locations with a more general association mechanism that allows to make associations that go beyond the domain. This second upper layer will be used when the original domain is discarded, for example when we want to find other solutions or we want to mix different domains. The kind of associations made at the second level will be the associations made when the focus of attention is relaxed and associative connections can be made even outside of a specific domain. The structure we propose is represented in fig. 3.

Input from sensors are sent to the proper domain at the first level and they are memorised or completed when necessary. The second level contains the associations among different domains that will be further explained in the following paragraphs.

The associative memory module that we propose is inspired by the work in (Morse et al. 2010) and is implemented using a Self Organising Map (SOM) neural networks (Kohonen, Schroeder, and Huang 2001).

Self Organising Maps are neural networks constituted by a single layer of neural units usually organised in a 2D grid. After a successful training phase each neural unit ideally approximates the centroid of an input pattern cluster and the neighbour units represent similar values. This way each neural unit corresponds to a sort of average pattern for a cluster of inputs.

The architecture proposed in (Morse et al. 2010), is made by multiple SOM, each one receiving inputs from a different sensory modality. In our architecture the SOM array, in the upper part of fig 3, receives inputs from different features extracted from the same sensory input, so that a SOM of the set can have colour features from image, another image boundaries, another one texture information and so on. The values of the SOMs are collected by the *hub–SOM* that synthetically represents the object gathering the representations of the different SOMs. This process is sketched in fig. 4, where different features are substituted by different parts of the image.

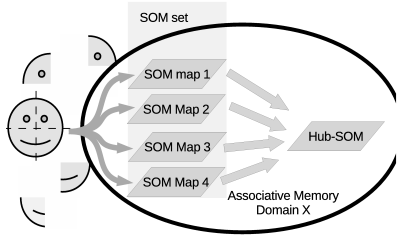


Figure 4: The associative domain memory training.

While the SOM set and the hub–SOM constitute the associative module for a domain there is also another SOM, named second level SOM, where the association among different domains takes place.

The information from the domain modules, in this second level, are represented using more general features. For example if a domain is used to memorise images of trees and one of the SOM in the array in fig.3 memorises the shape of the leafs, the second level SOM can use the dimension of the bounding box[1] as a feature. When we want to mix together objects from other domains we can consider objects that have the same bounding box. The substitution will

[1]the bounding box is the rectangle surrounding an image detail

be driven by the second level SOM whose aim is to faithfully reflect the general structure of the image. A substitution according the bounding box dimensions is a simple criterion but a more general set of features that could also be employed. This second level SOM will implement the spreading of the attention focus because it will mix objects from different domains and group them just considering very rough characteristics.

The next two subsections will explain how we can implement the effects of a variable focus of attention: with a narrow focus we can obtain simple completion inside the same domain while with the spread of the focus we can recover objects from different domains.

Completion in the same domain

Completion in the same domain is the simplest form of completion. For example, let us assume that a domain is trained to memorise simple images, and imagine that, inside this domain there are SOMs that memorise very specific parts of the image: we can think that each SOM memorises a quadrant of the image or, when representing faces, segments of human faces. In this case the basic components would be eyes, lips, noses and so on, memorised along with their positions, in different SOMs. The hub–SOM takes into account all the positions of the components. This is sketched in fig. 4.

If a part of an image is missing, only some of the SOMs can recall the corresponding memory locations and help to reconstruct the memorised image: one, or more, SOMs will not answer because they do not have any input. The hub–SOM accomplishes the task to recall the necessary memory locations from the SOMs that do not have any input, in order to put together all the pieces of the image.

This procedure is depicted in fig.5: the missing piece of image causes a failure of recalling in SOM Map 4, so that the hub–SOM, containing the reference of the whole, outputs the address of the location of the SOM Map 4 and recall the missing piece.

Completion in a different domain

When completion is obtained using "parts" or memories that are outside the domain of the original image, or input, we are making an association that is not causal. This can happen when the recalled part is used to obtain memory contents from other, different domains. In this case the associations are the ones memorised in the second layer SOM, i.e. an association that corresponds to features of different kind.

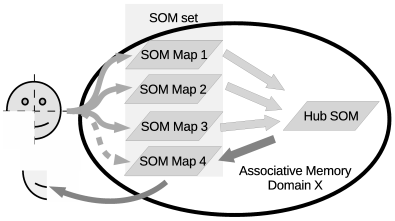


Figure 5: The completion procedure in the domain

In fig. 6 the whole process is sketched : the missing part is recalled as said before; however, in this case, it is not sent to the output but it is sent to the second level SOM where it is used for recalling objects from different domains.

The recovered information is used as a reference in order to obtain the missing part that is sent to the second level SOM. This signal excites a unit of the second level SOM and its output is sent back to all the associative memory of the other domains. Each domain answers with a list of the excited units that point out to a set of signal corresponding to the memorised objects. As indicated in fig 6 all these objects are proposed as substitution of the missing part. At this point the completion proposed from the original domain is again used as a reference: all the proposed substitution are compared to the original completion and the most similar one is chosen as a substitute. This mechanism is implemented in the box "Implementation with Expectation" in fig 6.

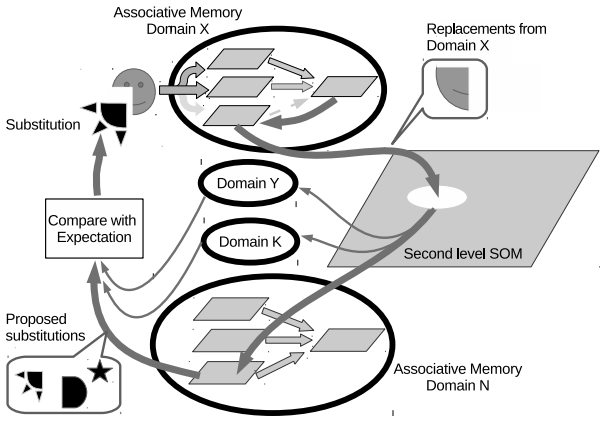


Figure 6: The completion procedure outside the domain.

Some Experimental Results and Conclusive Remarks

The experiments were mainly performed to evaluate the effectiveness mechanism of the replacement of some parts of the images by the associative memory previously described. We have chosen a *face domain* that allows an immediate recognition and a *leaves and flowers domain* in order to resemble the effect of the Arcimboldo style. A sample of the images in our dataset is given in fig.7. The system was trained using 113 grey-scale images of faces and 100 images of leaves and flowers. Each image is 100×100 pixel in order to maintain a manageable size of the neural architecture.

In order to reproduce the completion mechanism and to partially simulate the mechanism of focus of attention, each quarter of the image has been memorised in a different map of the array of SOMs (see fig.4). We have tried a quad tree decomposition and the learning process described above. An example is reported in fig.9.

Each SOM in the array has a size of 20×20 units and is trained with segments in the same position of the image using the fast training procedure described in (Rizzo 2013),

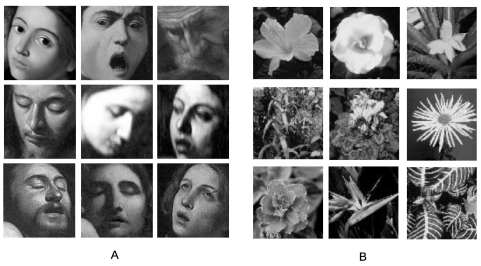


Figure 7: An example of the images in the faces domain (A) and leafs and flowers (B) domain.

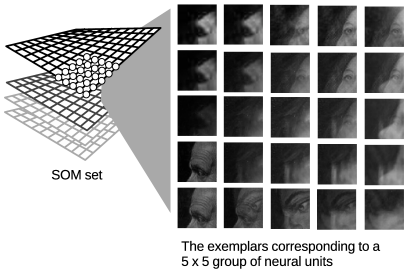


Figure 8: The map in the array of SOMs after the training are used as memory units.

and the result is shown in fig. 8. At the end of the process it is possible to train the hub–SOM submitting the images of the training set to the SOM array; for each SOM we will get the two digits coordinates on the neural units array, of the most similar exemplar (often called best matching unit or b.m.u.). These coordinates are submitted to the hub–SOM that learns this 8 digits image coding and, after training, will be able to rebuild the correct coding for each image.

This kind of representation is too precise to be used also at higher level, were we want to "mix together different things". At higher levels we want a representation that captures just some of the characteristics of the images, for example colour masses, boundaries and shapes, and so on. For this reason we used the Haar and Gabor features, which contain less information.

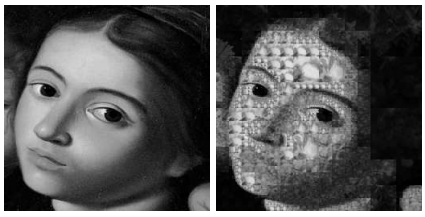


Figure 9: Final artwork obtained by our approach

Conclusion

The preliminary experimental results show that the proposed associative memory module is promising for the implementation of a sort of combinatorial creativity mechanism. Future works will regard the modelling of artist's behaviour and motivation, the choice of domains during the completion process, and the evaluation of both creative process and produced artworks, according to the literature works (Pease and Colton 2011) (Colton and Wiggins 2012) (Jordanous 2012).

Acknowledgment

This work has been partially supported by the PON01_01687 - SINTESYS Research Project.

References

Augello, A.; Infantino, I.; Pilato, G.; Rizzo, R.; and Vella, F. 2013a. Binding representational spaces of colors and emotions for creativity. *Biologically Inspired Cognitive Architectures* 5:64–71.

Augello, A.; Infantino, I.; Pilato, G.; Rizzo, R.; and Vella, F. 2013b. Introducing a creative process on a cognitive architecture. *Biologically Inspired Cognitive Architectures* 6:131–139.

Boden, M. 2009. Computer Models of Creativity. *AI Magazine* 23–34.

Bogart, B., and Pasquier, P. 2013. Context Machines: A Series of Situated and Self-Organising Artworks. *Proceedings of the fourth conference on Creativity & cognition - C&C '02* 46(2):114–122.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *ECAI'12*, 21–26.

Gabora, L. 2002. Cognitive mechanisms underlying the creative process. *Proceedings of the fourth conference on Creativity & cognition - C&C '02* 126–133.

Gärdenfors, P. 2004. *Conceptual spaces: The geometry of thought*. The MIT Press.

Heilman, K. M.; Nadeau, S. E.; and Beversdorf, D. O. 2003. Creative innovation: possible brain mechanisms. *Neurocase* 9(5):369–379.

Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.

Kohonen, T.; Schroeder, M. R.; and Huang, T. S., eds. 2001. *Self-Organizing Maps*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 3rd edition.

Morse, A. F.; de Greeff, J.; Belpeame, T.; and Cangelosi, A. 2010. Epigenetic Robotics Architecture (ERA). *IEEE Trans. on Autonomous Mental Development* 2(4):325–339.

Pease, A., and Colton, S. 2011. Computational creativity theory: Inspirations behind the face and the idea models. In *ICCC 2011*.

Rizzo, R. 2013. A new training method for large self organizing maps. *Neural Processing Letters* 37(3):263–275.

Thagard, P., and Stewart, T. C. 2011. The aha! experience: Creativity through emergent binding in neural networks. *Cognitive science* 35(1):1–33.

Exploring Conceptual Space in Language Games Using Hedonic Functions

Anhong Zhang
Design Lab
University of Sydney
Sydney, NSW 2010 Australia
azha3482@uni.sydney.edu.au

Rob Saunders
Design Lab
University of Sydney
Sydney, NSW 2010 Australia
rob.saunders@sydney.edu.au

Abstract

The ambiguity of natural language can be an important source of creative concepts. In compositional languages, a many-to-many network of associations exists linking concepts by the polysemy and synonymy of utterances. This network allows utterances to represent the combination of concepts, forming new and potentially interesting compound meanings. At the same time, new experiences of external and internal contexts provide abundant materials for the evolution of language. This paper focuses on exploring the role of compositional language for social creativity through the simulation of language games running on multi-agent systems using a hedonic function to evaluate the interest of utterances as design requirements and the resulting design works.

Introduction

A single word may be associated with multiple meanings while one meaning can be represented by multiple words. Such ambiguity of polysemy and synonymy can be a source of creative inspiration, allowing the exploration of conceptual spaces by traversing the many-to-many mappings between words and meanings. Many-to-many mappings between utterances not only construct connections between seemingly unrelated concepts, but also provide more opportunities to recombine sub-utterances to new utterances representing novel meanings.

The function of the ambiguity of language for social creativity can be explored through the use of language games combined with multi-agent simulation. In the *guessing game* (Steels, 1995), a speaker-agent describes an object using an *utterance* to a listener-agent who attempts to identify the *topic* of the utterance based on its experience of previous utterances and the current context. By repeating the guessing game for many generations, a simple language, grounded in use, may evolve (Steels, 1995).

In the *generation game* (Saunders and Grace, 2008), agents that were previously speakers or listeners in a guessing game, explore a conceptual space using communication between client-agent and designer-agents. A requirement is expressed as an utterance by a client-agent and may be related with various meanings by multiple designer-agents that have different experiences of similar utterances. The creativity of communication primarily depends on client-agent generating an “interesting” requirement and selecting “interesting” design works produced by designer-agents in response. The evaluation of interest can be modelled using a hedonic function, e.g., the Wundt Curve (see Figure 1), where similar but different perceptual experiences are preferred (Saunders, 2009).

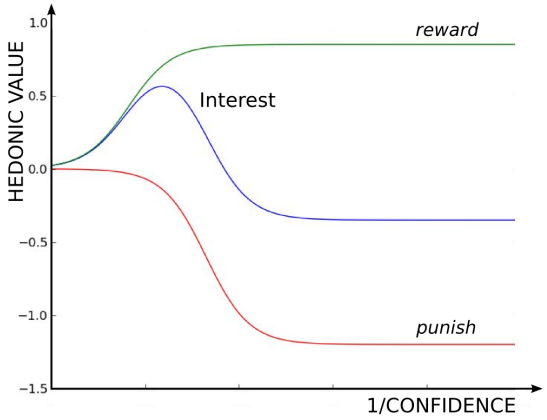


Figure 1. The Wundt Curve, a hedonic function for evaluating interest based on agents' confidence

Methods

The language games used in the simulations described in this paper produce utterances as a result of a compositional language. Compositional languages, as opposed to holistic languages, permit utterances to be composed using multiple words. Composition can be utilized to generate new utterances denoting valuable concepts. For example, given the previous utterances RED SQUARE, RED TRIANGLE and BLUE TRIANGLE, new utterances such as BLUE SQUARE may

be generated by recombining the evolved sub-utterances BLUE and SQUARE.

The agents in the simulation use Adaptive Resonance Theory (ART) networks to categorize utterances and concepts. ART networks are both stable and dynamic, they can not only retain existing categories but also add new categories for unfamiliar inputs which exceed the threshold of recognition of ART system (Saunders, 2002).

Experiment

The experiment focuses on exploring the combination of existing utterances generating new utterances representing interesting meanings through the communication between agents who play the roles of speaker and listener in guessing game as well as client and designer in generation game.

Experiment Settings

1. Initial settings

The first set of experiments were initialized with 50 samples randomly selected from 121 objects, which were generated by combining 11 colors and 11 shapes. Each of the shapes is represented by a list, e.g., [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]. The population of agents is 6. The language game uses one combination rule combining two features including color and shape; and each feature is represented by one letter as its name. So the length of utterance is limited to 2. For example, {color:0.2, shape:0.3}'s utterance may be “ha”.

2. Guessing game settings

In the guessing game, 8 topics are selected randomly from the samples available for each exchange between speaker and listener, selected among 6 agents randomly. When the success rate (see Equation (1)) is above 60%, the guessing game is finished and generation game is started.

$$rate_{success} = times_{success} / (times_{success} + times_{failure}) \quad \textbf{(1)}$$

3. Generation game settings

In the generation game, four types of procedures with or without evaluation of the interest of requirements (utterances) or works using The Wundt Curve (Figure 1) are implemented. Each design cycle is repeated 1000 times. Every time, the last agent always plays the role of client while others play the role of designers.

Experiment Procedures

Procedure 1. Guessing game

The following guessing game is implemented repeatedly till success rate reaches 60%.

1. The speaker selects one topic randomly from randomly generated context.
2. The speaker generates an utterance representing the selected topic and tells listener.
3. The listener guesses the topic by exploring its existing associations between utterances and the ART categories. If an appropriate association cannot be found, a new association between the utterance and the topic in current context is generated. Then listener tells the speaker its guess.
4. If successful, both speaker and listener increase the weight of their association between the topic’s ART category and the utterance and increase the frequency of each related instance or generate a new association connecting the selected topic with the utterance. Otherwise, if guessing failed, the listener decreases the weight of the related association and generates a new association between the topic’s ART category and the utterance, then increases the weight of the newly generated association and generates a new association connecting the correct topic and the utterance.

After completing the guessing game, the agents (Group A) are cloned three times to get three new groups of agents (Group B, Group C and Group D) to implement different procedures for the generation game.

Procedure 2. Generation game without evaluation of interest

The generation game is implemented for 1000 generations by Group A without evaluating the interest of requirements and works.

1. Each designer-agent generates a set of design works (topic) by searching existing associations or generating a new association connecting related an ART category with client-agent’s requirement (utterance).
2. A client-agent generates an utterance by combining two randomly selected names of each feature’s ART category (prototype) without evaluation.
3. The client-agent selects the most similar design works compared with its requirement-associated topic. But if the most similar works did not belong to the same ART category of client-agent’s associated topic, game fails; and all designer-agents decrease the weights of their own selected associations. Otherwise, client-agent finds its own relevant association or generates a new association connecting its own ART category of the selected design works and the utterance, then increases the weight of the association and increases the frequency of related instance or generates new instance connecting the works and the utterance. At the same time successful designer-agents increase the weight of related rule and increase the frequency of related instance or generate a new instance while other designer-agents decrease the weights of related associations.

Procedure 3. Generation game with evaluation of works interest

The generation game is implemented for 1000 generations by Group B. The procedure is the same as Procedure 2 ex-

cept that the client-agent selects design works using the Wundt Curve. In the process of selecting design works, the distances between each design works' features and client-agent's original topic's features are first measured, then their hedonic value is evaluated. The design works with the highest interest are selected by client-agent. If all interests were negative, the generation game fails.

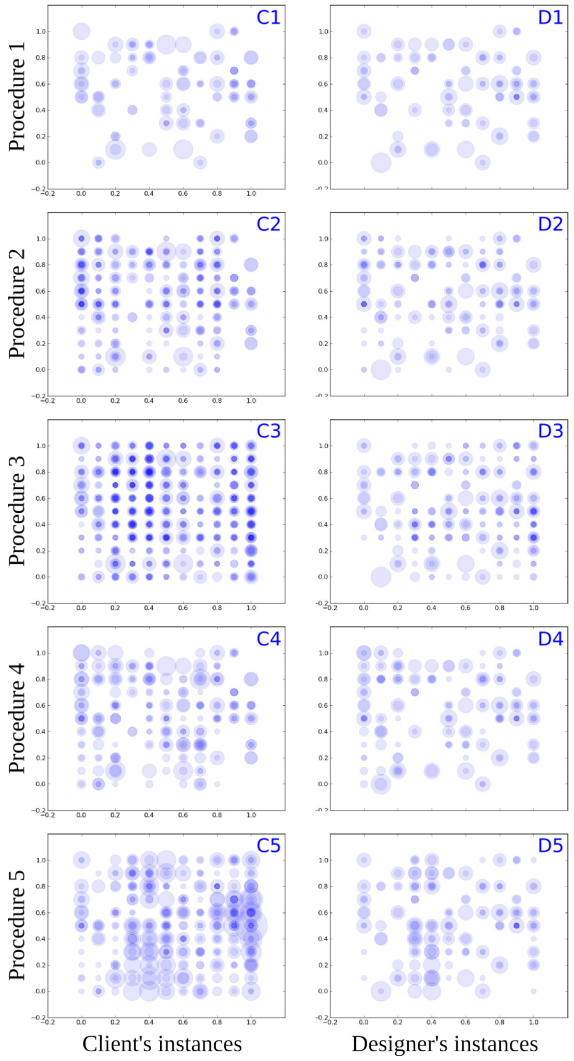


Figure 2. An example of the distributions of agents' instances

Procedure 4. Generation game with evaluation of requirements interest

The generation game is implemented for 1000 generations by Group C. Each time, the procedure is the same as Procedure 2 except that client-agent generates several requirements (utterances) and select the most interesting one. Firstly, the weight of each single utterance in every requirement is calculated by summing the frequencies of the utterance used in all instances. Then the interest values of these requirements are calculated by summing the interests of their own utterances. Finally, the requirement with the highest interest is selected.

Procedure 5. Generation game with evaluation of both requirements interest and works interest

The generation game is implemented for 1000 generations by Group D. In each generation game, the procedure is the same as Procedure 2 except for the generation of interesting requirements and the selection of interesting design works by client-agent. The process of generating interesting utterance is the same as Procedure 4. The process of selecting interesting design works is the same as Procedure 3.

Results

In Figure 2, the radius of each circle represents the frequency of an instance used by an agent. If a topic is associated with more than one utterance, several circles will be drawn at the same place resulting in a darker color.

The results of the experiments show that agents explored a greater number of new topics and generated more instances (the associations between topics and utterances) when the client-agent used the Wundt Curve only for selecting interesting design works, see Figure 2(C3). But the frequency differences between the instances are not distinctive compared with when client-agent utilized the Wundt Curve not only for selecting interesting works, but also for generating interesting requirements, see Figure 2(C5). This suggests that the client-agent preferred using a small set of interesting utterances frequently. Hence, the frequency-distribution of instances is nonuniform. This is similar as the signature of life with uneven frequency distribution comparing Figure 2(C5) with Figure 2(C3).

The number of designer-agent's instances are less than that of client-agent's instances, see Figure 2(C2–D5) except that generated in guessing game, see Figure 2(C1,D1) because only one designer-agent's works could be accepted by the client-agent in the most successful interactions while other designer-agents had no opportunities of updating their instances, but the client-agent can update its instances almost every successful time in generation game.

The average number of instances generated by client-agent and that by designer-agents in a generation game are shown in Figure 3. When only using the Wundt Curve to assess the interest of design works, the number of instances increased sharply especially for client-agent. However, when using the Wundt Curve to evaluate not only the interest of design works, but also that of utterances, the number of instances decreases even below that of instances generated without evaluation of interest except the average number of designer's instances generated in Procedure 5, which is slightly higher than that in Procedure 2 but still lower than that in Procedure 3.

The average max degree of the graph networks of instances generated by client-agent and designer-agents respectively are illustrated in Figure 4. As can be seen, the highest average max degree belongs to the instances generated using

the Wundt Curve selecting both interesting requirements and interesting works. The average max degree related with the evaluation of only requirements are higher than that of only woks. Therefore, the evaluation of the interest of utterance may be more important than that of works.

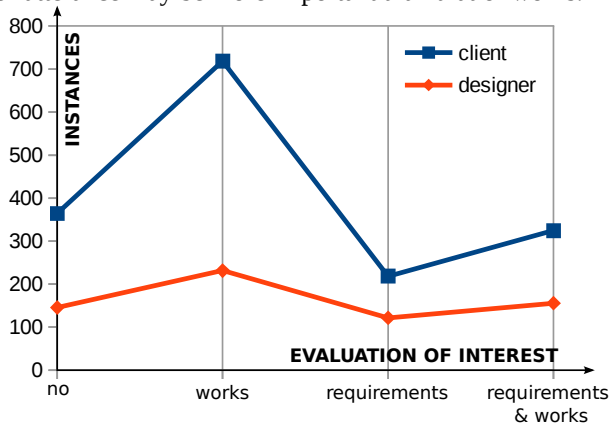


Figure 3. The average number of agents' instances generated with or without evaluation of interest in generation games

Discussion

Based on the results of the simulations, client requirements may be more important than designer's works because the final pattern of the distribution of utterances and design works are primarily determined by the client-agent rather than the designer-agents. "Interesting" requirements narrow the combination area of utterances initially generated via crossover of two randomly selected utterances, resulting into the selection of interesting artifacts.

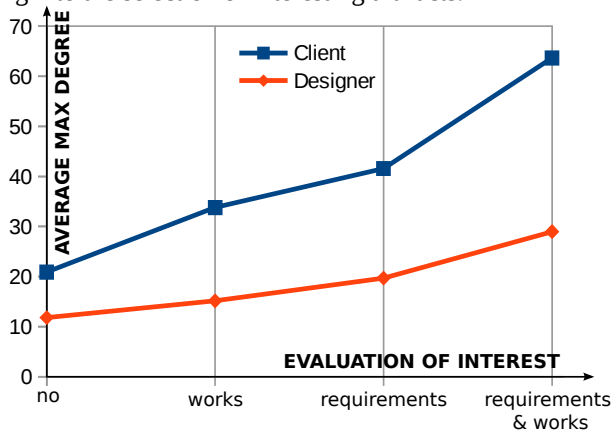


Figure 4. The average max degree of the graph networks of agents' instances generated with or without evaluation of interest in generation games

According to the illustrations of both Figure 3 and Figure 4, "less is more" is realized as less instances and more connections. In other words, many meanings may be associated with one utterance while the total number of utterances can be relatively small when using a hedonic function to select randomly combined utterances. Consequently, more connections may lead to discovering more new concepts.

Therefore, the procedures of language games described in this paper could be adopted in brainstorming by both clients and designers evolving original requirements and novel concepts. The combination of guessing game and generation game can also be utilized in artificial collaborative system to handle the evolution of compositional language for creative design.

Conclusions

The results of the simulations suggest the following conclusions:

1. Using a hedonic function to evaluate the interest of utterances affects the direction for exploring conceptual space.
2. The ambiguity of language especially caused by polysemy may play an important role in creative communication by using compositional language.
3. Client-demand driven design may be more important than content driven design in social creative systems.

Future work

Graph theory has been used in this paper for evaluating the degree of connections between utterances and meanings. Other graph-theoretic functions (Hagberg, Swart, and S Chult 2008) such as density, diameter related with the evaluation of social creativity will be explored.

Language games based on Fuzzy sets have been implemented in our most recent experiments. So, the simulations using Fuzzy sets to represent vague and ambiguous concepts will be studied in near future.

References

Hagberg, A., Swart, P., & S Chult, D. (2008) Exploring network structure, dynamics, and function using NetworkX (No. LA-UR-08-05495; LA-UR-08-5495). Los Alamos National Laboratory (LANL).

Saunders, R. (2002) Curious Design Agents and Artificial Creativity—A Synthetic Approach to the Study of Creative Behavior. University of Sydney, NSW.

Saunders, R., and Grace, K. (2008) Towards a computational model of creative cultures, *AAAI Spring Symposium on Creative Intelligent Systems,* 26–28 March 2008, Stanford University.

Saunders, R. (2009) Artificial Creative Systems and the Evolution of Language. *In Proceedings of the Second International Conference on Computational Creativity.*

Steels, L. (1995) *A self-organizing spatial vocabulary.* Artificial Life 2, (3):319–332.

The apprentice framework: planning and assessing creativity

Santiago Negrete-Yankelevich and Nora Morales-Zaragoza
División de Ciencias de la Comunicación y Diseño
Universidad Autónoma Metropolitana-Cuajimalpa
Vasco de Quiroga 4871, Santa Fe Cuajimalpa de Morelos 05348, D.F., México
{snegrete, nmorales}@correo.cua.uam.mx

Abstract

In this paper we introduce and discuss the apprentice framework, which we speculate can be used to plan and evaluate computational creativity projects. The framework defines a sequence of phases a system must follow in order to reach a level of creativity acceptable to a set of human judges. It also establishes four aspects of a creative piece susceptible of creative work. We mention some examples from different artistic disciplines. Our work focuses on establishing an environment as well as a team of people and machines to foster, study and monitor the emergence of creativity.

On Human and Machine Creativity

Assessing creativity in machines has become a prime issue in computational creativity after many systems have been built that exhibit a behavior that can intuitively be considered creative. The domains of such systems are so varied, and the versions of each one so many, that comparing them to one another or with different versions of themselves has become a hard task. Every system that claims to be creative must have criteria associated as to what kind of creativity it aims to achieve. After all, in different domains, different notions of creativity may be established.

Even so, recently, several frameworks and models of creativity have been advanced to try to capture a generic or general notion of creativity for computer systems (cite: Ritchie 2007, Wiggins 2006, Jordanous 2012, Maher et al. 2013, Colton et al. 2011). They all propose a practical method to unify criteria across the community so that creativity can be measured in systems from different disciplines of application.

Computer programs, so far, are designed to produce valuable things for humans. Therefore their creativity is always assessed against human values or needs. This is an unfair situation since it is very hard to program computers to produce valuable objects for humans when these are not well defined and only they can say whether they are valuable or not.

If computers were subject to a survival economy like living things in the planet, as Stuart Geiger suggests (2012), then it would be easier to establish what is valuable for them and hence a process parallel to human creativity could be defined to assess creativity by computers. But, for the time being, computers are still doomed to serve our purposes and their creativity will be assessed by human standards.

Creative computer systems are still considered in a separate realm to human creativity in practice. They are often assessed against toy scenarios or their products considered as computational creativity (as opposed to simply creative) to avoid measuring them against human products or creativity. This leaves in the observer the decision of whether the system's behavior is creative in general terms or could be generalized to reach a state where it could be considered so.

In creativity we still don't assign the same expectations to machines and humans. Yet the very concept is defined with respect to the latter. So, either human or machine, creativity is not the same, in the sense that they fulfill the same expectations, or they should be assessed by the same standards.

But as computers get more involved in creative processes, it is possible to view them as participants and describe what they do as playing a role in a team (Jones et al. 2012). It is possible to interpret the process leading to a creation as collaboration between humans and computers and assign roles to all of them according to their activities.

Our view of creativity evaluation is that, although there can be many axes along which it can be measured that seem to be common to several disciplines, ultimately, actual criteria seem to be elusive, arbitrary, subjective and ever changing. These characteristics of creativity don't seem to be problematic to society and most people accept them. It is when we require precision to measure the performance of computer systems that vagueness is problematic. The only way we have to tell whether a computational system is creative, is by inserting it into a human environment and ask humans to assess whether the outcome of the process is creative in the general sense of the term.

Thus a concert composed by a computer, for example, will have to be listened by the same group of human experts who would decide whether a composition made by a human is creative, in the general sense applied in music.

In this text, we describe a framework we call the *apprentice framework* to plan and evaluate CC projects. It

derives from our multidisciplinary experience in an ongoing project called e-Motion (Negrete, S. & Morales, N. 2013), aimed at building a creative system to produce *animatics*. These animated shorts, precursors to a final animation, are an essential element of the overall creative process. In the project we examine the relationship between a computing system and the human counterparts that collaborate with it within a successful, creative team.

Where is Creativity?

Creative products are the result of creative processes. These can take infinite forms but we identify four aspects of creations (creative pieces). Aspects are properties of creations that may be the result of creative work. They are identifiable as the results of separate mental processes that may have occurred at separate times, and may have even been performed by different people:

Structure is the basic architecture of a piece; it is what allows spectators to make out different parts of it, to analyze it to understand its main organization.

Plot is the specialization scaffold of the structure to one purpose; it is the basis for narrative and the most detailed part of planned structure. It is upon plots that pieces are rendered.

Rendering is a particular way in which the plot was developed and filled with detail in order to be delivered to the audience.

Remediation is the transformation of a creative piece already rendered into another one, re-rendered, possibly into another media.

We now discuss some examples of this model in different creative disciplines.

Music. If we consider a piece of music, for example, a composer can be innovative in the structure: a new form of concert, symphony or even something not known to this day; she may also be innovative in the plot: a new score, that is a new piece of written music; a new concert, for instance.

Musicians can also be innovative in the rendering of the piece. That is the execution of the score with the realization of all the details needed to deliver the piece to the audience: a performance. Or they can also do remediation: transcribe an already composed piece of music from a string quartet to a rock band, for example.

Literature. Here the structure refers to the genre. The most general structure of texts: tragedy, satire, comedy, etc. Plot is the structure of a particular story and rendering is the process to transform a plot into a complete literary piece that the audience can read. The rendered piece can also be subject to a process of remediation and be adapted to cinema or theater, etc.

Performing arts usually concentrate on elaborating different renderings for given plots (scores). Each staging of a play in a theater is, in the terms used here, the rendering of a plot. That is, the specification of all the details needed for the audience to receive the original idea. If the performance is improvised, then the performers create, at the same time, both plot and rendering for the audience.

Visual arts. In painting, we can consider the plot to be the sketch on a canvas, the initial drawings where the composition is outlined and the main elements designed. The rest of the work has to do with filling the details to complete the painting: details, colors, texture, etc. This is what we call the *rendering*. In this context, the structure aspect of the painting is its general description as a piece of art: oil on canvas.

The audience perceives, in first instance, the rendering, then the plot and finally the structure. They go from the most emotional aspect of the set, to the most intellectual or logical one. Remediation may or may not be part of the piece, it is only included when a certain translation from media to media is needed.

The rendered piece produces emotion in the audience while the plot produces understanding of the design behind the piece. Plot and structure enable communication, rendering and remediation, expression.

All aspects are present in a piece in different degrees: in some works of abstract art like Jackson Pollock's paintings, plot plays a minimal role, rendering is the most important aspect, there is hardly any structure, the emotion produced by the lines and colors is what constitutes its main expressive motif. In some pieces of conceptual art, on the other hand, structure and plot are the most important aspects while rendering is not as important. In Gabriel Orozco's *Cats and Watermelons*, the number, order, size or disposition of the cans or the fruits (*rendering*) is not as important as the idea behind it all, the plot.

One important thing about these four aspects of creativity is that they are not stages in the creative process, but they emerge during such a process, in any order or simultaneously. These properties might be the result of creative activity of and individual or a collective instance and they influence each other. Distinction between each, can characterize different forms of creativity.

An art piece puts more emphasis on rendering while Design does it in plot. Literature and visual narratives strive to attain a balance between the two in order to maintain the equilibrium between clarity and expression to be enjoyed by an audience (McCloud, S. 2006).

The Role of the Computer in a Creative Team

Computers and computer programs are often used in creative processes. They can be used to store information, as tools, as means of displaying work, and many more. But not all of them have the same degree of creativity. We therefore distinguish five roles a computer program can play in a creative process:

Environment. The computer is a medium where other members of the team can store, display, transmit and, in

general, act as an environment where the work is created.

Toolkit. The computer is used by members of the team, as a set of tools to transform and shape the work creation.

Generator. The computer has been programmed to generate specimens or prototypes of partial or complete pieces of work that meet correctness rules. That is, the specimens belong to the desired kind (chairs, paintings, sonatas, stories, etc.) and team members can adjust parameters in order to vary the specimens generated. The final piece of work is either selected from the set generated or it is an elaboration of some elements of the set.

Apprentice/assistant. The computer produces a reduced set of prototypes that, besides being correct members of the desired kind, they also fulfill some of the properties of creative products: e.g. valuable, innovative, surprising, etc. In this case, other team members have to choose the best of the candidates proposed by the system according to some more subjective human criteria (e.g. trendiness, politics, commission requirements, etc.).

Master. The computer produces a complete and finished work that is considered creative by the designated experts. The rest of the team does management and configuration of the system and handling of the finished work.

The environment role is the most common use of computers for creative purposes. Many people performing creative tasks have found that computers provide them with a suitable environment to work digitally on their subject matter. Working within a computational environment is often simpler, cheaper and more efficient.

Another common role for computers is that of toolkit. Programs like Photoshop and many more like it that provide a set of tools a user can apply interactively and see the work progress are also ever more popular amongst artists. These systems have become indispensable for artists and creators and many activities like photography have already integrated tools like Photoshop into the basic set of tools for the profession.

Many sophisticated systems apply a set of well-studied rules to produce correct pieces of work. These works are easily identifiable as part of the desired kind (poem, tale, motet, sculpture, etc.). It is useful to develop systems like these because they raise the level of abstraction in the process of creation. The programs generate works that can be considered candidates (or nearly) for a final. People using the system modify its parameters in order to alter the generation process and thus obtain better specimens.

The user can be subtracted from the problem of assembling a product and concentrate on a new process by which the machine assembles the product and the user considers whether it is good enough or it needs to be modified somehow. Works produced by a generator system may be novel to itself, but not necessarily to the rest of the world. As we've said before, it takes a human eye to tell. Yet the generation process may expedite the overall process by speeding up a trial and error cycle.

An apprentice system is one that has managed a new level of sophistication by showing a degree of knowledge that produces work specimens that fulfill general criteria for creativity (e.g. valuable, innovative, and surprising). Perhaps going from generator to apprentice is the challenge most computational creativity systems are facing in recent days. It can be seen as a search problem: moving from trial-and-error up to informed search methods.

This last level is set as a reference. In the upper limit, a system that does all the important work and delivers a finished product that can be ascribed to a creative process is the ultimate capacity a computer system can acquire.

We find several advantages to the model just described for the development of computational creativity:

1. A machine embedded in a creative process ensures that any development of the programs in it can be checked to see how much impact it has on the overall creative process. In particular, it is possible to verify whether the outcome of the whole process is still creative, thus eliminating the problem of generalizing toy worlds.
2. Versions of programs can be benchmarked according to the roles they are expected to play.
3. A staged plan for a research program can be drawn with clear goals and strategies based on roles assigned to participants.
4. The four aspects of creative products we described help teams to identify, for a particular role, what it is trying to achieve and decide how to evaluate its performance.

Evaluating Creativity as Participation

We have just described a framework that, we believe, can be used to assess creativity in a computational system by using it combined with already known methods from other fields like Design and applied Arts. These fields use participative and integrating methods to find out what is desirable and valuable for people (Ranjan, P.M. 2013).

Participatory approaches are about including participants in the process of creation of product or experience. Evaluation aspects in this kind of projects are needed to measure impact and performance in the roles participants play.

Nina Simon has developed a way to evaluate impact of participation of visitors in the context of museums and we think is relevant for our task. Her method consists of three main steps:

1. Stating the project goals.
2. Defining behaviors and outcomes that reflect those goals.
3. Measuring or assessing incidence and impact of the outcomes via observable indicators.

Based on Simon's model and using the apprentice framework described above, we can know how to proceed to either develop or assess a creative computational system. We should start by identifying which aspect of creativity is

being emphasized, by doing so we are setting constrains and framing the context to work. This would drive the statement of project goals. Then we need to identify a particular role and skill that the computational system is expected to have by taking explicit knowledge from the humans members. Setting the skills and responsibility of the computer in the overall creative process would be the criteria for constant evaluation and modification of the system.

It is important to stress that participatory projects often benefit from incremental and adaptive measurement techniques. Many creative outputs are process-based. So they have to be valued many times and incrementally before the project ends so that they stay aligned with the goals and all those involved are satisfied. (Simon, N. 2012).

Conclusion

Our experience with eMotion has led us to question many of the underlying principles of CC. We have found by looking at a complex creative human team that it is difficult to pinpoint where creativity really lies. All members of the team can be credited for some percentage of the overall creativity. In the very same way, machines partaking in the process can be assigned their own share of credit and be considered creative too. This view of creativity as part of a process that also gives context seems more promising as a generic framework to develop creative systems than the traditional view of a system designed to perform well in the whole process from the start. Often, the parameters of creative behavior in media projects are either not known from the beginning or highly subjective. Therefore, setting off to develop a computer system that plays a creative role in a team and can be readily assessed by the other members of the team, as it would happen with human members, gives a perspective where several levels of proficiency can be planned ahead and assessed. Other frameworks share similar ideas with our framework (Jones et al. 2012, Colton et al. 2011). The main difference with ours is that we erase the difference between assessing human creativity and machine creativity and try to establish a common methodology. Our framework seeks to evaluate different roles within a creative group, regardless of their being played by a person or a computer program.

A creation (the result of a creative process), can have several creative components, built by sub-process that can also be considered creative. In some disciplines, this is recognized explicitly: in cinema, many prizes around the world, like the Oscars, recognize a whole movie, but also, separately, other creative sub-processes and products: script, musical score, set design, etc. Each one of these is valued under different sets of rules and criteria by people who are experts in those areas. Yet, all those sub-processes contribute to a whole movie, which, in turn is valued on its own right.

In many creative projects these sub-products can be identified and evaluated separately. CC systems can be inserted as part of a team to take a specific role to create a particular creative sub-product. This view allows CC systems to be provided of a context where their development can be planned and they can be properly evaluated.

The four aspects of creative products we have mentioned in this paper allow teams to decide where a particular role is supposed to be innovative and, therefore, how it ought to be assessed.

References

Colton, S., Charnley, J. and Pease, A. 2011. Computational Creativity Theory:The FACE and IDEA Descriptive Models. *Proceedings of the second international conference of Computational Creativity*.

Jones, D., Brown, A. and d'Inverno, M. 2012. The extended composer, in McCormack, J. and d'Inverno, M. *Computers and Creativity*, 175-203, Springer Berlin: Heidelberg.

Jordanous A. 2012. *Evaluating Computational Creativity: A Standardised Procedure for Evaluating Creative Systems and its Application*. Unpublished PhD thesis. Department of Informatics, University of Sussex.

Maher, M. L. and Fisher D.H. 2012. Using AI to Evaluate Creative Design. In *The 2nd International Conference on Design Creativity* (ICDC2012) Glasgow, UK.

Mary Lou Maher, Katherine Brady, Douglas H. Fisher. 2013. Computational Models of Surprise in Evaluating Creative Design. In *The Fourth International Conference on Computational Creativity*. Sydney, Australia.

Negrete-Yankelevich, S. and Morales-Zaragoza, N. 2013 e-Motion: a system for the development of creative animatics. In *Proceedings of the fourth international conference of Computational Creativity*. Sydney 184-188, ICCC2013

McCloud S. 2006. *Making Comics: Storytelling Secrets of Comics, Manga and Graphic Novels*. Harper Collins Publishers. 19-25, 336-2900.

Geiger, S; 2012. The Ethnography of Robots. Interviewed by Heather Ford for Ethnography Matters. Accessed January 28th 2014. http://ethnographymatters.net/2012/01/15/theethnographyof-robots/

Ranjan, P. M. 2013. *Design Thinking: Workshop for designers and Craftpeople*. Design Innovation and Craft Resource Center (DICRC), CEPT University, Ahmedabad, India. 22.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, 17, 67–99.

Simon, N. 2012. The Particpiatory Museum. Museum 2.0 Santa Cruz, California. 301-307.

Wiggins, G. A. 2006. *A preliminary framework for description, analysis and comparison of creative systems.* Knowledge-Based Systems, 19(7), 449–458.

Criteria for Evaluating Early Creative Behavior in Computational Agents

Wendy Aguilar
Posgrado en Ciencia e Ingeniería de la Computación,
UNAM, México D.F.
weam@turing.iimas.unam.mx

Rafael Pérez y Pérez
Departamento de Tecnologías de la Información,
División de Ciencias de la Comunicación y Diseño
UAM Cuajimalpa, México D.F.
rperez@correo.cua.uam.mx

Abstract

Our research is focused on the study of the genesis of the creative process. With this purpose we have created a developmental computational agent, which allows us to watch the generation of the first behaviors we could consider as creative. It is very important to develop methodologies to evaluate the behaviors generated by this kind of agents. This paper represents our first effort towards that end. Here we propose five criteria for its evaluation, and we use them to test the behaviors created by our developmental agent.

Introduction

The construction of artificial systems which simulate the creative process is currently a topic of great interest among the artificial intelligence and cognitive sciences community. A great effort has been made to build methodologies that help us evaluate such systems (e.g. Ritchie, 2007; Colton, 2008; and Jordanous, 2012). Nevertheless, it has not been an easy task, and it is necessary to do more research. This article is intended to contribute to it. Our research is focused on the study of the genesis of the creative process, which takes place during the first years of our lives, as explained in the next section. With this purpose we have created a computational agent that simulates cognitive development (introduced in Aguilar and Pérez y Pérez, 2013), which allows us to watch the generation of the first behaviors we could consider as creative. In this article, we focus on proposing some criteria which may let us evaluate the behaviors generated by this kind of agents.

Concepts and Definitions

Creative Behavior

In the literature on the subject we can find a number of definitions for creative behavior. For example, from a behaviorist viewpoint, Razik (1976) defines creative behavior as a unique response or pattern of responses to an internal or external discriminative stimulus. Or, from the point of view of artificial systems, for example Maher, Merrick, and Saunders (2008) propose that the developing of creative behavior in artificial systems focuses on the automatic generation of sequences of actions that are novel and useful. This article is based on Cohen's point of view (1989), who describes creativity as a series of adaptive behaviors in a continuum of seven levels of development: initially, creativity involves adaptation of the individual to the world; and at higher levels, it involves adaptation of the world to the individual. For the context of this article, we will focus only on the first level, called "Learning Something New: Universal Novelty". This kind of creativity is that resulting in behaviors that are useful and new to the individual, but not strange or valuable to others. Cohen considered that it can be observed in babies and toddlers as a result of their need to start to adapt to the world.

Adaptation

For Piaget (see for example Piaget, 1952), adaptation takes place by means of two complementary processes he called assimilation and accommodation. The assimilation process allows children to face new situations by using their knowledge from past experiences. However, in some cases, the situations they face contradict its current knowledge of the world. In these cases of conflict, the accommodation process allows children to deal with new situations by progressively modifying their knowledge (throughout the continuous interaction with their environment) in order to include the results of their new experiences. In this way, Cohen's first level creative-adaptive activity helps us adapt to our world either modifying our perception of the environment so that it fits our knowledge acquired from past experiences (it is, adaptation by assimilation), or modifying and producing new knowledge when it does not match reality (it is, adaptation by accommodation).

Cognitive Development

Piaget considered that when children interact with their environment by using their previously acquired knowledge, they are in a stage called cognitive equilibrium. Whereas when the interaction with their environment causes a conflict between their knowledge and reality, they then experience a crisis moment called cognitive disequilibrium. He also suggested the change from equilibrium to disequilibrium and back to equilibrium (through accommodation) promotes children evolving across four continuous qualitatively different stages, from birth to adulthood (the interested reader can find a brief summary of his theory in Crain, 2010, chapter 6). The first of them is called sensorimotor stage, which starts at

birth and finishes at around 2 years old (approximately the same age in which Cohen's first level adaptive creativity is observed). According to his theory, the sensorimotor stage is subdivided into six substages, each characterized by the emergence of new behaviors. For example, the second substage is characterized by the acquisition of behaviors centered on his body, such as learning how to follow any object visually or how to keep objects of interest grasped; whereas the third substage is characterized by the acquisition of behaviors involving consequences on external objects, such as learning how to squeeze a rubber duck in order to have it quack.

This first stage of development is quite interesting from the point of view of creativity. On the one hand because it is in this stage that the children's behaviors start to be goal oriented; this is the beginning of means-end differentiation, a basic skill to become capable of solving problems. And problem solving has been considered as a form of creativity (Runco 2007). On the other hand, because Piaget himself considered it as the most creative period of life, since it is during this stage that newborns must start to build their knowledge of the world, and such construction requires creativity (Runco and Pritzker 1999, p. 13). So, during this period, children's first manifestations of creative behavior arise.

Piaget called the evolution through the different substages and stages *cognitive development*.

Evaluation Criteria

Inspired by Maher, Merrick, and Saunders's paper (2008), we propose that an artificial agent that simulates cognitive development (e.g. Stojanov 2001; and Aguilar and Pérez y Pérez 2013) generates creative behaviors if they comply with the following characteristics:

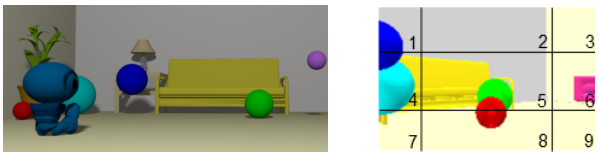
- **Novelty**. A behavior is considered novel if it did not exist explicitly on the agent's initial knowledge base.
- **Usefulness.** A behavior is considered useful if it serves as a basis for the construction of new knowledge that eventually leads the agent to acquire new skills that are characteristic of its next developmental stage. For example, those driving it from behaviors characteristic of the second substage of the sensorimotor period (behaviors based on the body) to behaviors characteristic of the third substage (behaviors involving consequences on external objects).
- **Emergence**. Based on Steels's (2014) definition, we propose to consider a behavior as emerging if its origin cannot be directly traced back to the system's components, but if it originates as a result of the way such components interact.
- **Motivations.** Amabile (1983, 1999) distinguished between two types of creativity: intrinsically and extrinsically motivated creativity. Intrinsic motivation refers to a behavior that is driven by internal rewards, while extrinsic motivation is focused on external reward, recognition or punishment avoidance. In this article we propose that a behaviour that an agent develops should be considered creative only if it resulted from an intrinsic and/or extrinsic motivation.
- **Adaptation to the environment.** The ability to adapt to our environment has been seen traditionally (perhaps as of Darwin) as a necessary condition for really creative behavior (Runco 2007, p. 398). We therefore propose that a behavior that an agent develops should be considered creative only if it resulted from an agent's adaptive process to its environment.

Case Study

In order to illustrate the application of the evaluation criteria we propose, we assessed the agent presented in (Aguilar and Pérez y Pérez 2013).

Brief description of the agent

The agent lives in a 3D virtual environment with which it interacts. It can lift its head, move it down and turn left and right; as well as open and close its hand. It has a visual and a tactile sensor. The visual sensor is implemented as a virtual camera with a field of vision of 60 degrees. Its field of vision is divided into the nine areas shown in Figure 1b. It implements five main cognitive capabilities: 1) it can see and touch its world; 2) it simulates an attention process; 3) it simulates affective responses of pleasure and displeasure (represented by variables with values -1 for displeasure and +1 or +2 for two intensities of pleasure), emotional states of interest, surprise and boredom, and an intrinsic motivation of cognitive curiosity (represented by boolean variables with value "true" when the agent shows such state or motivation); 4) it has a memory where it stores its knowledge on how to interact with its world, and it does so in structures called schemas of which there are two types: *basic schemas* representing default or "innate" behaviors (defined as two-part structures consisting of a context and an action), and *developed schemas* representing behaviors created by the agent as it interacts with its world (defined as three-part structures consisting of a context, an action, and a set of expectations); and 5) it simulates an adaptation process which is inspired by Piaget's theory.



(a) The agent in its virtual world (b) The division of the agent's field of vision

Figure 1: The agent and its virtual world

The agent interacts with its environment: 1) by sensing its world, 2) by choosing one of the sensed objects as its center of attention, 3) by choosing what action to carry out, and 4) by executing the chosen action. Steps 1 to 4 are called *perception-action cycle*.

Its central component is its adaptation module called *Dev E-R* (*Developmental Engagement-Reflection*). It is imple-

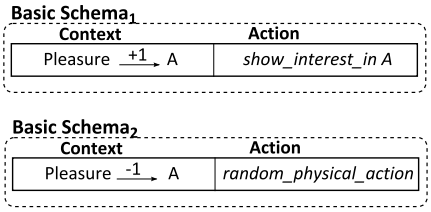


Figure 2: Initial basic schemas. They represent the initial behaviors the agent knows for interacting with its world. $Basic\ Schema_1$ represents the tendency to preserve a pleasant stimulus, and $Basic\ Schema_2$ represents the tendency to perform a groping to get a pleasant stimulus back when it disappears.

mented with a new extended version of the computational model of the creative process *Engagement-Reflection* (Pérez y Pérez and Sharples 2001). Dev E-R simulates the assimilation process by searching in the memory for schemas representing similar contexts to the current perceived situation (which is defined in terms of the current agent's affective responses, emotional states and motivations). On the other hand, Dev E-R models the accommodation process by creating new schemas and/or modifying the existing ones as a result of dealing with new situations in the world. The creation and modification of the schemas takes place by means of one of the two following methods: generalization or differentiation. This way, the agent interacts with its virtual world, assimilating and accommodating its knowledge, until it manages to reach a cognitive equilibrium state. It is, until it does not need to modify its schemas during the last NC cycles, since they allow it to interact with its environment satisfactorily. When the agent reaches a cognitive equilibrium state, it is able to face new situations by partially using its knowledge from past experiences. This may cause new schemas to be built, having the agent enter a cognitive disequilibrium state again. It is this way that the agent goes from equilibrium states to disequilibrium states frequently, until it stops its development because it keeps equilibrium for a certain number of cycles. It is then that its execution ends.

Testing

In Aguilar and Pérez y Pérez (2013) it was reported that the agent interacted with the environment shown in Figure 1a, for 9000 cycles. In this world there were balls in different colors and sizes, moving from the left to the right and downwards, independently. Nevertheless, they never made contact with the agent's hand. It was also reported that the agent was initialized with two basic schemas representing behaviors characteristic of the first substage of the sensorimotor period (shown in Figure 2).

Novelty At the beginning of the agent's execution, it constantly lose the objects of its interest. This was due to the fact that it could only use its basic predefined behaviors to interact with its world. When this happened, the behavior it showed was a random movement of its head (resulting from the use of its $Basic\ Schema_2$). Nevertheless, after letting the agent interact with its environment for 9,000 cycles, it built 13 new schemas that did not exist in its initial knowledge base. The seven first, called *schemas type 1*, represented behaviors meant to recover the objects of interest it had lost in different positions of its field of vision. For example, if it lost an object on its right, then its schema indicated to turn its head right, generating the expectation of recovery. The six next schemas, called *schemas type 2*, represented behaviors meant to keep the objects of interest within its field of vision, causing the number of objects it lost to progressively decrease and even reach zero. The 13 schemas the agent built are different in structure and contents; and, more important, they represent different behaviors to those it was initialized with. Although it is also important to note that the seven schemas of the first type are very similar among themselves, since all of them represent behaviors of recovery for lost objects of interest. Similarly, the six schemas of the second type are very similar among themselves, since all of them represent how to keep the objects of its interest within its field of vision. We can therefore conclude the agent built 2 groups of schemas, schemas type 1 and schemas type 2, that represent totally novel behaviors to the agent (recover and keep objects of interest). It also built 7 and 6 schemas within such groups, representing behaviors less novel among themselves.

Usefulness In order to evaluate the usefulness of the behaviors the agent developed, lets remember it was initialized with behaviors characteristic of the first substage of the sensorimotor period. From there, throughout the interaction with its environment, it built its first seven schemas related to recovering the objects of interest (schemas type 1). These were later used as a base, on partially using them in the new situations it faced, in order to build the following six schemas related to keeping objects of interest (schemas type 2). The use of these 13 schemas together caused the agent to show the new behaviors of following the objects of interest visually (by moving its head), and of keeping them centered within its field of vision most of the times. These two new abilities that the agent acquired were described by Piaget as two of the main abilities related to vision which children develop during the second substage of the sensorimotor period. Therefore, the schemas the agent developed are considered useful since they allowed it to go from predefined or "innate" behaviors (typical of the first substage of the sensorimotor period) to behaviors based on its body (typical of the second substage of the sensorimotor period).

Emergence The construction of the different behaviors the agent develops depends on various factors, among them: 1) the characteristics of its environment, 2) its physical characteristics, and 3) its current knowledge. For example, regarding the first point, if the agent lived in a world in which it always had to lift its head in order to recover the objects, it would build schemas representing that particular characteristic of its environment. Similarily, if the agent were enabled with the ability to touch but not to see its world (it means, if it were blind), it would develop different behaviors to those it developed with vision (using exactly the same adaptation processes in both cases). Except that now, the new abilities would be related to touching. For example, it would learn

how to keep the object of its interest grasped. Also, regarding the third point, the behaviors the agent develops depend on its current knowledge. For example, schemas type 2 require the construction of schemas type 1 first, since it is not until they exist and are stable that those type 2 can originate. This is because the agent uses its knowledge on how to recover objects of interest in the different positions in order to learn how to keep them within its field of vision. We can therefore conclude that the behaviors created by the agent emerged as a result of the way the different components of the system interacted among themselves, since the new behaviors were not set up by default, and also because they are contextual. It is, because they depend on its interaction with the environment, on its sensory abilities and on its current knowledge.

Motivations One of the core components of the agent is that it simulates affective responses, emotional states and an intrinsic motivation of cognitive curiosity that push it to act. Particularly, regarding the development of new schemas, they are created, modified or eliminated as a result of the triggering of: 1) an emotional state of surprise (for example, caused by the unexpected recovery of an object of interest), or 2) a cognitive curiosity motivation (that is generated when dealing with unknown situations that contradict its current knowledge of the world). So, in this model, the emotional state of surprise and the intrinsic motivation of cognitive curiosity trigger the necessity of modifying and building new schemas.

Adaptation to the environment The schemas the agent developed originated as a consequence of its facing new unknown situations, to which it reacted whether assimilating the new situation into its acquired knowledge (by means of the process of searching a schema representing a similar situation to that of its current context in memory) or accommodating its knowledge so that it fitted the new experience (by creating a new schema or by differentiating, generalizing or deleting an existing one). Therefore, the construction of new schemas took place as a result of a complementary assimilation and accommodation process. In other words, they originated from an adaptation process of the agent to its world.

Conclusions

In this article we propose five criteria to evaluate if the behaviors created by agents that simulate having cognitive development can be considered creative. The criteria we propose are: novelty, usefulness and emergence. Additionally, it is requested that such behaviors had originated as a result of intrinsic and/or extrinsic motivations, as well as of the adaptation to its environment. The results of the evaluation of the agent of the case study showed that, under these criteria, the first behaviors it develops (learning how to follow objects of its interest visually and how to keep them centered within its field of vision) are considered creative. These results represent our first approach to the evaluation of this kind of agents. There is still much more research to do on this matter.

Acknowledgements

This research was sponsored by the National Council of Science and Technology in México (CONACYT), project number 181561 and doctoral scholarship number 239740.

References

Aguilar, W., and Pérez y Pérez, R. 2013. A computer model of a developmental agent to support creative-like behavior. In *AAAI Spring Symposium: Creativity and (Early) Cognitive Development*, 8–13.

Amabile, T. M. 1983. *The social psychology of creativity*. Springer-Verlag, New York.

Amabile, T. M. 1999. *Motivation and creativity. In: R.J. Sternberg (Editor), Handbook of creativity*. Cambridge University Press, Cambridge.

Cohen, L. 1989. A continuum of adaptive creative behaviors. *Creativity Research Journal* 3.

Colton, S. 2008. Creativity versus the perception of creativity in. In *Proceedings of AAAI symposium on creative systems*, 1420.

Crain, W. 2010. *Theories of Development: Concepts and Applications*. Pearson.

Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4:246–279.

Maher, M. L.; Merrick, K.; and Saunders, R. 2008. Achieving creative behavior using curious learning agents. In *AAAI Spring Symposium on Creative Intelligent Systems*.

Pérez y Pérez, R., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence* 13:2:119–139.

Piaget, J. 1952. *The Origins of Intelligence in Children*. London: Routledge and Kegan Paul, 1936 (French version published in 1936, translation by Margaret Cook published 1952).

Razik, T. 1976. Programming creative behaviour. *British Journal of Education Technology* 3:5–21.

Ritchie, G. 2007. Some empirical criteria for attributing creativity. *Minds and Machines* 17:67–99.

Runco, M., and Pritzker, S. 1999. *Encyclopedia of Creativity.* Academic Press.

Runco, M. 2007. *Creativity Theories and Themes: Research, Development,*. Academic Press in an imprint of Elsevier.

Steels, L. 2014. Towards a theory of emergent functionality, from animals to animats. In *First International Conference on Simulation of Adaptive Behaviour*, 451–461. Bradford Books (MIT Press).

Stojanov, G. 2001. Petitagé: A case study in developmental robotics. In *C. Balkenius, J. Zlatev,H. Kozima, K. Dautenhahn, and C. Breazeal, editors, Proceedings of Epigenetic Robotics 1*.

COINVENT: Towards a Computational Concept Invention Theory

Marco Schorlemmer,[1] **Alan Smaill,**[2] **Kai-Uwe Kühnberger,**[3] **Oliver Kutz,**[4] and **Simon Colton**,[5] **Emilios Cambouropoulos**[6] and **Alison Pease**[7]

[1]Artificial Intelligence Research Institute, IIIA-CSIC, Spain [2]School of Informatics, The University of Edinburgh, UK
[3]Institute of Cognitive Science, University of Osnabrück, Germany
[4]Institute of Knowledge and Language Engineering, Otto-von-Guericke University Magdeburg, Germany
[5]Department of Computing, Goldsmiths, University of London, UK
[6]School of Music Studies, Aristotle University of Thessaloniki, Greece [7]School of Computing, University of Dundee, UK

Abstract

We aim to develop a computationally feasible, cognitively-inspired, formal model of concept invention, drawing on Fauconnier and Turner's theory of conceptual blending, and grounding it on a sound mathematical theory of concepts. Conceptual blending, although successfully applied to describing combinational creativity in a varied number of fields, has barely been used at all for implementing creative computational systems, mainly due to the lack of sufficiently precise mathematical characterisations thereof. The model we will define will be based on Goguen's proposal of a Unified Concept Theory, and will draw from interdisciplinary research results from cognitive science, artificial intelligence, formal methods and computational creativity. To validate our model, we will implement a proof of concept of an autonomous computational creative system that will be evaluated in two testbed scenarios: mathematical reasoning and melodic harmonisation. We envisage that the results of this project will be significant for gaining a deeper scientific understanding of creativity, for fostering the synergy between understanding and enhancing human creativity, and for developing new technologies for autonomous creative systems.

Introduction

Of the three forms of creativity put forward in (Boden 1990)—combinational, exploratory, and transformational—the most difficult to capture computationally turned out to be the combinational type (Boden 2009), i.e., when novel ideas (concepts, theories, solutions, works of art) are produced through unfamiliar combinations of familiar ideas. Although generating novel ideas, or concepts, by combining old ones is not complicated in principle, the difficulty lies in doing this in a computationally tractable way, and in being able to recognise the value of newly invented concepts for better understanding a certain domain; even without it being specifically sought—i.e., by 'serendipity' (Boden 1990, p. 234), (Pease et al. 2013).

To address this problem, we will concentrate on an important development that has significantly influenced the current understanding of the general cognitive principles operating during creative thinking, namely Fauconnier and Turner's theory of *conceptual blending*, also known as conceptual integration (Fauconnier and Turner 1998). Fauconnier and Turner proposed conceptual blending as the fundamental cognitive operation underlying much of everyday thought and language, and modelled it as a process by which people subconsciously combine particular elements and their relations, of originally separate mental spaces, into a unified space, in which new elements and relations emerge, and new inferences can be drawn. For instance, a 'houseboat' or a 'boathouse' are not simply the intersection of the concepts of 'house' and 'boat'. Instead, the concepts 'houseboat' and 'boathouse' selectively integrate different aspects of the source concepts in order to produce two new concepts, each with its own distinct internal structure (see Figure 1 for the 'houseboat' blend).

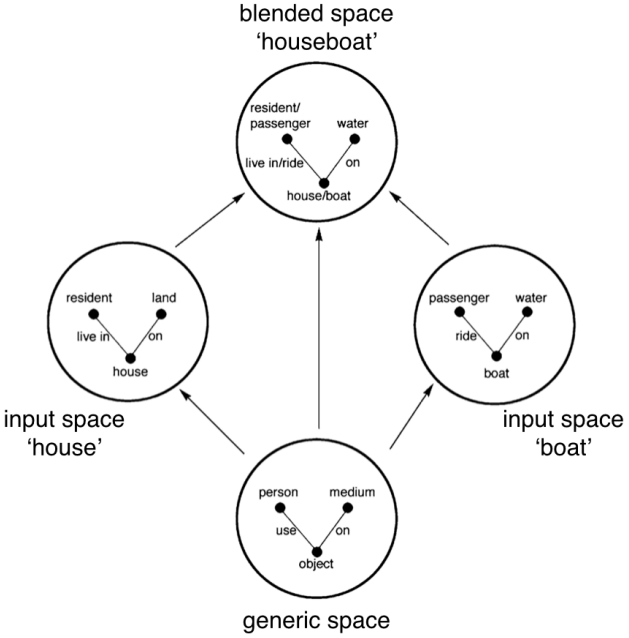


Figure 1: 'Houseboat' blend, adapted from (Goguen and Harrell 2010)

The cognitive, psychological and neural basis of conceptual blending has been extensively studied (Fauconnier and Turner 2003; Gibbs, Jr. 2000; Baron and Osherson 2011). Moreover, Fauconnier and Turner's theory has been successfully applied for describing existing blends of ideas and concepts in a varied number of fields, such as linguistics, mu-

sic theory, poetics, mathematics, theory of art, political science, discourse analysis, philosophy, anthropology, and the study of gesture and of material culture (Turner 2012). However, the theory has hardly been used for implementing creative computational systems. Indeed, since Fauconnier and Turner did not aim at computer models of cognition, they did not develop their theory in sufficient detail for conceptual blending to be captured algorithmically. Consequently, the theory is silent on issues that are relevant if conceptual blending is to be used as a mechanism for designing creative systems: it does not specify how input spaces are retrieved; or which elements and relations of these spaces are to be projected into the blended space; or how these elements and relations are to be further combined; or how new elements and relations emerge; or how this new structure is further used in creative thinking (i.e., how the blend is "run"). Conceptual blending theory does not specify *how* novel blends are constructed.

Nevertheless, a number of researchers in the field of computational creativity have recognised the potential value of Fauconnier and Turner's theory for guiding the implementation of creative systems, and some computational accounts of conceptual blending have already been proposed (Veale and O'Donoghue 2000; Pereira 2007; Goguen and Harrell 2010; Thagard and Stewart 2011). They attempt to concretise some of Fauconnier and Turner's insights, and the resulting systems have shown interesting and promising results in creative domains such as interface design, narrative style, poetry generation, and visual patterns. All of these accounts, however, are customised realisations of conceptual blending, which are strongly dependent on hand-crafted representations of domain-specific knowledge, and are limited to very specific forms of blending. The major obstacle for a general account of computational conceptual blending is currently the lack of a mathematically precise theory that is suitable for the rigorous development of creative systems based on conceptual blending.

A Formal Model of Conceptual Blending

To address the relative lack of study of the computational potential of conceptual blending, in the FP7-ICT project COINVENT[1], we are setting out to:

1. develop a novel, computationally feasible, formal model of conceptual blending that is sufficiently precise to capture the fundamental insights of Fauconnier and Turner's theory, while being general enough to address the syntactic and semantic heterogeneity of knowledge representations;
2. gain a deeper understanding of conceptual blending and its potential role in computational creativity, by linking this novel formal model to relevant, cognitively-inspired computational models, such as analogical and case-based reasoning, induction, semantic alignment, and coherence-based reasoning;
3. design a generic, creative computational system based on this novel formal model, capable of serendipitous invention and manipulation of novel abstract concepts, enhancing thus the creativity of humans when this system is instantiated to particular application domains for which conceptual blending is a core process of creative thinking;
4. validate the model and its computational realisation in two representative working domains: mathematics and music.

The only attempt so far to provide a general and mathematically precise account of conceptual blending has been put forward by Goguen, initially as part of algebraic semiotics (Goguen 1999), and later in the context of a wider theory of concepts: Unified Concept Theory (Goguen 2005a). He has also shown its aptness for formalising information integration (Goguen 2005b) and reasoning about space and time (Goguen 2006).

Goguen's intuition was that conceptual blending could be modelled based on the *colimit* construct of category theory—a field of abstract mathematics that has provided deep insights in mathematical logic and computer science, and has often been used as a guide for finding good definitions and research directions. In his *Categorical Manifesto*, he intuitively describes this construct as follows: "Given a category of widgets, the operation of putting a system of widgets together to form some super-widget corresponds to taking the colimit of the diagram of widgets that shows how to interconnect them." (Goguen 1991)

To model conceptual blending we would start with a collection of input spaces—Goguen defines them as semiotic spaces of signs and their relations—and of structure-preserving mappings between them, capturing how the structure of these spaces is related. The colimit would be the optimal way to put these spaces together into one single space taking into account how they were originally connected by structure-preserving mappings. Here 'optimal' means that the colimit includes all structure of the input spaces, but not more; and that it would not make unnecessary fusion of structure. An important property of colimits is that they are unique up to isomorphism. But since conceptual blending does not operate in general under this notion of optimality, Goguen suggested to extend this idea by including a notion of 'quality' of structure-preserving mappings between mental spaces to cope with the idea of partial mappings that selectively map only certain structure into the blend, and to model conceptual blends as colimits in this extended setting.

As it stands, Goguen's account is still very abstract and lacks concrete algorithmic descriptions. There are several reasons, though, that make it an appropriate candidate theory on which to ground the formal model we are aiming at:

- It is an important contribution towards the unification of several formal theories of concepts, including the geometrical conceptual spaces of (Gärdenfors 2004), the symbolic conceptual spaces of (Fauconnier 1994), the information flow of (Barwise and Seligman 1997), the formal concept analysis of (Ganter and Wille 1999), and the lattice of theories of (Sowa 2000). This makes it possible to potentially draw from existing algorithms that have already been developed in the scope of each of these frame-

[1] www.coinvent-project.eu

works.

- It covers any formal logic, even multiple logics, supporting thus the integration and processing of concepts under various forms of syntactic and semantic heterogeneity. This is important, since we cannot assume conceptual spaces represented in a homogeneous manner across diverse domains. Current tools for heterogeneous specifications such as Hets (Mossakowski, Maeder, and Lüttich 2007) allow parsing, static analysis and proof management incorporating various provers and different specification languages.

By developing a formal model of conceptual blending building on Goguen's initial account, we aim to provide general principles that will guide the design of computer systems capable of inventing new higher-level, more abstract concepts and representations out of existing, more concrete concepts and interactions with the environment, and to do so based on the sound reuse and exploitation of existing computational implementations of closely related models, such as those for analogical and metaphorical reasoning (Falkenhainer, Forbus, and Gentner 1989), semantic integration (Schorlemmer and Kalfoglou 2008), or cognitive coherence (Thagard 2000). With such a formal, but computationally feasible model, we will ultimately bridge the existing gap between the theoretical foundations of conceptual blending and their computational realisations. This, in turn, will contribute to the much-needed foundations for the design of creative systems that effectively enhance both artificial and human creativity when deployed in the kinds of genuinely creative tasks underlying the sort of abstract reasoning common to many branches of the sciences and the arts.

Working Domains

To explore the genericity of the proposed formal model of concept invention and of the computational realisation we are after, we will focus on two representative working domains of creativity: mathematics and music, "the most sharply contrasted fields of intellectual activity which one can discover, and yet bound together, supporting one another as if they would demonstrate the hidden bond which draws together all activities of our mind, and which also in the revelations of artistic genius leads us to surmise unconscious expressions of a mysteriously active intelligence," as noted wisely in (von Helmholtz 1885).

In mathematics, the creative act of providing novel definitions, conjectures, theorems, examples, counter-examples, or proofs can be seen as particular cases of concept invention (Montano-Rivas et al. 2012). In music, concept invention may apply to the generation of new melodies, harmonies, rhythms, or counterpoints (and their combination) (Mazzola, Park, and Thalmann 2011), and to the integration of musical and textual spaces to achieve novel musical metaphors (Zbikowski 2002).

The following examples illustrate the sort of creative activity we want to address with our formal model and its computational realisation.

Example 1 The historical example of the discovery of the quaternions by Hamilton is one that is well documented (e.g., (Hersh 2011)), so much is known about the intermediate steps involved in the discovery. This can be treated by our approach, by taking the starting point as the unproblematic blend between the algebraic structure of the complex numbers as a field (with addition, multiplication and division), and the geometric structure of the 2-dimensional real plane as a real vector space (with addition, and scalar multiplication). In our terms, Hamilton wanted to find a similar blend involving an algebraic structure corresponding to 3-dimensional real vector space. He ended up, however, by finding a blend involving a 4-dimensional real vector space, and the algebra of the quaternions — which involves leaving out from the algebraic theory the commutativity of multiplication. We thus see the characteristic features of blending, in the diagram of Figure 2, where the arrows indicate morphisms in Unified Concept Theory. This shows the characteristic features of blending, where:

- there are two given concepts: commutative fields, and (4-dimensional) real vector spaces;
- a common concept, structurally similar to some aspects of the given concepts is identified (*Common*);
- the initial concepts are blended, respecting the common aspects,
- an initially inconsistent blended concept of quaternions is obtained;
- this is modified by dropping an initial feature (commutativity of multiplication), to obtain a consistent concept.

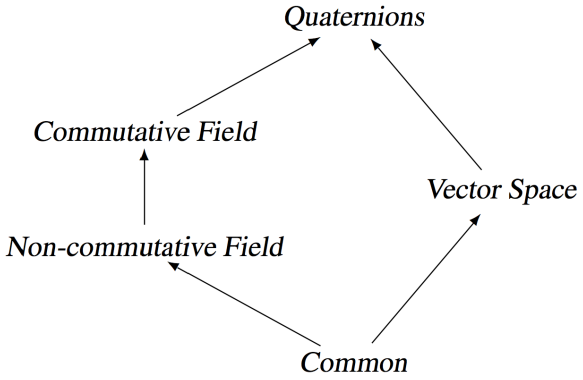


Figure 2: Blend for quaternions.

By deploying COINVENT-based technology in this working domain, our ultimate goal is to transcend the capabilities of current state-of-the-art automated reasoning support tools, which as of today are reluctantly accepted by their users and perceived more as an obstacle to than a facilitator of creative thinking. The choice of the domain of mathematics is further supported by the following reasons:

- Evidence from cognitive science, education, and history of mathematics suggests that the hierarchy of mathematical concepts is grounded on some simple numerical abili-

ties humans have, combined with know-how about physical scenarios of interaction with the environment (Lakatos 1976; Lakoff and Núñez 2000). This means that by tackling the case of mathematics, we need to address problems concerning the situatedness of agents.

- The span of usage of mathematical concepts goes from rather concrete situations (children learning to count how many toys you give them) to the very abstract (as when professional mathematicians do research) — see (Lakoff and Núñez 2000; Alexander 2011).
- Mathematics allows us to explore the social dimension of concept invention and the forces external to cognition that shape the process of conceptual blending over time, crucial in educational and research environments (Lakatos 1976; Goguen 1997).
- Currently, there is no cognitive model of the way in which people invent mathematical concepts; there are to our knowledge no models of how humans create mathematics. Hence only a few computational creativity systems exist that support creative mathematical thinking, such as (Colton 2002).

Example 2 Devising appropriate chordal harmonisations for melodies derived from non-Western cultures or, even, for new creations could potentially be tackled computationally based on our approach. A computational system could autonomously explore different chordal spaces generating novel harmonic combinations/blends appropriate for the melodies at hand. This could be applied for the design of an interactive compositional tool or computer game where the user inputs a melody (may 'sing in' a melody) and the automatic harmonisation system produces interactively novel harmonisations that creatively combine harmonic properties from different music idioms. It could also be applied, for instance, for video-game design and programming, by endowing game creations with the capacity of generating new harmonisations on-the-fly; the creative melodic harmonisation assistant could provide appropriate harmonisations following the mood changes or activity or gestural patterns emerging as the game unfolds. In Figure 3, a traditional melody is harmonised in radically different ways corresponding to individual harmonic spaces (tonal, modal, atonal). The creative harmonisation assistant may generate such original harmonisations or enable the emergence of new unpredicted harmonisations stemming from blends between such spaces.

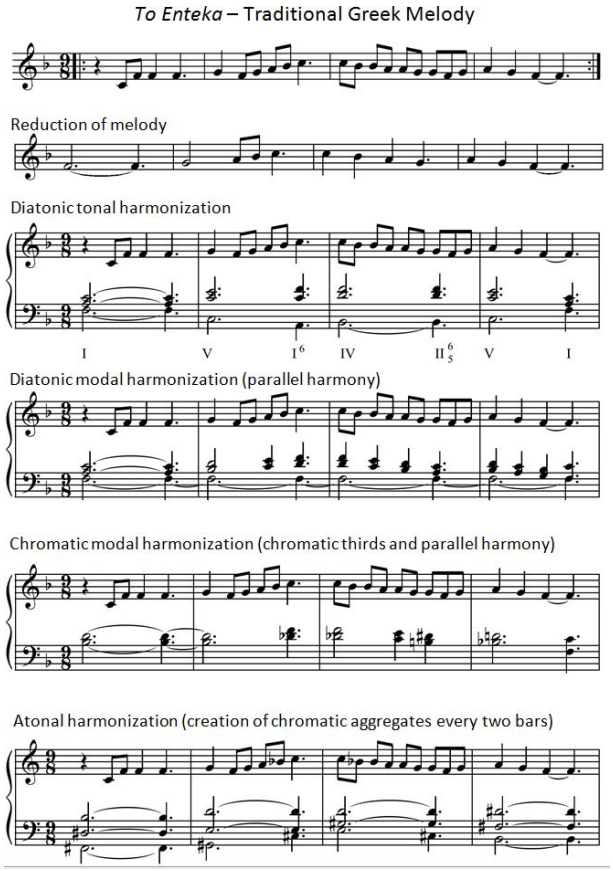


Figure 3: Four different harmonisations of a traditional melody (first four-bar phrase) — harmonizations created by C. Tsougras (Aristotle University of Thessaloniki).

By deploying COINVENT-based technology in this working domain our ultimate goal is to be capable of making software go beyond a mere application of compositional rules, so as to refute the common belief that creativity is separated from the computational processes used in music composition, and that these processes just do uncreative calculations. The choice of the domain of music is further supported by the following reasons:

- The conceptual level of music, together with the role of cognitive models such as conceptual blending in musical analysis, has gained increased attention in the field of music theory (Zbikowski 2002).
- A substantial body of contemporary research on musical creativity from the philosophy of computer modelling, through music semiotics, education, performance and neuroscience, to experimental psychology (Deliège and Wiggins 2006; Mazzola, Park, and Thalmann 2011) provides the necessary background for exploring computational creativity in a scientific manner in the domain of music.
- Traditional music analysis has weak conceptual power for studying complex constructions. Formal theories of musical structure and processes, as employed in contemporary computational modelling of music (Anagnostopoulou and Cambouropoulos 2012; Conklin and Anagnostopoulou 2006; Steedman 1996), are considered an adequate tool for computer-aided composition of advanced music.
- The language of modern mathematics, whose conceptual character has been stressed by contemporary mathematicians (Lawvere and Shanuel 1997; Boulez and Connes 2011), has been advocated as a way forward in the analysis of its effectiveness in musical creativity (Future and

Emerging Technologies 2011).

- Musical creativity, particularly musical performance, is ultimately contextualised, situated, and embodied (Goguen 2004). In particular, in musical gesture theory, conceptual blending has been suggested as a powerful model of musical interpretation (Echard 2006).

We believe that the exploration of the domains of mathematics and music should reveal very general principles applicable to other creative domains.

Relevant Prior Research

COINVENT is a collective effort to advance the understanding of creativity through a precise formalisation of an important cognitive model and a concrete computational realisation thereof. We shall do so informed by the main contributions towards a science of creativity (Sternberg 1999) and drawing from several foundational theories that have hitherto largely been pursued independently.

During the last decades, scholars and researchers in cognitive linguistics and cognitive psychology have made significant contributions to the understanding of the fundamental role that metaphor and analogy play in cognition (Lakoff and Johnson 1980; Gentner, J.Holvoak, and Kokinov 2001; Fauconnier and Turner 2003), at the same time that significant evidence has been gathered supporting a philosophy of mind grounded on the embodiment of mind and meaning (Maturana and Varela 1987; Varela, Thompson, and Rosch 1992; Lakoff and Johnson 1999; Johnson 2007). This research has been heavily influenced by the dramatic progress in imaging techniques carried out in the field of neuroscience, such as functional MRI.

In parallel, the development of the field of Category Theory has led to a remarkable unification and simplification of mathematics (Mac Lane 1971; Lawvere and Shanuel 1997), which has helped to reach a deep understanding across different fields such as computer science, mathematical logic, physics, and linguistics. More recently, these techniques have been applied to obtain some preliminary formalisations of conceptual metaphor and blending (Goguen 1999; Old and Priss 2001; Guhe, Smaill, and Pease 2009) by applying techniques such as institution theory (Goguen and Burstall 1992) or information flow theory (Barwise and Seligman 1997), which are based on category theory.

Automated reasoning techniques from artificial intelligence that are either based on cognitive principles such as case-based reasoning (Aamodt and Plaza 1994) —grounded on the prototype theory of categorisation (Rosch 1973) and reasoning by analogy making (Gentner 1983)—or on formal methods for inductive reasoning such as anti-unification (Plotkin 1971) will be some of the seed technologies for the computational realisation of our model. Some preliminary steps have been made already, in joint research by some of the consortium members of COINVENT, by taking ideas from Lakatos (Lakatos 1976) and from (Lakoff and Núñez 2000) as starting points and extending the HDTP system (Heuristic-Driven Theory Projection, developed at the University of Osnabrück (Gust, Kühnberger, and Schmid 2006; Schwering et al. 2009) and based on anti-unification) to give a computational account of how these processes can give rise to basic concepts of arithmetic (Guhe et al. 2011). Another set of important seed technologies for COINVENT originates in research carried out originally at the University of Bremen, and now at the University of Magdeburg, and addresses the knowledge representation and reasoning layer of the project. This includes the distributed ontology language DOL, currently standardised within the Object Management Group OMG (`www.ontoiop.org`), a major international effort with over 40 experts involved worldwide, and which supports an extensible number of logical languages, major modularisation and logical structuring techniques, and in particular supports the specification of basic blending diagrams as formalised by Joseph Goguen. Moreover, the Hets system[2] will serve as a central, and extensible, reasoning infrastructure, with which other tools developed within COINVENT will interface. Lastly, the technology developed in the `OntoHub.org` project will allow the building of a dedicated semantic repository for formalised concepts in the mathematics and music domains, supporting heterogeneous specifications in a semantically-backed logical context, and providing interfaces for sharing, browsing, and the integration of reasoning services. This repository will be hosted at `conceptportal.org`.

In addition, the consortium members of COINVENT have shown an important experience in the development and application of the above foundational theories and seed technologies to a wide variety of fields, in computational creativity and other related areas: by studying the combination of case solutions and knowledge transfer in case-based reasoning (CBR) (Ontañón and Plaza 2010; Ontañón and Plaza 2012) and its application to computational creativity (Ontañón and Plaza 2012; Arcos 2012); by providing formal foundations for distributed reasoning with heterogeneous logics and their representations (Mossakowski, Maeder, and Lüttich 2007), and by applying them to achieve semantic alignment and integration (Schorlemmer and Kalfoglou 2008; Kutz, Mossakowski, and Lücke 2010; Kalfoglou and Schorlemmer 2010; Kutz et al. 2012); by proposing novel architectures for coherence-driven, cognitively-inspired (BDI) agents (Joseph et al. 2010) and computational frameworks for multi-agent interaction-based agreement on concepts and their semantics (Ontañón and Plaza 2010; Atencia and Schorlemmer 2012); by formalising *Lakatos*-style automated theorem proving (Colton and Pease 2004) and mathematical theory formation (Colton 2002).

Expected Contributions

We expect that a mathematically precise theory, as the one we are proposing in the context of the COINVENT project, will lead to the following contributions:

Theory and Technology. Computational implementations of cognitive and psychological models serve, in general, two main purposes:

[2]See `http://www.informatik.uni-bremen.de/agbkb/forschung/formal_methods/CoFI/hets/index_e.htm`

1. *Computational implementations are tools for exploring implications of the ideas embedded in a particular model, beyond the limits of human thinking. Thus, they are vehicles of further scientific inquiry of the cognitive and psychological processes that the model seeks to describe.*

   In this sense, the formal model coming out of the COINVENT project, together with its computational realisation, will be an important tool for exploring the implications of Fauconnier and Turner's theory of conceptual blending for understanding creative thinking. One such implication is the role concept creation and invention plays in serendipitous reasoning, i.e., in recognising the value of newly invented concepts not only for better understanding a certain domain, but even for advancing the understanding of a previously unidentified problem that was initially not the concern of inquiry. If our model advances the understanding of implications such as how serendipity might work, cognitive science and psychology could take these results to explore serendipitous reasoning from a cognitive and psychological point of view. This alone would already be an important step forward in developing a science of creativity.

   By grounding our research on Goguen's proposal for a Unified Concept Theory, we will build upon the deep understanding gained by relating different approaches to the notion of concept invention, and do so on a firm mathematical foundation that is consequently of great help in providing precise descriptions of what can and should be implemented in a computational system.

2. *Computational implementations make a general model that is usually stated in abstract terms more concrete, facilitating the study of its formal and computational properties, and guiding the design and implementation of computer systems that attempt to display the cognitive capabilities captured in the model. Hence, they provide direct engineering advances.*

   We will demonstrate these advances through two prototype implementations of autonomous creative systems that display creative activity through the accomplishment of concept creation and invention in the domains of mathematics and music. Ideally, these systems will be developed with the following properties:

   - an ability to form abstractions over both semantic and syntactic aspects of a domain;
   - an ability to form new representations, by conceptual blending;
   - an ability to revise representations on the basis of new concrete information that fits badly with the current conceptualisation (using ideas from Lakatos); and
   - heuristically guided algorithms to solve problems, based on combinations of the above abilities.

   If our intuitions are right about the power of conceptual blending to boost the capabilities of autonomous creative systems and our project is successful, our contribution could go even beyond that direction, in developing novel ways to use methodologies from cognitive science in systems engineering, and vice versa.

Working Domains. In the domain of mathematics, we plan to build a computational system that aids mathematicians in by supporting their reasoning at a conceptual level and in their creative work, for example

- proposing potentially interesting novel definitions, theories, and conjectures that are motivated by conceptual (not only formal) reasons, and
- evaluating the potential of such ideas when proposed by the mathematician.

Not only mathematicians, but also others engaged in similar sorts of reasoning, when developing new concepts and theories, can benefit greatly from the processes of building new conceptualisations from combinations of existing conceptualisations and particular examples and counter-examples.

The particular system we propose as our proof-of-concept would be the first of its kind in mathematics, as it goes well beyond what proof assistants do. More importantly, if, as we intend, the system turns out to be judged by mathematicians attractive and even potentially useful in their work of conceptually advancing mathematics, this would open the door to something not seen before. The system resulting from this project will, therefore, be a showcase of how systems like proof assistant systems can be improved so that they are useful for mathematicians.

In the domain of music, we plan to build a pioneering computational system that aids musicians in composition, namely in melodic harmonisation, that allows exploration of novel uncharted conceptual territories, for example

- proposing new harmonic concepts emerging from learned harmonic spaces, examples and counter-examples;
- suggesting new harmonic conceptualisations emerging from combinations/blends of different harmonic spaces that give rise to potentially interesting new harmonies.

Computer-aided compositional systems are often 'accused' of merely replicating/mimicking given music styles and being confined to the initial musical space that has been explicitly modelled in the system. The creativity of such systems is considered rather limited as the system cannot supersede its built-in concepts and cannot generate new unforeseen concepts. The particular system we propose as our proof-of-concept would be the first of its kind that goes well beyond what current melodic harmonisation systems are capable of doing. It would open the way more generally to music/art creativity assistance tools that enable people to explore the borders of their artistic creativity by giving them new original ideas for further exploration.

Measures of Creativity. The computational creativity community needs concrete measures of evaluation to enable us to make objective, falsifiable claims about progress made from one version of a program to another, or for comparing and contrasting different software systems for the same creative task. There are currently three main models of evaluation (Ritchie 2007; Colton, Pease, and Charnley 2011; Jordanous 2011), but they are still rarely used, and there are problems with each. We will extend these measures: for instance, serendipity, which is an important aspect of human

creativity, currently does not feature in any of the evaluation models. We will formulate ways of evaluating this and other under-represented notions. We will also contribute to the methodology of computational creativity by applying all three models, as well the new measures we develop, to our system and to other creative systems. One of the best ways to evaluate and improve measures of creativity is to apply them in a reflective manner. We will furthermore evaluate each model of evaluation according to principles in the philosophy of science, and survey other experts for ease of use and adherence to intuitions about creativity.

Acknowledgements

The project COINVENT acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant number: 611553.

References

Aamodt, A., and Plaza, E. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications* 7(1):39–52.

Alexander, J. 2011. Mathematical blending. *Semiotica* 2011(187):1–48.

Anagnostopoulou, C., and Cambouropoulos, E. 2012. Semiotic analysis and computational modeling: Two case studies on works by Debussy and Xenakis. In Sheinberg, E., ed., *Music Semiotics: A Network of Significations*. Ashgate Publishing, Surrey, UK. 129–145.

Arcos, J. L. 2012. Music and similarity based reasoning. In *Soft Computing in Humanities and Social Sciences*, volume 273 of *Studies in Fuzziness and Soft Computing*. Springer. 467–478.

Atencia, M., and Schorlemmer, M. 2012. An interaction-based approach to semantic alignment. *Journal of Web Semantics* 12–13:131–147.

Baron, S. G., and Osherson, D. 2011. Evidence for conceptual combination in the left anterior temporal lobe. *Neuroimage* 55(4):1847–1852.

Barwise, J., and Seligman, J. 1997. *Information Flow: The Logic of Distributed Systems*, volume 44 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press.

Boden, M. A. 1990. *The Creative Mind: Myths and Mechanisms*. George Weidenfeld and Nicolson Ltd.

Boden, M. A. 2009. Computer models of creativity. *AI Magazine*.

Boulez, P., and Connes, A. 2011. La créativité en musique et en mathématique. A dialogue during the 3rd International Conference on Mathematics and Computation in Music (MCM 2011).

Colton, S., and Pease, A. 2004. Lakatos-style automated theorem modification. In *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004*.

Colton, S.; Pease, A.; and Charnley, J. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *2nd International Conference on Computational Creativity*.

Colton, S. 2002. *Automated Theory Formation in Pure Mathematics*. Distinguished Dissertations Series. Springer.

Conklin, D., and Anagnostopoulou, C. 2006. Segmental pattern discovery in music. *INFORMS Journal on Computing* 18(3):285–293.

Deliège, I., and Wiggins, G., eds. 2006. *Musical Creativity: Multidisciplinary Research in Theory and Practice*. Psychology Press.

Echard, W. 2006. 'Plays guitar without any hands': Musical movement and problems of immanence. In Gritten, A., and King, E., eds., *Music and Gesture*. Ashgate.

Falkenhainer, B.; Forbus, K. D.; and Gentner, D. 1989. The structure-mapping engine: Algorithm and examples. *Artificial Intelligence* 41(1):1–63.

Fauconnier, G., and Turner, M. 1998. Conceptual integration networks. *Cognitive Science* 22(2):133–187.

Fauconnier, G., and Turner, M. 2003. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. New York: Basic Books.

Fauconnier, G. 1994. *Mental Spaces*. Cambridge University Press.

Future and Emerging Technologies. 2011. Creativity and ICT. FET consultation workshop. Report, European Commission, Directorate-General Information Society and Media.

Ganter, B., and Wille, R. 1999. *Formal Concept Analysis*. Springer.

Gärdenfors, P. 2004. *Conceptual Spaces*. A Bradford Book.

Gentner, D.; J.Holvoak, K.; and Kokinov, B. N., eds. 2001. *The Analogical Mind*. MIT Press.

Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science* 7:155–170.

Gibbs, Jr., R. W. 2000. Making good psychology out of blending theory. *Cognitive Linguistics* 11(3–4):347–358.

Goguen, J., and Burstall, R. 1992. Institutions: Abstract model theory for specification and programming. *Journal of the ACM* 39(1):95–146.

Goguen, J. A., and Harrell, D. F. 2010. Style: A computational and conceptual blending-based approach. In Argamon, S.; Burns, K.; and Dubnov, S., eds., *The Structure of Style*. Springer. chapter 12, 291–316.

Goguen, J. 1991. A categorical manifesto. *Mathematical Structures in Computer Science* 1(49–67).

Goguen, J. 1997. Towards a social, ethical theory of information. In Bowker, G.; Gasser, L.; Star, L.; and Turner, W., eds., *Social Science Research, Technical Systems and Cooperative Work: Beyond the Great Devide*. Erlbaum. 27–56.

Goguen, J. 1999. An introduction to algebraic semiotics, with applications to user interface design. In Nehaniv, C. L., ed., *Computation for Metaphors, Analogy, and Agents*, volume 1562 of *Lecture Notes in Computer Science*. Springer. 242–291.

Goguen, J. 2004. Musical qualia, context, time, and emotion. *Journal of Consciousness Studies* 11(3/4):117–147.

Goguen, J. 2005a. What is a concept? In Dau, F.; Mugnier, M.-L.; and Stumme, G., eds., *Conceptual Structures: Common Semantics for Sharing Knowledge. 13th International Conference on Conceptual Structures, ICCS 2005, Kassel, Germany, July 17-22, 2005. Proceedings*, volume 3596 of *Lecture Notes in Artificial Intelligence*, 52–77. Springer.

Goguen, J. 2005b. Information integration in institutions. To appear in a memorial volume for Jon Barwise edited by L. Moss. Draft available at `http://www.cs.ucsd.edu/users/goguen/pps/ifi04.pdf`.

Goguen, J. 2006. Mathematical models of cognitive space and time. In Andler, D.; Ogawa, Y.; Okada, M.; and Watanabe, S., eds., *Reasoning and Cognition*, volume 2 of *Interdisciplinary Conference Series on Reasoning Studies*. Keio University Press.

Guhe, M.; Pease, A.; Smaill, A.; Martínez, M.; Schmidt, M.; Gust, H.; Kühnberger, K.-U.; and Krumnack, U. 2011. A computational account of conceptual blending in basic mathematics. *Cognitive Systems Research* 12(3–4):249–265.

Guhe, M.; Smaill, A.; and Pease, A. 2009. Using information flow for modelling mathematical metaphors. In Howes, A.; Peebles, D.; and Cooper, R. P., eds., *Proceedings of the 9th International Conference on Cognitive Modeling (ICCM 2009)*.

Gust, H.; Kühnberger, K.-U.; and Schmid, U. 2006. Metaphors and heuristic-driven theory projection (hdtp). *Theoretical Computer Science* 354(1):98–117.

Hersh, R. 2011. From counting to quaternions – the agonies and ecstasies of the student repeat those of d'Alembert and Hamilton. *Journal of Humanistic Mathematics* 1(1):65–93.

Johnson, M. 2007. *The Meaning of the Body*. The University of Chicago Press.

Jordanous, A. 2011. Evaluating evaluation: Assessing progress in computational creativity research. In *Proceedings of the Second International Conference on Computational Creativity (ICCCX) Mexico City, Mexico*.

Joseph, S.; Sierra, C.; Schorlemmer, M.; and Dellunde, P. 2010. Deductive coherence and norm adoption. *Logic Journal of the IGPL* 18(1):118–156.

Kalfoglou, Y., and Schorlemmer, M. 2010. The information-flow approach to ontology-based semantic integration. In Poli, R.; Healey, M.; and Kameas, A., eds., *Theory and Applications of Ontology: Computer Applicactions*. Springer. chapter 4, 101–114.

Kutz, O.; Mossakowski, T.; Hois, J.; Bhatt, M.; and Bateman, J. 2012. Ontology blending in DOL. In Besold, T. R.; Kühnberger, K.-U.; Schorlemmer, M.; and Smaill, A., eds., *Computational Creativity, Concept Invention and General Intelligence. 1st International Workshop. Montpellier, France, August 27, 2012*.

Kutz, O.; Mossakowski, T.; and Lücke, D. 2010. Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design. *Logica Universalis* 4(2):255–333.

Lakatos, I. 1976. *Proofs and Refutations*. Cambridge University Press.

Lakoff, G., and Johnson, M. 1980. *Metaphors We Live By*. University of Chicago Press.

Lakoff, G., and Johnson, M. 1999. *Philosophy in the Flesh*. Basic Books.

Lakoff, G., and Núñez, R. E. 2000. *Where Mathematics Comes From*. Basic Books.

Lawvere, F. W., and Shanuel, S. H. 1997. *Conceptual Mathematics*. Cambridge University Press.

Mac Lane, S. 1971. *Categories for the Working Mathematician*. Springer.

Maturana, H. R., and Varela, F. J. 1987. *The Tree of Knowledge: The Biological Roots of Human Understanding*. Shambhala.

Mazzola, G.; Park, J.; and Thalmann, F. 2011. *Musical Creativity: Strategies and Tools in Composition and Improvisation*. Computational Music Science. Springer.

Montano-Rivas, O.; McCasland, R.; Dixon, L.; and Bundy, A. 2012. Scheme-based theorem discovery and concept invention. *Expert Systems with Applications* 39:1637–1646.

Mossakowski, T.; Maeder, C.; and Lüttich, K. 2007. The Heterogeneous Tool Set. In Grumberg, O., and Huth, M., eds., *Tools and Algorithms for the Construction and Analysis of Systems. 13th International Conference, TACAS 2007, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2007 Braga, Portugal, March 24 - April 1, 2007. Proceedings*, volume 4424 of *Lecture Notes in Computer Science*, 519–522. Springer.

Old, L. J., and Priss, U. 2001. Metaphor and information flow. In *Proceedings of the 12th Midwest Artificial Intelligence and Cognitive Science Conference*, 99–104.

Ontañón, S., and Plaza, E. 2010. Amalgams: A formal approach for combining multiple case solutions. In *ICCBR'10: 18th International Conference on Case-Based Reasoning*, volume 6176 of *Lecture Notes in Artificial Intelligence*, 257–271. Springer.

Ontañón, S., and Plaza, E. 2012. Toward a knowledge transfer model of case-based inference. In *Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, 341–346. AAAI Press.

Pease, A.; Colton, S.; Ramezani, R.; Charnley, J.; and Reed, K. 2013. A discussion on serendipity in creative systems. In *Proceedings of the Fourth International Conference on Computational Creativity*.

Pereira, F. C. 2007. *Creativity and Artificial Intelligence*, volume 4 of *Applications of Cognitive Linguistics*. Mouton de Bruyter.

Plotkin, G. D. 1971. A further note on inductive generalization. *Machine Intelligence* 6:101–124.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.

Rosch, E. H. 1973. Natural categories. *Cognitive Psychology* 4(3):328–350.

Schorlemmer, M., and Kalfoglou, Y. 2008. Institutionalising ontology-based semantic integration. *Applied Ontology* 3(3):131–150.

Schwering, A.; Krumnack, U.; Kühnberger, K.-U.; and Gust, H. 2009. Syntactic principles of heuristic-driven theory projection. *Cognitive Systems Research* 10(3):251–269.

Sowa, J. F. 2000. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks/Cole.

Steedman, M. 1996. The blues and the abstract truth: Music and mental models. In Garnham, A., and Oakhill, J., eds., *Mental Models in Cognitive Science*. Erlbaum, Mahwah, NJ.

Sternberg, R. J., ed. 1999. *Handbook of Creativity*. Oxford University Press.

Thagard, P., and Stewart, T. C. 2011. The AHA! experience: Creativity through emergent binding in neural networks. *Cognitive Science* 35:1–33.

Thagard, P. 2000. *Coherence in Thought and Action*. Life and Mind: Philosophical Issues in Biology and Psychology. MIT Press.

Turner, M. 2012. Blending and conceptual integration. http://markturner.org/blending.html.

Varela, F. J.; Thompson, E. T.; and Rosch, E. 1992. *The Embodied Mind*. MIT Press.

Veale, T., and O'Donoghue, D. 2000. Computation and blending. *Cognitive Linguistics* 11(3/4):253–281.

von Helmholtz, H. 1885. *On the Sensations of Tone*. Longmans & Co.

Zbikowski, L. M. 2002. *Conceptualizing Music*. Oxford University Press.

Blending in the Hub
Towards a computational concept invention platform

Oliver Kutz and **Fabian Neuhaus** and **Till Mossakowski** and **Mihai Codescu**

Institute of Knowledge and Language Engineering, Otto-von-Guericke University of Magdeburg, Germany

Abstract

Conceptual blending has been employed very successfully to understand the process of concept invention, studied particularly within cognitive psychology and linguistics. However, despite this influential research, within computational creativity little effort has been devoted to fully formalise these ideas and to make them amenable to computational techniques. We here present the basic formalisation of conceptual blending, as sketched by the late Joseph Goguen, and show how the Distributed Ontology Language DOL can be used to declaratively specify blending diagrams. Moreover, we discuss in detail how the workflow and creative act of generating and evaluating a new, blended concept can be managed and computationally supported within Ontohub, a DOL-enabled theory repository with support for a large number of logical languages and formal linking constructs.

Concept Invention via Blending

In the general methodology of conceptual blending introduced by Fauconnier and Turner (2003), the blending of two thematically rather different *conceptual spaces* yields a new conceptual space with emergent structure, selectively combining parts of the given spaces whilst respecting common structural properties.[1] The 'imaginative' aspect of blending is summarised as follows in Turner (2007):

> [...] the two inputs have different (and often clashing) organising frames, and the blend has an organising frame that receives projections from each of those organising frames. The blend also has emergent structure on its own that cannot be found in any of the inputs. Sharp differences between the organising frames of the inputs offer the possibility of rich clashes. Far from blocking the construction of the network, such clashes offer challenges to the imagination. The resulting blends can turn out to be highly imaginative.

A classic example for this is the blending of the concepts *house* and *boat*, yielding as most straightforward blends the concepts of a *houseboat* and a *boathouse*, but also an *amphibious vehicle* (Goguen and Harrell, 2009).

In the almost unlimited space of possibilities for combining existing ontologies to create new ontologies with emergent structure, conceptual blending can be built on to provide a structural and logic-based approach to 'creative' ontological engineering. This endeavour primarily raises the following two challenges: (1) when combining the terminologies of two ontologies, the shared semantic structure is of particular importance to steer possible combinations. This shared semantic structure leads to the notion of base ontology, which is closely related to the notion of 'tertium comparationis' found in the classic rhetoric and poetic theories, but also in more recent cognitive theories of metaphor (see, e.g., Jaszczolt (2003)); (2) having established a shared semantic structure, there is typically still a huge number of possibilities that can capitalise on this information in the combination process: here, structural optimality principles as well as ontology evaluation techniques take on a central role in selecting interesting blends.

We believe that the principles governing ontological blending are quite distinct from the rather informal principles employed in blending phenomena in language or poetry, or the rather strict principles ruling blending in mathematics, in particular in the way formal inconsistencies are dealt with. For instance, whilst blending in poetry might be particularly inventive or imaginative when the structure of the basic categories found in the input spaces is almost completely ignored, and whilst the opposite, i.e., rather strict adherence to sort structure, is important in areas such as mathematics in order to generate meaningful blends[2], ontological blending is situated somewhere in the middle: re-arrangement and new combination of basic categories can be rather interesting, but has to be finely controlled through corresponding interfaces, often regulated by or related to choices found in foundational or upper ontologies.

[1] The usage of the term 'conceptual space' in blending theory is not to be confused with the usage established by Gärdenfors (2000).

[2] For instance when creating the theory of transfinite cardinals by blending the perfective aspect of counting up to any fixed finite number with the imperfective aspect of 'endless counting' (Núñez, 2005).

The core contributions of the paper can be summarised as follows.[3] We:

- sketch the logical analysis of conceptual blending in terms of blending diagrams and colimits, as originally proposed by Joseph Goguen, and give an abstract definition of ontological blendoids capturing the basic intuitions of conceptual blending in the ontological setting;
- provide a formal language for declaratively specifying blending diagrams by employing the OWL[4] fragment of the distributed ontology language DOL for blending. This provides a structured approach to ontology languages and combines the simplicity and good tool support for OWL with the more complex blending facilities of OBJ3 (Goguen and Malcolm, 1996) or Haskell (Kuhn, 2002);
- discuss the capabilities of the Ontohub/Hets ecosystem with regard to collaboratively managing, creating, and evaluating blended concepts and theories; this includes an investigation of the evaluation problem in blending, together with a discussion of structural optimality principles and current automated reasoning support.

We close with a detailed discussion of open problems and future work.

Blending Computationalised

Goguen has created the field of *algebraic semiotics* which logically formalises the structural aspects of semiotic signs, sign systems, and their mappings (Goguen, 1999). In Goguen and Harrell (2009), algebraic semiotics has been applied to user interface design and blending. Algebraic semiotics does not claim to provide a comprehensive formal theory of blending—indeed, Goguen and Harrell admit that many aspects of blending, in particular concerning the meaning of the involved notions, as well as the optimality principles for blending, cannot be captured formally. However, the structural aspects *can* be formalised and provide insights into the space of possible blends.

Goguen defines semiotic systems to be algebraic theories that can be formulated by using the algebraic specification language OBJ (Goguen and Malcolm, 1996). Moreover, a special case of a semiotic system is a *conceptual space*: it consists only of constants and relations, one sort, and axioms that define that certain relations hold on certain instances.

As we focus on standard ontology languages, namely OWL and first-order logic, we here replace the logical language OBJ. As structural aspects in the ontology language are necessary for blending, we augment these languages with structuring mechanisms known from algebraic specification theory (Kutz et al., 2008). This allows to translate most parts of Goguen's theory to these ontology languages. Goguen's main insight has been that semiotic systems and conceptual spaces can be related via *morphisms*, and that blending is comparable to *colimit computation*, a construction that abstracts the operation of disjoint unions modulo the identification of certain parts, explained in more detail below. In particular, the blending of two concepts is often a *pushout* (also called a *blendoid* in this context).

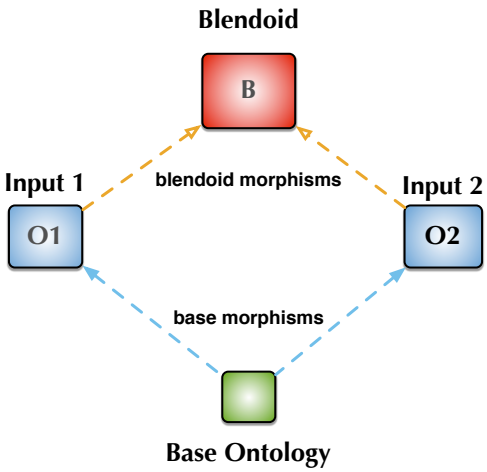


Figure 1: The basic integration network for blending: concepts in the base ontology are first refined to concepts in the input ontologies and then selectively blended into the blendoid.

Some basic definitions:[5] Non-logical symbols are grouped into **signatures**, which for our purposes can be regarded as collections of kinded symbols (e.g. concept names, relation names). **Signature morphisms** are maps between signatures that preserve (at least) kinds of symbols (i.e. map concept names to concept names, relations to relations, etc.). A **theory** or **ontology** pairs a signature with a set of sentences over that signature, and an **theory morphism** (or **interpretation**) between two theories is just a signature morphism between the underlying signatures that preserves logical consequence, that is, $\rho : T_1 \to T_2$ is a theory morphism if $T_2 \models \rho(T_1)$, i.e. all the translations of sentences of T_1 along ρ follow from T_2. This construction is completely logic independent.

Signature/theory morphisms are an essential ingredient for describing conceptual blending in a logical way.

We now give a general definition of ontological blending capturing the basic intuition that a blend of input ontologies shall partially preserve the structure imposed by base ontologies, but otherwise be an almost arbitrary extension or fragment of the disjoint union of the input ontologies with appropriately identified base space terms.

For the following definition, which we first introduced in Kutz et al. (2012), a diagram consists of a set of ontologies and a set of morphisms between them. The **colimit** of a diagram is similar to a disjoint union of its ontologies, with some identifications of shared parts as specified by the morphisms in the diagram. We refrain from presenting the category-theoretic definition here (which can be found in

[3] This paper elaborates on ideas first introduced in Hois et al. (2010); detailed technical definitions are given in Kutz et al. (2012).

[4] With 'OWL' we refer to OWL 2 DL, see `http://www.w3.org/TR/owl2-overview/`

[5] Note that these definitions apply to OWL, but also to many other logics. Indeed, they apply to any logic formalised as an *institution* (Goguen and Burstall, 1992).

Adámek, Herrlich, and Strecker (1990)), but explain the colimit operation using the examples below.

Definition 1 (Ontological Base Diagram) *An **ontological base diagram** is a diagram D for which the minimal nodes $(B_i)_{i \in D_{min} \subseteq |D|}$ are called **base ontologies**, the maximal nodes $(I_j)_{j \in D_{max} \subseteq |D|}$ called **input ontologies**, and where the theory morphisms $\mu_{ij} : B_i \to I_j$ are called the **base morphisms**. If there are exactly two inputs I_1, I_2, and one base $\mathcal{B}$, the diagram D is called **classical** and has the shape of a V. In this case, $\mathcal{B}$ is also called the **tertium comparationis**.*

Fig. 1 illustrates the basic, classical case of an ontological blending diagram. The lower part of the diagram shows the base space (tertium), i.e. the common generalisation of the two input spaces, which is connected to these via total (theory) morphisms, the base morphisms. The newly invented concept is at the top of this diagram, and is computed from the base diagram via a colimit. More precisely, any consistent subset of the colimit of the base diagram may be seen as a newly invented concept, a **blendoid** (a more precise definition of this notion is given in Kutz et al. (2012)). Note that, in general, ontological blending can deal with more than one base and two input ontologies.

Computing the Tertium Comparationis

To find candidates for base ontologies that could serve for the generation of ontological blendoids, much more shared semantic structure is required than the surface similarities that alignment approaches rely on. The common structural properties of the input ontologies that are encoded in the base ontology are typically of a more abstract nature. The standard example here relies on *image schemata*, such as the notion of a *container* (see e.g. Kuhn (2002)). Thus, in particular, foundational ontologies can support such selections. In analogical reasoning, 'structure' is (partially) mapped from a source domain to a target domain (Forbus, Falkenhainer, and Gentner, 1989; Schwering et al., 2009). Therefore, intuitively the operation of computing a base ontology can thus be seen as a bi-directional search for analogy or generalisation into a base ontology together with the corresponding mappings. Providing efficient means for finding a number of suitable such candidate generalisations is essential to making the entire blending process computationally feasible. Consider the example of blending 'house' with 'boat' discussed below in detail: even after fixing the base ontology itself, guessing the right mappings into the input ontologies means guessing within a space of approximately 1.4 Billion signature morphisms. Three promising candidates for finding generalisations are:

(1) **Ontology intersection:** Normann (2008) has studied the automatisation of theory interpretation search for formalised mathematics, implemented as part of the Heterogeneous Tool Set (HETS, see below). Kutz and Normann (2009) applied these ideas to ontologies by using the ontologies' axiomatisations for finding their shared structure. Accidental naming of concept and role names is deliberately ignored and such names are treated as arbitrary symbols (i.e., any concept may be matched with any other). By computing mutual theory interpretations between the inputs, the method allows to compute a base ontology as an *intersection* of the input ontologies together with corresponding theory morphisms. While this approach can be efficiently applied to ontologies with non-trivial axiomatisations, lightweight ontologies are less applicable, e.g., 'intersecting' a smaller taxonomy with a larger one clearly results in a huge number of possible taxonomy matches (Kutz and Normann, 2009). In this case, the following techniques are more appropriate.

(2) **Structure-based ontology matching:** matching and alignment approaches are often restricted to find simple correspondences between atomic entities of the ontology vocabulary. In contrast, work such as (Ritze et al., 2009; Walshe, 2012) focuses on defining a number of *complex correspondence patterns* that can be used together with standard alignments in order to relate complex expressions between two input ontologies. For instance, the 'Class by Attribute Type Pattern' may be employed to claim the equivalence of the atomic concept $\mathsf{PositiveReviewedPaper}$ in ontology O_1 with the complex concept $\exists \mathsf{hasEvaluation}.\mathsf{Positive}$ of O_2. Such an equivalence can be taken as an axiom of the base ontology; note, however, that it could typically not be found by intersecting the input ontologies. Giving such a library of design patterns may be seen as a variation of the idea of using image schemata.

(3) **Analogical Reasoning:** *Heuristic-driven theory projection* is a logic-based technique for analogical reasoning that can be employed for the task of computing a common generalisation of input theories. Schwering et al. (2009) establish an analogical relation between a source theory and a target theory (both first-order) by computing a common generalisation (called 'structural description'). They implement this by using anti-unification (Plotkin, 1970). A typical example is to find a generalisation (base ontology) formalising the structural commonalities between the Rutherford atomic model and a model of the solar system. This process may be assisted by a background knowledge base (in the ontological setting, a related domain or foundational ontology). Indeed, this idea has been further developed in Martinez et al. (2011).

Selecting the Blendoids: Optimality Principles

Having a common base ontology (computed or given), there is typically a large number of possible blendoids. For example, even in the rather simple case of combining House and Boat, allowing for blendoids which only partially maintain structure (called *non-primary* blendoids in Goguen and Harrell (2009)), i.e., where any subset of the axioms may be propagated to the resulting blendoid, the number of possible blendoids is in the magnitude of 1000. Clearly, from an ontological viewpoint, the overwhelming majority of these candidates is rather meaningless. A ranking therefore needs to be applied on the basis of specific ontological principles. In conceptual blending theory, a number of **optimality principles** are given in an informal and heuristic style (Fauconnier and Turner, 1998, 2003). While they provide useful guidelines for evaluating natural language blends, they do not suggest a direct algorithmic implementation, as also

analysed in Goguen and Harrell (2009). However, the importance of designing computational versions of optimality principles has been realised early on, and one such attempt may be found in the work of Pereira and Cardoso (2003), who proposed an implementation of the eight optimality principles presented in Fauconnier and Turner (1998) based on quantitative metrics for their more lightweight logical formalisation of blending. Such metrics, though, are not directly applicable to more expressive languages such as OWL or first-order logic. Moreover, the standard blending theory of Fauconnier and Turner (2003) does not assign types, which might make sense in the case of linguistic blends where type information is often ignored. A typical example of a type mismatch in language is the operation of *personification*, e.g., turning a boat into an 'inhabitant' of the 'boathouse'. However, in the case of blending in mathematics or ontology, this loss of information is often rather unacceptable: on contrary, a fine-grained control of type or sort information may be of the utmost importance.

Optimality principles for ontological blending are of two kinds.

(1) purely **structural/logical principles**: these extend and refine the criteria as given in Goguen and Harrell (2009), namely *degree of commutativity* of the blend diagram, *type casting* (preservation of taxonomical structure), *degree of partiality* (of signature morphisms), and *degree of axiom preservation*. In the context of OWL, typing needs to be replaced with preservation of specific axioms encoding the taxonomy.

(2) **heuristic principles**: these include introducing preference orders on morphisms (an idea that Goguen labelled $3/2$ pushouts (Goguen, 1999)) reflecting their 'quality' e.g. measured in terms of degree of type violation; specific ontological principles, e.g. adherence to the OntoClean methodology (Guarino and Welty, 2002), or general ontology evaluation techniques such as competency questions, further discussed below. Another set of heuristics is quantitative, statistical metrics, similar in style to those proposed in Pereira and Cardoso (2003).

The Distributed Ontology Language DOL

The distributed ontology language DOL is an ideal formal language for specifying both ontologies, base diagrams, and their blends. DOL is a metalanguage in the sense that it enables the reuse of existing ontologies (written in some ontology language like OWL or Common Logic) as building blocks for new ontologies and, further, allows to specify intended relationships between ontologies. One important feature of DOL is the ability to combine ontologies that are written in different languages without changing their semantics. DOL is going to be submitted as response to the Object Management Group's (OMG) Ontology, Model and Specification Integration and Interoperability (OntoIOp) Request For Proposal.[6]

[6] http://www.omg.org/cgi-bin/doc?ad/2013-12-02

In this section, we introduce DOL only informally. A formal specification of the language and its model theoretic semantics can be found in Mossakowski et al. (2013).

For the purpose of ontology blending the following features of DOL are relevant:

- **DOL library**. A DOL library consists of basic and structured ontologies and ontology interpretations. A basic ontology is an ontology written in some ontology language (e.g., OWL or Common Logic). A structured ontology builds on basic ontologies with the help of ontology translations, ontology unions, and symbol hiding.
- **ontology translation** (written O_1 **with** σ). A translation takes an ontology O_1 and a renaming function (technically, signature morphism) σ. The result of a translation is an ontology O_2, which differs from the ontology O_1 only by substituting the symbols as specified by the renaming function.
- **ontology union** (written O_1 **and** O_2). The union of two ontologies O_1 and O_2 is a new ontology O_3, which combines the axioms of both ontologies.
- **symbol hiding** (written O_1 **hide** $\{s_1, ..., s_n\}$). A symbol hiding takes an ontology O_1 and a set of symbols $s_1, ..., s_n$. The result of the hiding is a new ontology O_2, which is the result of 'removing' the symbols $s_1, ..., s_n$ from the signature of ontology O_1. Nevertheless, O_2 keeps all semantic constraints from O_1.[7]
- **ontology interpretation** (written **interpretation** $INT\_NAME : O_1$ to $O_2 = \sigma$). An ontology interpretation is a claim about the relationship between two ontologies O_1 and O_2, giving some renaming function σ. It states that all the constraints that are the result of translating O_1 with σ can be proven by O_2.

Some additional features that are necessary for blending will be introduced in the next section.

Formalising Blending in DOL

The novelty proposed by DOL is that the user can specify the base diagram of the blendoid. This is a crucial task, as the resulting blendoid depends on the dependencies between symbols that are stored in the diagram. Ontohub, our web platform and repository engine for managing distributed heterogeneous ontologies and discussed in more detail below, is able to use the specification of a base diagram to automatically generate the colimit-blendoid. In this section, we illustrate the specification of base diagrams in DOL and the resulting blendoids by blending house and boat to houseboat and boathouse.

The main inputs for the blendings consist of two ontologies, one for HOUSE and the other for BOAT. We adapted them from Goguen and Harrell (2009) but gave a stronger axiomatisation, making them more realistic. The purpose of this exercise is to show, using this classic blend, that our

[7] By approximation, one could consider O_2 as the ontology that is the result of existentially quantifying $s_1, ..., s_n$ in O_1.

framework allows to blend in a generic way complex ontological theories, thus not being restricted theoretically to any particular domain or even logical language.

Fig. 2 shows the ontology for HOUSE in OWL Manchester Syntax.[8]

```
Class: Artifact
Class: Capability
ObjectProperty: has_function
    Range: Capability
ObjectProperty: executes
    Range: Capability
ObjectProperty: is_located_on
Class: Person
Class: Plot
ObjectProperty: is_inhabited_by
    Domain: House
    Range:  Person
Class: ServeAsResidence
    SubClassOf: Capability
Class: ArtifactThatExecutesResidenceFunction
    EquivalentTo: Artifact that executes
                  some ServeAsResidence
    SubClassOf: is_inhabited_by some Person
Class: House
    SubClassOf: Artifact
        that is_located_on some Plot
        and has_function some
                    ServeAsResidence
```

Figure 2: Ontology House

As discussed above, finding candidate base ontologies and base morphisms is a non-trivial task. For the purpose of this example, we created them manually. The base ontologies are both quite simple, they mostly introduce shared concepts and contain only weak axiomatisations. The second base ontology only differs from the first by replacing the class Agent by Person and two additional classes, namely Object and Site.

```
ontology base1 =
    Class: Artifact  [...]  Class: Agent
     end

ontology base2 =
    Class: Artifact [...]  Class: Person
    Class: Object       Class: Site
    end
```

The blending of boat and house to boathouse is achieved by turning the boat into a habitat and moving the house from a plot of land to a body of water. This can be represented by two interpretations boat_habitable and house_floating.

[8] In the examples, note that concepts such as 'ArtifactThatExecutesResidenceFunction' are auxiliary symbols that are needed because of limitation of the Manchester Syntax being used, which does not allow to use complex concepts on the left-hand side of subsumption statements. The ontology for BOAT is axiomatized similarly, it can be found at http://www.ontohub.org/repositories/conceptportal.

```
interpretation boat_habitable : base2 to Boat =
    Object |-> Boat,
    Site |-> BodyOfWater

interpretation house_floating : base2 to House =
    Object |-> House,
    Site |-> Plot
```

The base ontologies and the interpretations above provide the necessary ingredients for a blending of BOAT and HOUSE to BOATHOUSE. The syntax of combinations is

combine $O_1, \ldots, O_m,\ M_1, \ldots, M_n$

where the O_i are ontologies, and M_i are morphism names. The semantics of combinations is the colimit of the generated diagram. A colimit involves both pasting together (technically: disjoint union) and identification of shared parts (technically: a quotient).

In our example, houseboat can be defined by the colimit based on the interpretations. To make the result easier to read, some of the classes are renamed:

```
ontology house_boat =
    combine boat_habitable, house_floating
    with Object |-> HouseBoat, Site |-> BodyOfWater
```

Ontohub is able to compute the colimit, which combines both the boat and house ontologies along the morphism. The colimit inherits most of the axioms of the ontologies and the base. Here we just show the declaration of the blended class Houseboat:

```
Class: HouseBoat
  SubClassOf: Artifact
    and has_function some MeansOfTransportation
    and has_function some Floating
    and is_navigated_by some Agent
  SubClassOf: Artifact
    and is_located_on some BodyOfWater
    and has_function some ServeAsResidence
```

In the case of blending of BOAT and HOUSE to BOATHOUSE, the crucial part in this blend is to view a boat as a kind of "person" that lives in a house. The two ontologies House and Boat presented above can be blended by selecting a base, which here provides (among others) a class Agent, and two interpretations, mapping Agent to Boat and Person, respectively. In this way, we let a boat play the role of a person (that inhabits a house).

```
interpretation boat_personification :
    base1 to Boat =
    Agent |-> Boat

interpretation house_import :
base1 to House =
    Agent |-> Person

ontology boat_house =
 combine boat_personification, house_import
 with Agent ↦ Boat, House ↦ BoatHouse
```

As before, Ontohub is able to compute the colimit. As above, we present here only the relevant declarations of the blended concept.

```
Class: BoatHouse
  SubClassOf: Artifact
    and is_located_on some Plot
    and has_function some ServeAsResidence
Class: ArtifactThatExecutesResidenceFunction
       EquivalentTo: Artifact
           and executes some ServeAsResidence
       SubClassOf: is_inhabited_by some Boat
```

Of course, the possibilities for blending the two concepts do not stop here. For example, we could map the `agent` in the base ontology to `person` in the boat ontology. This can be achieved by first defining an additional interpretation and by blending all three interpretations.

```
interpretation boat_import :
    base1 to Boat =
        Agent |-> Person

ontology boat_house =
        combine boat_personification, house_import,
boat_import
 with Agent ↦ Boat, House ↦ BoatHouse
```

The resulting blendoid is consistent, but it contains some strange consequences. For example, in the blendoid boats are driven by boats. However, if we are interested both in hosting boats and a hub for autonomous vehicles, this would count as an interesting result. In general, whether such more creative aspects of blendoids are desirable or not will depend on the context of the blending. We will address this issue in the section on evaluation below.

Blending in the Hub

Representation and Computation

Indeed, combinations and colimits can be computed by our web platform Ontohub. Ontohub is a repository engine for managing distributed heterogeneous ontologies. Ontohub supports a wide range of formal logical and ontology languages and allows for complex inter-theory (concept) mappings and relationships with formal semantics, as well as ontology alignments and blending. Ontohub understands various input languages, among them OWL and DOL.

We describe the basic design and features of Ontohub in general, and outline the extended feature-set that we pursue for conceptportal.org - a specialised repository within the distributed ontohub architecture.

The back-end of Ontohub is the Heterogeneous Tool Set HETS, which is used by Ontohub for parsing, static analysis and proof management of ontologies. HETS can also compute colimits of OWL diagrams and even approximations of colimits in the case where the input ontologies live in different ontology languages (Codescu and Mossakowski, 2008).

Computation of colimits in HETS is based on HETS' general colimit algorithm for diagrams of sets and functions (note that signatures in most cases are structured sets, and signature morphisms structure preserving functions). Such a colimit of sets and functions is computed by taking the disjoint union of all sets, and quotienting it by the equivalence relation generated by the diagram, which more precisely is obtained by the rule that given any element x of an involved set, any images of x under the involved functions are identified. The quotient is computed by selecting a representative of each equivalence class.

A difficulty that arises is that we have to make a choice of these representatives, and therefore of names for the symbols in the colimit, as a symbol may be not always identically mapped in the base diagram of the blendoid. The convention in HETS is that in case of ambiguity, the name of the symbol is chosen to be the most frequently occurring one. This gives the user control over the namespace, such that the symbols of the colimit can be later renamed. We can see this for our boathouse example above, where `Agent` appears most often in the diagram and therefore the symbol has been explicitly renamed.

Evaluating the Blending Space

Optimality principles, in particular structural ones, can be used to rank candidate blendoids on-the-fly during the ontology blending process. However, even if they improve on existing logical and heuristic methods, optimality principles will only narrow down the potential candidates and not tell us whether the result is a 'successful' blend of the ontologies. For example, assume that we had optimality principles that would show that from the roughly 1000 candidate blendoids of *House* and *Boat* that Goguen computed, only two candidates $\mathfrak{B}_{hb}$ and $\mathfrak{B}_{bh}$ are optimal. Is either $\mathfrak{B}_{hb}$ or $\mathfrak{B}_{bh}$ any good? And, if so, which of them should we use? To answer these question, it seems natural to apply ontology evaluation techniques.

Ontologies are human-intelligible and machine-interpretable representations of some portions and aspects of a domain that are used as part of information systems. To be more specific, ontology is a logical theory written in some knowledge representation language, which is associated with some intended interpretation. The intended interpretation is partially captured in the choice of symbols and natural language text (often in the form of annotations or comments). The evaluation of an ontology covers both the logical theory and the intended interpretation, their relationship to each other, and how they relate to the requirements that are derived from the intended use within a given information system. Therefore, ontology evaluation is concerned not only with formal properties of logical theories (e.g., logical consistency), but, among other aspects, with the *fidelity* of an ontology; that is whether the formal theory accurately represents the intended domain (Neuhaus et al., 2013). For example, if $\mathfrak{B}_{hb}$ is an excellent representation of the concept *houseboat*, then $\mathfrak{B}_{hb}$ provides a poor representation of the concept *boathouses*. Thus, any evaluation of the blend $\mathfrak{B}_{hb}$ depends on what domain $\mathfrak{B}_{hb}$ is intended to represent.

The lesson is that the evaluation of the results of ontology blending is dependent on the intended goal and, more generally, on the requirements that one expects the outcome of the blending process to meet. One way to capture these requirement is similar to competency questions, which are widely used in ontology engineering (Grüninger and Fox, 1995). Competency questions are usually initially captured in natural language, they specify examples for questions that

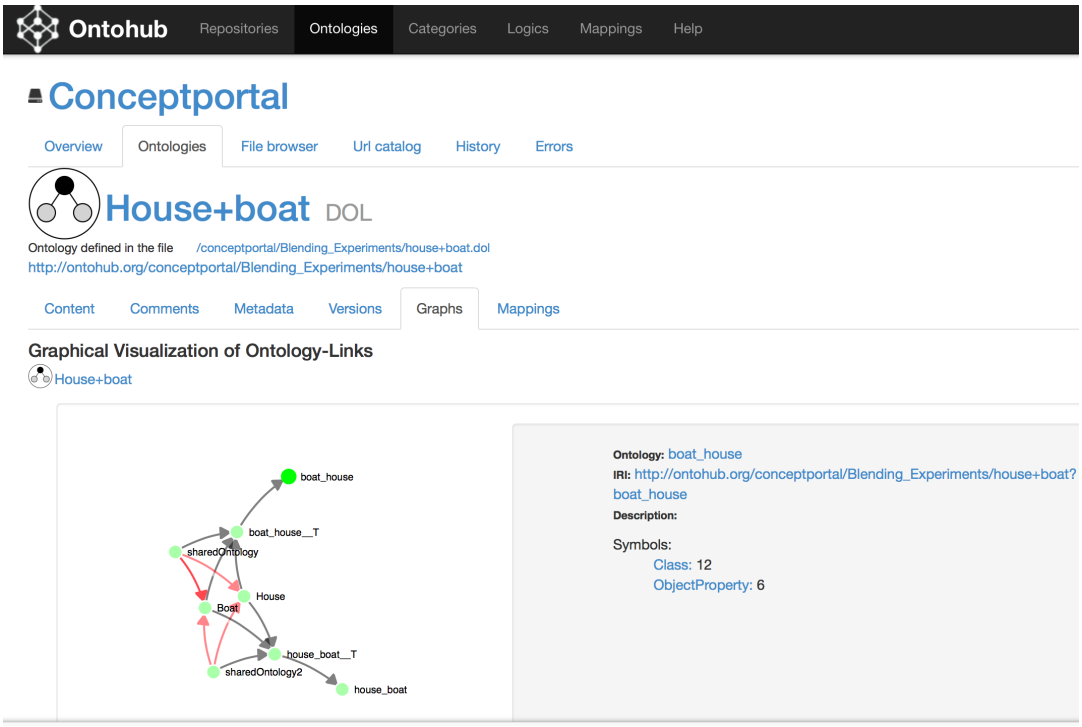


Figure 3: Blendoid representation and colimit computation via Hets/Ontohub: the screenshot of Ontohub shows the heterogeneous ontology house+boat.dol, hosted in the Conceptportal repository. The entire double-blend of house and boat into boathouse and houseboat is shown in the Graph to the left. The red arrows denote the interpretations of the shared ontologies into the blend. The concept boat_house is selected and shown on the right: its theory can be inspected by following the link to the respective ontology specification.

an ontology needs to be able to answer in a given scenario. By formalising the competency questions one can use automatic theorem provers to evaluate whether the ontology meets the intended interpretation.

The requirements that are used to select between the different blends fall, roughly, into two categories. ontological constraints and consequence requirements. Ontological constraints prevent the blends from becoming 'too creative' by narrowing the space for conceptual blending. E.g., it may be desirable to ensure that the `is_inhabited_by` relationship is asymmetric and that `is_navigated_by` is irreflexive. To achieve that any blendoid can be checked for logical consistency with the following ontology:

```
ontology OntologicalConstraints =
  ObjectProperty: is_inhabited_by
    Characteristics: Asymmetric
  ObjectProperty: is_navigated_by
    Characteristics: Irreflexive
```

Given these requirements, any blendoid that involves a house that lives in itself, or any boat navigated by itself (see the blendoid `boat_house1` above) would be discarded.

Consequence requirements specify the kind of characteristics the blendoid is supposed to have. E.g., assume the purpose of the conceptual blending is to find alternative housing arrangements, because high land prices make newly build houses unaffordable. In this case, the requirement could be 'a residence that is not located on a plot of land', which can be expressed in OWL as follows:

```
ontology ConsequenceRequirements =
[...]
  Class PlotFreeResidence
    EquivalentTo: Residence
      and (is_located_on only (not (Plot)))
```

Ontohub allows to use ontological constraints and consequence requirements to evaluate blended concepts automatically. The requirements are managed as DOL files, which allow to express that a given blendoid is logically consistent with a set of ontological constraints or that it entails some consequence requirements. The requirements themselves may be stored as regular ontology files (e.g., in OWL Manchester syntax). Ontohub executes the DOL files with the help of integrated automatic theorem provers, and is able to detect whether a blendoid meets the specified requirements.

At this time, the evaluation of blendoids for ontological constraints and consequence requirements depends on the use of DOL files. We are planning to integrate this functionality into the GUI of Ontohub to make it more convenient for the user.

Another way to evaluate a blendoid is to analyse its structure for typical ontological errors. For this purpose, Ontohub has integrated OOPS!. OOPS! automatically analy-

ses ontologies for common pitfalls, which is developed by the Ontology Engineering Group at the Technical University of Madrid (Poveda-Villalón, Suárez-Figueroa, and Gómez-Pérez, 2012). We are planning to add additional evaluation tools to Ontohub in the future.

Outlook

Our work in this paper follows a research line in which blending processes are primarily controlled through mappings and their properties (Gentner, 1983; Forbus, Falkenhainer, and Gentner, 1989; Veale, 1997; Pereira, 2007). By introducing blending techniques to ontology languages, we have provided a method which allows us to combine two thematically different ontologies into a newly created ontology, the blendoid, describing a novel concept or domain. The blendoid creatively mixes information from both input ontologies on the basis of structural commonalities of the inputs and combines their axiomatisations.

We have illustrated that the tool HETS and the DOL language (Mossakowski et al., 2013) provide an excellent starting point for developing the theory and practice of ontology blending further. They: (1) support various ontology language and their heterogeneous integration (Kutz et al., 2008); (2) allow to specify theory interpretations and other morphisms between ontologies (Kutz, Mossakowski, and Lücke, 2010); (3) support the computation of colimits as well as the approximation of colimits in the heterogeneous case (Codescu and Mossakowski, 2008); (4) provide (first) solutions for automatically computing a base ontology through ontology intersection (Kutz and Normann, 2009) and blendoid evaluation using requirements or tools such as OOPS!.

In particular, we have shown that the blending of ontologies can be declaratively encoded in a DOL ontology representing the respective blending diagram—here, employing the homogeneous fragment of DOL just using OWL ontologies. Blendoid ontologies, as well as their components, i.e. input and base ontologies, can be stored, formally related, and checked for consistency within Conceptportal, a repository node within Ontohub dedicated to blending experiments carried out in the European FP7 Project COINVENT. Ontohub moreover gives access to thousands of ontologies from a large number of different scientific and common sense domains. They are searchable via rich metadata annotation, logics used, formality level, and other dimensions, to provide not only a rich pool of ontologies for blending experiments, but also for the evaluation of newly created concepts. Ontohub also supports a growing set of collaborative features, including online editing of ontologies, commenting, version control, and group and permission management.

To make concept invention via ontological blending feasible in practice from within Ontohub, a number of further plugins into the architecture are planned covering in particular the automatic creation of base ontologies together with their mappings, the implementation of filtering blendoids by structural optimality principles and preference orders on morphisms, as well as the addition of more ontologically motivated evaluation techniques as discussed above.

Acknowledgements

The project COINVENT acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open Grant number: 611553.

References

Adámek, J.; Herrlich, H.; and Strecker, G. 1990. *Abstract and Concrete Categories*. Wiley, New York.

Codescu, M., and Mossakowski, T. 2008. Heterogeneous colimits. In Boulanger, F.; Gaston, C.; and Schobbens, P.-Y., eds., *MoVaH'08*.

Fauconnier, G., and Turner, M. 1998. Conceptual integration networks. *Cognitive Science* 22(2):133—187.

Fauconnier, G., and Turner, M. 2003. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books.

Forbus, K.; Falkenhainer, B.; and Gentner, D. 1989. The structure-mapping engine. *Artificial Intelligence* 41:1–63.

Gärdenfors, P. 2000. *Conceptual Spaces - The Geometry of Thought*. Bradford Books. MIT Press.

Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science* 7:155–170.

Goguen, J. A., and Burstall, R. M. 1992. Institutions: Abstract Model Theory for Specification and Programming. *Journal of the Association for Computing Machinery* 39(1):95–146. Predecessor in: LNCS 164, 221–256, 1984.

Goguen, J. A., and Harrell, D. F. 2009. Style: A Computational and Conceptual Blending-Based Approach. In *The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning*. Springer.

Goguen, J., and Malcolm, G. 1996. *Algebraic Semantics of Imperative Programs*. MIT Press.

Goguen, J. 1999. An introduction to algebraic semiotics, with applications to user interface design. In Nehaniv, C. L., ed., *Computation for Metaphors, Analogy, and Agents*, volume 1562 of *Lecture Notes in Computer Science*. Springer. 242–291.

Grüninger, M., and Fox, M. S. 1995. The role of competency questions in enterprise engineering. In *Benchmarking—Theory and Practice*. Springer. 22–31.

Guarino, N., and Welty, C. 2002. Evaluating ontological decisions with OntoClean. *Commun. ACM* 45(2):61–65.

Hois, J.; Kutz, O.; Mossakowski, T.; and Bateman, J. 2010. Towards Ontological Blending. In *Proc. of the The 14th International Conference on Artificial Intelligence: Methodology, Systems, Applications (AIMSA-2010)*.

Jaszczolt, K. M. 2003. On Translating 'What Is Said': *Tertium Comparationis* in Contrastive Semantics and Pragmatics. In *Meaning Through Language Contrast Vol. 2*. J. Benjamins. 441–462.

Kuhn, W. 2002. Modeling the Semantics of Geographic Categories through Conceptual Integration. In *Proc. of GIScience 2002*, 108–118. Springer.

Kutz, O., and Normann, I. 2009. Context Discovery via Theory Interpretation. In *Proc. of the IJCAI Workshop on Automated Reasoning about Context and Ontology Evolution, ARCOE-09, Pasadena, California*.

Kutz, O.; Lücke, D.; Mossakowski, T.; and Normann, I. 2008. The OWL in the CASL—Designing Ontologies Across Logics. In *Proc. of OWLED-08*, volume 432. CEUR.

Kutz, O.; Mossakowski, T.; Hois, J.; Bhatt, M.; and Bateman, J. 2012. Ontological Blending in DOL. In Besold, T.; Kuehnberger, K.-U.; Schorlemmer, M.; and Smaill, A., eds., *Computational Creativity, Concept Invention, and General Intelligence, Proc. of the 1st Int. Workshop C3GI@ECAI*, volume 01-2012. Montpellier, France: Publications of the Institute of Cognitive Science, Osnabrück.

Kutz, O.; Mossakowski, T.; and Lücke, D. 2010. Carnap, Goguen, and the Hyperontologies: Logical Pluralism and Heterogeneous Structuring in Ontology Design. *Logica Universalis* 4(2):255–333. Special Issue on 'Is Logic Universal?'.

Martinez, M.; Besold, T. R.; Abdel-Fattah, A.; Kühnberger, K.-U.; Gust, H.; Schmidt, M.; and Krumnack, U. 2011. Towards a Domain-Independent Computational Framework for Theory Blending. In *Proc. of the AAAI Fall 2011 Symposium on Advances in Cognitive Systems*.

Mossakowski, T.; Kutz, O.; Codescu, M.; and Lange, C. 2013. The Distributed Ontology, Modeling and Specification Language. In et al., C. D. V., ed., *Proceedings of the 7th International Workshop on Modular Ontologies (WoMO-13)*, volume 1081. CEUR-WS.

Neuhaus, F.; Vizedom, A.; Baclawski, K.; Bennett, M.; Dean, M.; Denny, M.; Grüninger, M.; Hashemi, A.; Longstreth, T.; Obrst, L.; et al. 2013. Towards ontology evaluation across the life cycle: The Ontology Summit 2013. *Applied Ontology* 8(3):179–194.

Normann, I. 2008. *Automated Theory Interpretation*. Ph.D. Dissertation, Department of Computer Science, Jacobs University, Bremen.

Núñez, R. E. 2005. Creating mathematical infinities: Metaphor, blending, and the beauty of transfinite cardinals. *Journal of Pragmatics* 37:1717–1741.

Pereira, F. C., and Cardoso, A. 2003. Optimality Principles for Conceptual Blending: A First Computational Approach. *AISB Journal* 1(4).

Pereira, F. C. 2007. *Creativity and Artificial Intelligence*, volume 4 of *Applications of Cognitive Linguistics*. Mouton de Bruyter.

Plotkin, G. D. 1970. A note on inductive generalization. *Machine Intelligence* 5:153–163.

Poveda-Villalón, M.; Suárez-Figueroa, M. C.; and Gómez-Pérez, A. 2012. Validating Ontologies with OOPS! In *Knowledge Engineering and Knowledge Management*. Springer. 267–281.

Ritze, D.; Meilicke, C.; Šváb Zamazal, O.; and Stuckenschmidt, H. 2009. A Pattern-based Ontology Matching Approach for Detecting Complex Correspondences. In *OM-09*, volume 551 of *CEUR*.

Schwering, A.; Krumnack, U.; Kühnberger, K.-U.; and Gust, H. 2009. Syntactic Principles of Heuristic-Driven Theory Projection. *Cognitive Systems Research* 10(3):251–269.

Turner, M. 2007. The Way We Imagine. In Roth, I., ed., *Imaginative Minds - Proc. of the British Academy*. Oxford: OUP. 213–236.

Veale, T. 1997. Creativity as pastiche: A computational treatment of metaphoric blends, with special reference to cinematic "borrowing". In *Proc. of Mind II: Computational Models of Creative Cognition*.

Walshe, B. 2012. Identifying complex semantic matches. In *The Semantic Web: Research and Applications*. Springer. 849–853.

Creativity in Conceptual Spaces

Antonio Chella[1] **Salvatore Gaglio**[1,3] **Gianluigi Oliveri**[2,3] **Agnese Augello**[3] **Giovanni Pilato**[3]

DICGIM (1)
University of Palermo
Viale Delle Scienze Ed. 6
Palermo, Italy
(antonio.chella,salvatore.gaglio)@unipa.it

Dipartimento di Scienze Umanistiche (2)
University of Palermo
Viale Delle Scienze Ed. 12
Palermo, Italy
gianluigi.oliveri@unipa.it

ICAR (3)
Italian National Research Council
Viale Delle Scienze Ed.11
Palermo, Italy
(augello,pilato)@pa.icar.cnr.it

Abstract

The main aim of this paper is contributing to what in the last few years has been known as computational creativity. This will be done by showing the relevance of a particular mathematical representation of Gärdenfors's conceptual spaces to the problem of modelling a phenomenon which plays a central role in producing novel and fruitful representations of perceptual patterns: analogy.

Introduction

There is an old tradition going back to Plato for which the phenomena which fall under the concept of creativity are those associated with the acquisition and mastery of some kind of craft (*techne*), rather than with random activity and aimless chance. According to this way of thinking, there is no reason to believe that an unschooled little ant that happens to draw in its course on the sand the first page of the score of the St. Matthew's Passion is engaged in a creative activity.

Indeed, for the supporters of this tradition, including the later Wittgenstein, creativity presupposes the existence of a high level linguistic competence typical of human beings. Here, of course, painting and music making — when seen as profoundly different from doodling or from casual humming —- are considered to be activities involving the use of some kind of articulated visual or auditory vehicles which give expression to feelings, emotions, etc., articulated visual or auditory vehicles which come with a syntax.

If we were successful in our attempt to model analogy within the particular mathematical representation of Gärdenfors's conceptual spaces we have chosen, this, besides scoring a point in favour of the computational creativity research programme (Cardoso, Veale, and Wiggins 2009), (Colton and Wiggins 2012), would also have important consequences with regard to the tenability of the old traditional view of creativity we mentioned above. For, since Gärdenfors's conceptual spaces, as we shall see in what follows, are placed in the sub-linguistic level of the cognitive architecture of a cognitive agent (CA), there would be at least a phenomenon intuitively belonging to creativity which could be represented independently of language.

After a section dedicated to a brief survey of some of the central contributions to the study of the connection between analogical thinking and computation, the paper proceeds to an explanation of how analogy is related to creativity. The article then develops by means of an illustration of the cognitive architecture of our CA in which the nature and function of Gärdenfors's conceptual spaces is made explicit.

A characterization of two conceptual spaces present in the 'library' of our CA — the visual and the music conceptual spaces — is then offered and visual analogues of music patterns are examined. The theoretical points made in the paper are, eventually, illustrated in the discussion of a case study.

Analogical thinking and computation

Human cognition is deeply involved with analogy-making processes. Analogical capabilities make possible perceiving clouds as resembling to animals, solving problems through the identification of similarities with previously solved problems, understanding metaphors, communicating emotions, learning, etc. (Kokinov and French 2006), (Holyoak et al. 2001).

Analogical reasoning is ordinarily used to 'transfer' structures, relational properties, etc. from a source domain to a target domain, and is clearly involved in that human ability which consists in producing generalizations.

Many models for analogical thinking are present in the literature. They are characterized by: (1) the ways of representing the knowledge on which the analogical capability is based, (2) the processes involved in realizing the analogical relation, and by (3) the manner in which the analogical transfer is fulfilled (Krumnack, Khnberger, and Besold 2013).

A known class of computational models for analogy-making are those based on Gentner's (1983) Structure Mapping Theory (SMT). This theory was the first that focussed on the role of the structural similarity existing between source and target domains, structural similarity which is generated by common systems of relations obtaining between objects in the respective domains. The structure mapping theory uses graphs to represent the domains and computes analogical relations by identifying maximal matching sub-graphs (Krumnack, Khnberger, and Besold 2013).

Other models are based on a connectionist approach, for example, we can mention here the Structured Tensor Analogical Reasoning (STAR) (Halford et al. 1994) and its 'evolution' STAR-2 (Wilson et al. 2001), which provide mechanisms for computing analogies using representations based

on the mathematics of tensor products (Holyoak et al. 2001); and the framework for Learning and Inference with Schemas and Analogies (LISA) (Hummel and Holyoak 1996) which exploits temporally synchronized activations in a neural network to identify a mapping between source and target elements.

In 1989 Keith Holyoak and Paul Thagard (Holyoak and Thagard 1989) proposed a theory of analogical mapping based upon interacting structural, semantic, and pragmatic constraints that have to be satisfied at the same time, implementing the theory as an emergent process of activation states of neuron-like objects.

According to (French 1995), metaphorical language, analogy making and couterfactualization are all products of the mind's ability to perform slippage (i.e. the replacement of one concept in the description of some situation by another related one) fluidly. All analogies involve some degree of conceptual slippage: under some pressure, concepts slip into related concepts. On the notion of conceptual slippage is based Copycat, a model of analogy making developed in 1988 by Douglas Hofstadter et al. (Hofstadter and Mitchell 1994).

In (Kazjon Grace and Saunders 2012), a computational model of associations, based on an interpretation-driven framework, was put forward and applied to the domain of real-world ornamental designs, where an association is understood in terms of the process of constructing new relationships between different ideas, objects or situations.

In (Grace, Saunders, and Gero 2008) a computational model for the creation of original associations has been presented. The approach is based on the concept of *interpretation*, which is defined as "a perspective on the meaning of an object; a particular way of looking at an object" [1] , and acts on conceptual spaces, where concepts are defined as regions in that space. In this context the authors represent the interpretation process as a transformation applied to the conceptual space from which feature-based representations are generated. The model tries to identify relationships that can be built between a *source* object and a *target* object. A new association is constructed when the transformations applied to these objects contribute to the emergence of some shared features which were not present before the application of the transformations.

Creativity and Analogy

It is intuitively correct to say that the use of a stick made by a bird to catch a larva in the bark of a tree is creative, as it is creative the writing of a poem or the introduction of a new mathematical concept. Creativity, indeed, covers a large variety of phenomena which also differ from one another in relation to their different degree of abstractness, i.e., the creativity of the hunting technique of the bird is much less abstract than that displayed by Beethoven in the writing of the fifth symphony.

It is not our intention in this paper even to attempt to give a definition of creativity. What we want to do here is simply to focus on the concept of analogy — the relation in which A is to B is the same as the relation in which α is to β — which is at the heart of much of what we can correctly describe as creative activity.

A traditional model of analogical thinking is provided by the concept of proportion:

$$\frac{A}{B} = \frac{\alpha}{\beta}$$

where A and B are entities homogeneous to each other — like α and β are homogeneous to each other — but A and B are non-homogeneous to α and β. Analogical thinking allows the emergence/recognition of a pattern in a certain environment E which is similar/the same as that which has already emerged/been recognized in another environment E'. Much of the work to be done in what follows will consist in rendering mathematically rigorous what we have called 'pattern', 'environment E', 'analogy as similarity of patterns given in different environments', 'identity of patterns given in different environments', etc. etc.

Let us say that patterns are here understood as relational entities (structures) defined on a given domain.[2] And since a necessary condition for the emergence/recognition of a pattern is the presence of a system of representation, we are going to identify the environment E with such a system, and choose as a model of such a system of representation Gärdenfors's conceptual spaces. Moreover, two patterns π_1 and π_2 given in two different conceptual spaces V_1 and V_2 are said to be 'analogous to one another' if there is an homomorphism between π_1 and π_2, whereas they are said to be 'exemplifying the same pattern' if there is an isomorphism between π_1 and π_2.

A cognitive architecture based on Conceptual Spaces

The introduction of a cognitive architecture for an artificial agent implies the definition of a conceptual representation model.

Conceptual spaces (CS), employed extensively in the last few years (Chella, Frixione, and Gaglio 1997) (De Paola et al. 2009) (Jung, Menon, and Arkin 2011), were originally introduced by Gärdenfors as a bridge between symbolic and connectionist models of information representation. This was part of an attempt to describe what he calls the 'geometry of thought'.

In (Gärdenfors 2000) and (Gärdenfors 2004) we find a description of a cognitive architecture for modelling representations. This is a cognitive architecture in which an intermediate level, called 'geometric conceptual space', is introduced between a linguistic-symbolic level and an associationist sub-symbolic level of information representation.

The cognitive architecture (see figure 1), is composed by three levels of representation: a *subconceptual level*, in which data coming from the environment are processed by means of a neural networks based system, a *conceptual level*, where data are represented and conceptualized independently of language; and, finally, a *symbolic level* which

[1] (Grace, Saunders, and Gero 2008), Section 2, page 2

[2] For the special case represented by mathematical patterns see (Oliveri 1997), (Oliveri 2007), ch. 5, and (Oliveri 2012).

makes it possible to manage the information produced at the conceptual level at a higher level through symbolic computations. The conceptual space acts as a workspace in which low-level and high-level processes access and exchange information respectively from bottom to top and from top to bottom. The description of the symbolic and subconceptual levels goes beyond the scope of this paper.

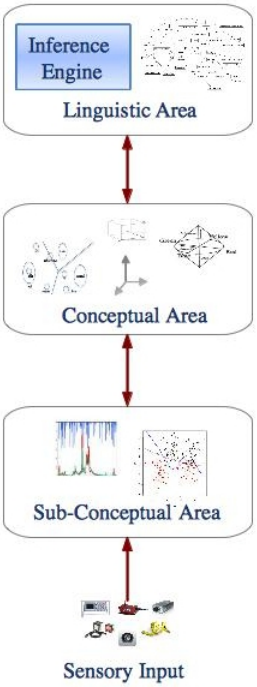


Figure 1: A sketch of the cognitive architecture

According to the linguistic/symbolic level:

> "Cognition is seen as essentially being *computation*, involving symbol manipulation (Gärdenfors 2000)".

whereas, for the associationist sub-symbolic level:

> "Associations among different kinds of information elements carry the main burden of representation. *Connectionism* is a special case of associationism that models associations using artificial neuron networks (Gärdenfors 2000), where the behaviour of the network as a whole is determined by the initial state of activation and the connections between the units (Gärdenfors 2000)".

Although the symbolic approach allows very rich and expressive representations, it appears to have some intrinsic limitations such as the so-called "symbol grounding problem", [3] and the well known A.I. "frame problem".[4] On the other hand, the associationist approach suffers from its low-level nature, which makes it unsuited for complex tasks, and representations.

Gärdenfors' proposal of a third way of representing information exploits geometrical structures rather than symbols or connections between neurons. This geometrical representation is based on a number of what Gärdenfors calls 'quality dimensions' whose main function is to represent different qualities of objects such as brightness, temperature, height, width, depth.

Moreover, for Gärdenfors, judgments of similarity play a crucial role in cognitive processes. And, according to him, it is possible to associate the concept of distance to many kinds of quality dimensions. This idea naturally leads to the conjecture that the smaller is the distance between the representations of two given objects the more similar to each other the objects represented are.

According to Gärdenfors, objects can be represented as points in a conceptual space, knoxels (Gaglio et al. 1988) [5], and concepts as regions within a conceptual space. These regions may have various shapes, although to some concepts — those which refer to natural kinds or natural properties — correspond regions which are characterized by convexity.[6]

For Gärdenfors, this latter type of region is strictly related to the notion of prototype, i.e., to those entities that may be regarded as the archetypal representatives of a given category of objects (the centroids of the convex regions).

One of the most serious problems connected with Gärdenfors' conceptual spaces is that these have, for him, a phenomenological connotation. In other words, if, for example, we take, the conceptual space of colours this, according to Gärdenfors, must be able to represent the geometry of colour concepts in relation to how colours are given to us.

However, we have chosen a non phenomenological approach to conceptual spaces in which we substitute the expression 'measurement' for the expression 'perception', and consider a cognitive agent which interacts with the environment by means of the measurements taken by its sensors rather than a human being.

Of course, we are aware of the controversial nature of our non phenomenological approach to conceptual spaces. But, since our main task in this paper is characterizing a rational agent with the view of providing a model for artificial agents, it follows that our non-phenomenological approach to conceptual spaces is justified independently of our opinions on perceptions and their possible representations within conceptual spaces

Although the cognitive agent we have in mind is not a human being, the idea of simulating perception by means of measurement is not so far removed from biology. To see this,

[3] How to specify the meaning of symbols without an infinite regress deriving from the impossibility for formal systems to capture their semantics. See (Harnad 1990).

[4] Having to give a complete description of even a simple robot's world using axioms and rules to describe the result of different actions and their consequences leads to the "combinatorial explosion" of the number of necessary axioms.

[5] The term 'knoxel' originates from (Gaglio et al. 1988) by the analogy with "pixel". A knoxel k is a point in Conceptual Space and it represents the epistemologically primitive element at the considered level of analysis.

[6] A set S is *convex* if and only if whenever $a, b \in S$ and c is between a and b then $c \in S$.

consider that human beings, and other animals, to survive need to have a fairly good ability to estimate distance. The frog unable to determine whether a fly is 'within reach' or not is, probably, not going to live a long and happy life.

Our CA is provided with sensors which are capable, within a certain interval of intensities, of registering different intensities of stimulation. For example, let us assume that CA has a visual perception of a green object *h*. If CA makes of the measure of the colour of *h* its present stereotype of green then it can, by means of a comparison of different measurements, introduce an ordering of gradations of green with respect to the stereotype; and, of course, it can also distinguish the colour of the stereotype from the colour of other red, blue, yellow, etc. objects. In other words, in this way CA is able to introduce a 'green dimension' into its colour space, a dimension within which the measure of the colour of the stereotype can be taken to perform the role of 0.

The formal model of a conceptual space that at this point immediately springs to mind is that of a metric space, i.e., it is that of a set X endowed with a metric. However, since the metric space X which is the candidate for being a model of a conceptual space has dimensions, dimensions the elements of which are associated with coordinates which are the outcomes of (possible) measurements made by CA, perhaps a better model of a conceptual space might be an n-dimensional vector space V over a field K like, for example, $\mathbb{R}^n$ (with the usual inner product and norm) on $\mathbb{R}$.

Although this suggestion is interesting, we cannot help noticing that an important disanalogy between an n-dimensional vector space V over a field K, and the 'biological conceptual space' that V is supposed to model is that human, animal, and artificial sensors are strongly non-linear. In spite of its cogency, at this stage we are not going to dwell on this difficulty, because: (1) we intend to examine the 'ideal' case first; and because (2) we hypothesize that it is always possible to map a perceptual space into a conceptual space where linearity is preserved either by performing, for example, a small-signal approach, or by means of a projection onto a linear space, as it is performed in kernel systems (Scholkopf and Smola 2001).

The Music and Visual Conceptual Spaces

Let us consider a CA which can perceive both musical tones and visual scenes. The CA is able to build two types of conceptual spaces in order to represent its perceptions. As reported in (Augello et al. 2013a) (Augello et al. 2013b), the agent's conceptual spaces are generated by measurement processes; in this manner each knoxel is, directly or indirectly, related to measurements obtained from different sensors. Each knoxel is, therefore, represented as a vector $k = (x_1, x_2, ..., x_n)$ where x_i belongs to the X_i quality dimension of our n-dimensional vector space. The Conceptual Spaces can also be manipulated according to changes of the focus of attention of the agent (Augello et al. 2013a) (Augello et al. 2013b), however the description of this process goes beyond the scope of this paper and will not be described here.

Visual conceptual space

According to Biederman's geons theory (see (Biederman 1987)), the visual perception of an object is processed by our brain as a proper composition of simple solids of different shapes (the geons). Following Biederman main ideas, we exploit a conceptual space for the description of visual scenarios (see fig. 2) where objects are represented as compositions of super-quadrics, and super-quadrics are vectors in this conceptual space.

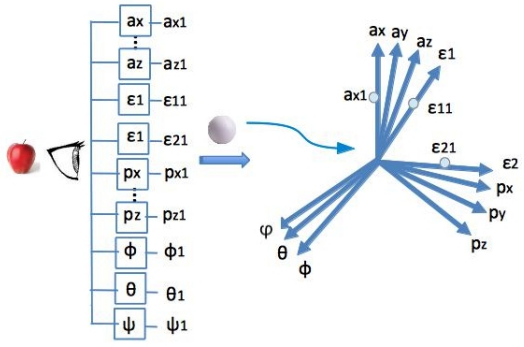


Figure 2: Visual perception and corresponding CS representation

For those who are not familiar with the concept of superquadric, let us say that super-quadrics are geometric shapes derived from the quadrics parametric equation with the trigonometric functions raised to two real exponents. The inside/outside function of the superquadric in implicit form is:

$$F(x,y,z) = \left[\left(\frac{x}{a_x}\right)^{\frac{2}{\epsilon_1}} + \left(\frac{y}{a_y}\right)^{\frac{2}{\epsilon_2}}\right]^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{a_z}\right)^{\frac{2}{\epsilon_1}}$$

where the parameters a_x, a_y, a_z are the lengths of the superquadric axes, the exponents $\varepsilon_1, \varepsilon_2$, called 'form factors', are responsible for the shapes form: values approaching 1 render the shape rounded.

To see this, let us suppose that the vision system can be approximated and modeled as a set of receptors, and that these receptors give as output, corresponding to the external perceived stimulation, the set of super-quadrics parameters associated to the perceived object. This leads to a superquadric conceptual representation of a 3D world. The situation is illustrated in Fig 2 where an object positioned in the 3D space, let us say an apple, is approximately perceived as a sphere and is consequently mapped as a knoxel in the related conceptual space.

In particular a knoxel in the Visual Conceptual space can be described by the vector:

$$\overrightarrow{k} = (a_x, a_y, a_z, \varepsilon_1, \varepsilon_2, p_x, p_y, p_z, \varphi, \theta, \psi)^T$$

In this perspective, knoxels correspond to simple geometric building blocks, while complex objects or situations are represented as suitable sets of knoxels (see figure 3).

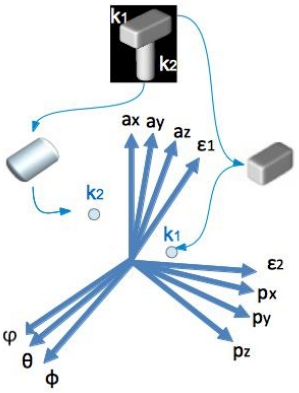


Figure 3: A representation of a hammer in the *visual conceptual space* as a composition of two super-quadrics

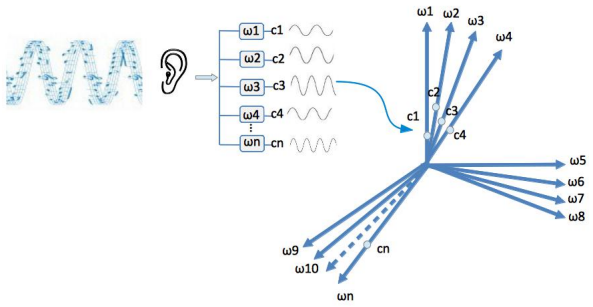


Figure 4: Music perception and corresponding CS representation

Music Conceptual Space

In (Gärdenfors 1988), Gardenfors discusses a program for musical spaces analysis directly inspired to the framework of vision proposed by Marr (Marr 1982). This discussion has been further analysed by Chella in (Chella 2013), where a music conceptual space has been proposed and placed into the layers of the cognitive architecture described in the previous sections.

As reported in (Shepard 1982), it has been highlighted that for the music of all human cultures, the relation between pitch and time appears to be crucial for the recognition of a familiar piece of music. In consideration of this, the representation of pitch becomes prominent for the representation of tones.

In the music CS the quality dimensions represent information about the partials composing musical tones. This choice is inspired by empirical results about the perception of tones to be found in (Oxenham 2013).

We model the functions of the ear as a finite set of filters, each one centred on the i-th frequency (we suppose for example to have N filters ranging from 20Hz to 20KHz at proper intervals. In this manner, a perceived sound will be decomposed into its partials and mapped as a vector $V = (c_1, c_2, \cdots, c_n)$ whose components correspond to the coefficients of the n frequencies that compose the sound $(\omega_1, \omega_2, \cdots \omega_n)$, as illustrated in Fig 4. The supposition is that here we use the discrete Fourier Series Transform, which is commonly used in signal processing, considering not only music but also other time-variant signals such as speech.

The vector $\overrightarrow{V}$ is, therefore, a knoxel of the music conceptual space. The partials of a tone are related both to the pitch and the timbre of the perceived note. Roughly, the fundamental frequency is related to the pitch, while the amplitudes of the remaining partials are also related to the timbre of the note. A similar choice is to be found in Tanguiane (Tanguiane 1993).

A knoxel in the music CS will change its position when the perceived tone changes its pitch or its loudness or timbre. In fig. 5 it is shown how the *symbolic level* given by the pentagram representation of a chord is mapped into a *conceptual* space representation.

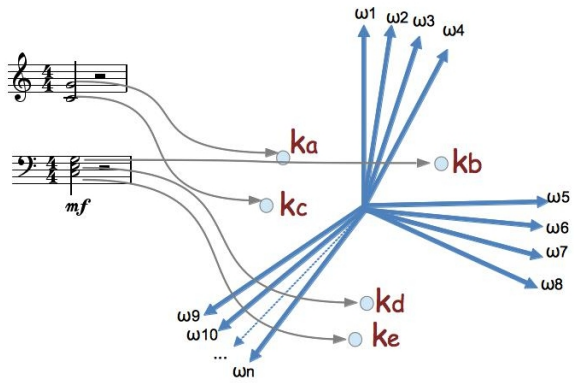


Figure 5: A representation of two chords in the *music conceptual space*.

From Visual Patterns to Music Patterns

A cognitive agent is able to represent its different perceptions in proper conceptual spaces; as soon as the agent perceives visual scenes or music, a given geometric structure will emerge. This structure will be made of vectors and regions, conceptual representations of perceived objects.

Music and visual conceptual spaces are two examples of conceptual representations that can be thought as a basis for computational simulation of an analogical thinking that provides the agent with some sort of creative capability. Knowledge and experiences made in a very specific domain of perception can be exploited by the agent in order to

better understand or to express in different ways the experiences and the perceptions that belong to other domains. This process resembles the synaesthesia [7] affecting some people, which allows to perform analogies between elements and experiences belonging to different sensory areas. Analogical thinking reveals similarities between patterns belonging to different domains.

For what concerns the music and vision domains, several analogies have been discussed in the literature. As an example, Tanguiane (Tanguiane 1993) compares visual and music perceptions, considering three different levels and both static and dynamic point of views. In particular, from a static point of view, a first visual level, that is the pixel perception level, can correspond the perception of *partials* in music. At the second level, the perception of simple patterns in vision corresponds to the perception of single *notes*. Finally at the third level, the perception of structured patterns (as patterns of patterns), corresponds to the perception of *chords*.

Concerning dynamic perception, the first level is the same as in the case of static perception, while at the second level the perception of visual objects corresponds to the perception of musical notes, and at the third final level the perception of visual trajectories corresponds to the perception of music *melodies*.

Gärderfors (Gärdenfors 1988), in his paper on "Semantics, Conceptual Spaces and Music" discusses a program for musical spaces analysis directly inspired to the framework of vision proposed by Marr (Marr 1982), where the first level is related to *pitch identification*; the second level is related to the identification of *musical intervals* and the third level to *tonality*, where scales are identified and the concepts of chromaticity and modulation arise. The fourth level of analysis is that at which the interplay of pitch and time is represented.

In what follows we are going to illustrate a framework for possible relationships between visual and musical domains. The mapping is one among many possible, and it has been chosen in order to make clear and easily understandable the whole process. As we have already said, it is possible to represent complex objects in a conceptual space as a set of knoxels. In particular, in the visual conceptual space, a complex object can be described as the set of knoxels representing the simple shapes of which it is composed, whereas in the music conceptual space we have seen how to represent chords as the set of knoxels representing the different tones played together.

In the two spaces will emerge recurrent patterns, given respectively by proper configurations of shapes and tones which occur more frequently. A fundamental analogy between the two domains can be highlighted, concerning the importance of the mutual relationships between the parts composing a complex object. In fact, in the case of perception of complex objects in vision, their mutual positions and shapes are important in order to describe the perceived object: e.g., in the case of an hammer, the mutual positions and the mutual shapes of the handle and the head are obviously important to classify the complex object as an hammer. A the same time, the mutual relationships between the pitches (and the timbres) of the perceived tones are important in order to describe the perceived chord (to distinguish for example, a major from a minor chord of the same note). Therefore, spatial relationships in static scenes analysis are in some sense analogous to sounds relationships in music conceptual space.

Although in this work we are overlooking the dynamic aspect of perception in the two domains of analysis, we can also mention some possible analogies, for example, we could correlate the trajectory of a moving object with a succession of different notes within a melody.

As certain movements are harmonious or not, so in music the succession of certain tones creates pleasant feelings or not.

Visual representation of musical objects: a case study

In what follows, we describe a procedure capable of simulating some aspects of analogical thinking. In particular, we consider an agent able to: (1) represent tones and visual objects within two different conceptual spaces; and (2) build analogies between auditory perceptions and visual perceptions.

At the heart of this procedure there is the ability on the part of the CA of individuating the appropriate homomorphism $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ which maps a knoxel belonging to a n-dimensional conceptual space $\mathbb{R}^n$ — the acoustic domain — on to another knoxel in a different m-dimensional conceptual space $\mathbb{R}^m$ — the visual domain.

For the sake of clarity we simplify the previously illustrated model of both music and visual conceptual representation of the agent. In particular:

- for what concerns the visual perceptions, we consider only a visual coding of spheres: this leads to the assumption that every observed object will be perceived as a sphere or as a composition of spheres by the agent;
- for what concerns the auditory perceptions, we consider only a limited set of discrete frequencies which the agent perceives. All information about pitch, loudness and timbre is implicitly represented in the auditory conceptual space by the Fourier Analysis parameters.

Figure 6 illustrates the mapping process leading from sensing and representation in the music conceptual spaces to a pictorial representation of the heard tone. The mapping is realized through an analogy transformation which let arise a visual knoxel in he visual conceptual space. The analogy process of the agent can be outlined in the following steps:

- the agent perceives a sound (A)
- the sound is sensed and decomposed through Fourier Transform Analysis (A)
- the measurements on the partials lead to a conceptual representation of the perceived sound as a knoxel in the acoustic space (A)

[7] a condition in which the stimulation of one sense causes the automatic experience of another sense

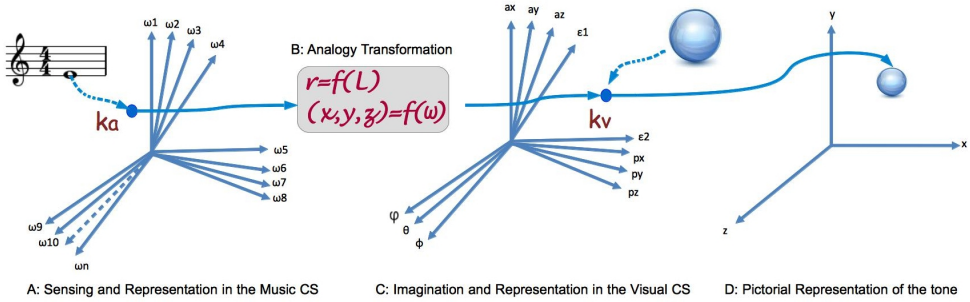


Figure 6: Mapping process leading from sensing and representation in the music conceptual spaces to a pictorial representation of the heard tone

- the knoxel k_A in the acoustic space is transformed into a knoxel k_V in the visual conceptual space (B)
- the mapping lets arise a conceptual representation of an object that is not actually perceived. It is only "imagined" by analogy. (C)
- the "birth" of this new item in the visual conceptual space, is directly related to the "birth" of an image, which, most importantly, is simply imagined and not perceived (D)

Given two conceptual spaces $\mathbb{R}^n$ and $\mathbb{R}^m$, the mapping can be any multidimensional function that realizes the appropriate transformation $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. The function f can be learned in a supervised or unsupervised way through machine learning algorithms.

At present, we superimpose the structure f. In order to make a choice for f we take some inspirations from Shepard in (Shepard 1982).

Many geometrical mappings have been proposed for pitch: the simplest one is that one which use of a monodimensional logarithmic scale where each pure tone is related to the logarithm of its frequency.

However, according to the two component theory (Révész 1954) (Shepard 1982), the best manner to pictorially represent pitches is a helix or 3D-spiral instead of a straight line. A mapping based on this theory is illustrated in fig. 7, where simple sounds are drawn on the helix, as spheres of different sizes, according to their associated loudness.

That mapping allows to complete one turn per octave and reaches the necessary geometric proximity between points which are an octave distant from each other.

The strong point of the uniform helix representation is that the distance corresponding to any particular musical interval is invariant throughout the representation. Each tone can be mapped onto a spiral laying on a cylinder where points vertically aligned correspond to the same tone with different octave. This projective property holds regardless of the slope of the helix (Shepard 1982).

In superimposing f we suppose that when the agent perceives a sound which is louder than another one, this evokes in his mind the view of something that is more cumbersome than another one. We assume that this perceived object has no preferred direction or shape, therefore the easiest way to represent it is a sphere, whose radius can be associated to the loudness of the perceived sound.

The other parameter is the pitch. As soon as the agent perceives different pitches, he tries to visualize them, imagine them, locate them according to the helix whose equations are:

$$x = rcos(2\pi\omega) \quad (1)$$
$$y = rsin(2\pi\omega) \quad (2)$$
$$z = c\omega \quad (3)$$

If we consider a simple tone of given frequency ω , the pitch will be represented by a point $p(x, y, z)$ in the spiral, while its loudness L will be represented by a sphere having centre in $p(x, y, z)$ and a radius whose length r is related to the perceived loudness.

The sphere corresponds to a knoxel in the Visual-conceptual-space, while the perceived tone corresponds to a knoxel in the Music-conceptual-space.

The agent therefore will visually imagine the perceived sound as a sphere whose radius is proportional to the perceived loudness, while its position corresponds to a point laying on the helical line representing all the tones that can be perceived by the agent, and a chord will be imagined as a set of spheres in this 3D space.

Conclusions

We have illustrated a methodology for the computational emulation of analogy, which is an important part of the imaginative process characterizing the creative capabilities of human beings.

The approach is based on a mapping between geometric conceptual representations which are related to the perceptive capabilities of an agent.

Even though this mapping can be built up in several different ways, we presented a proof-of-concept example of some analogies between music and visual perceptions. This allows the agent to associate imagined, unseen images to perceived sounds. It is worthwhile to point out that, in similar

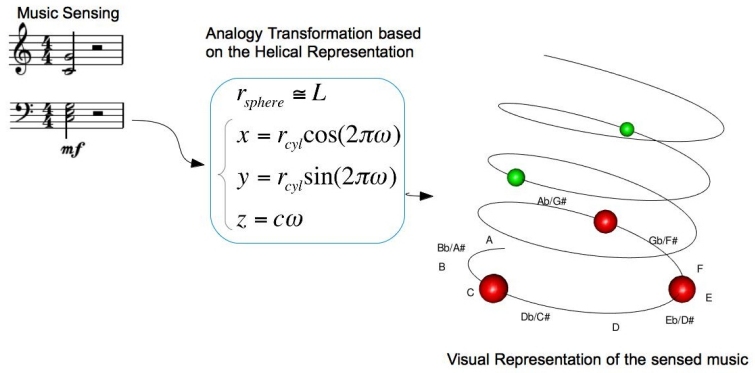


Figure 7: Visual representation of music chords deriving from a "two-component theory" based mapping

way, it is possible to imagine sounds to be associated to visual scenes, and the same can be done with different kinds of perceptions.

We claim that this approach could be a step towards the computation of many forms of the creative process. In future works different types of mapping will be investigated and properly evaluated.

Acknowledgment

This work has been partially supported by the PON01_01687 - SINTESYS (Security and INTElligence SYSstem) Research Project.

References

Augello, A.; Gaglio, S.; Oliveri, G.; and Pilato, G. 2013a. Acting on conceptual spaces in cognitive agents. In Lieto and Cruciani (2013), 25–32.

Augello, A.; Gaglio, S.; Oliveri, G.; and Pilato, G. 2013b. An algebra for the manipulation of conceptual spaces in cognitive agents. *Biologically Inspired Cognitive Architectures* 6(0):23 – 29. {BICA} 2013: Papers from the Fourth Annual Meeting of the {BICA} Society.

Biederman, I. 1987. Recognition-by-components: A theory of human image understanding. *Psychological Review* 94:115–147.

Cardoso, A.; Veale, T.; and Wiggins, G. A. 2009. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine* 30(3):15–22.

Chella, A.; Frixione, M.; and Gaglio, S. 1997. A cognitive architecture for artificial vision. *Artificial Intelligence* 89(1?2):73 – 111.

Chella, A. 2013. Towards a cognitive architecture for music perception. In Lieto and Cruciani (2013), 56–67.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In Raedt, L. D.; Bessière, C.; Dubois, D.; Doherty, P.; Frasconi, P.; Heintz, F.; and Lucas, P. J. F., eds., *ECAI*, volume 242 of *Frontiers in Artificial Intelligence and Applications*, 21–26. IOS Press.

De Paola, A.; Gaglio, S.; Re, G. L.; and Ortolani, M. 2009. An ambient intelligence architecture for extracting knowledge from distributed sensors. In *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, ICIS '09, 104–109. New York, NY, USA: ACM.

French, R. M. 1995. *The Subtlety of Sameness: A Theory and Computer Model of Analogy-making*. Cambridge, MA, USA: MIT Press.

Gaglio, S.; Puliafito, P. P.; Paolucci, M.; and Perotto, P. P. 1988. Some problems on uncertain knowledge acquisition for rule based systems. *Decision Support Systems* 4(3):307–312.

Gärdenfors, P. 1988. Semantics, conceptual spaces and the dimensions of music. In Rantala, V.; Rowell, L.; and Tarasti, E., eds., *Essays on the Philosophy of Music*. Helsinki: Philosophical Society of Finland. 9–27.

Gärdenfors, P. 2000. *Conceptual spaces - the geometry of thought*. MIT Press.

Gärdenfors, P. 2004. Conceptual spaces as a framework for knowledge representations. *Mind and Matter* 2(2):9–27.

Grace, K.; Saunders, R.; and Gero, J. 2008. A computational model for constructing novel associations. In Gervás, P.; Pérez; and Veale, T., eds., *Proceedings of the International Joint Workshop on Computational Creativity 2008*, 91–100. Madrid, Spain: Departamento de Ingeniera del Software e Inteligencia Artificial Universidad Complutense de Madrid.

Halford, G.; Wilson, W.; Guo, J.; Gayler, R.; Wiles, J.; and Stewart, J. 1994. Connectionist implications for processing capacity limitations in analogies. *Advances in connectionist and neural computation theory, Analogical Connections* 2:363–415.

Harnad, S. 1990. The symbol grounding problem. *Physica D* 42:335–346.

Hofstadter, D. R., and Mitchell, M. 1994. The copycat project: A model of mental fluidity and analogy-making. In Holyoak, K. J., and Barnden, J. A., eds., *Advances in Connectionist and Neural Computation Theory*. Norwood, NJ: Ablex Publishing Corporation.

Holyoak, K., and Thagard, P. 1989. Analogical mapping by constraint satisfaction. *Cognitive Science* 13:295–355.

Holyoak, K. J.; Gentner, D.; Kokinov, B.; and Gentner, D. 2001. *Introduction: The place of analogy in cognition*. The Analogical Mind: Perspectives from cognitive science. Cambridge, MA: MIT press.

Hummel, J., and Holyoak, K. 1996. Lisa: a computational model of analogical inference and schema induction. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society*.

Jung, H.; Menon, A.; and Arkin, R. C. 2011. A conceptual space architecture for widely heterogeneous robotic systems. In Samsonovich, A. V., and Johannsdottir, K. R., eds., *BICA*, volume 233 of *Frontiers in Artificial Intelligence and Applications*, 158–167. IOS Press.

Kazjon Grace, J. G., and Saunders, R. 2012. Representational affordances and creativity in association-based systems. In Maher, M. L.; Hammond, K.; Pease, A.; Pérez, R.; Ventura, D.; and Wiggins, G., eds., *Proceedings of the Third International Conference on Computational Creativity*, 195–202.

Kokinov, B., and French, R. M. 2006. *Computational Models of Analogy-making*, volume 1 of *Encyclopedia of Cognitive Science*. John Wiley and Sons, Ltd. 113–118.

Krumnack, U.; Khnberger, Kai-Uwe, S. A.; and Besold, T. R. 2013. Analogies and analogical reasoning in invention. In Carayannis, E., ed., *Encyclopedia of Creativity, Invention, Innovation and Entrepreneurship*. Springer New York. 56–62.

Lieto, A., and Cruciani, M., eds. 2013. *Proceedings of the First International Workshop on Artificial Intelligence and Cognition (AIC 2013) An official workshop of the 13th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2013), Torino, Italy, December 3, 2013*, volume 1100 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Marr, D. 1982. *Vision*. New York: W.H. Freeman and Co.

Oliveri, G. 1997. Mathematics. a science of patterns? *Synthese* 112(3):379–402.

Oliveri, G. 2007. *A Realist Philosophy of Mathematics*. Texts in Philosophy. Kings College Publications.

Oliveri, G. 2012. Object, structure, and form. *Logique et Analyse* 219:401–442.

Oxenham, A. 2013. The perception of musical tones. In Deutsch, D., ed., *The Psychology of Music*. Amsterdam, The Netherlands: Academic Press, third edition. chapter 1, 1–33.

Révész, G. 1954. *Introduction to the psychology of music*. University of Oklahoma Press.

Scholkopf, B., and Smola, A. J. 2001. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Shepard, R. N. 1982. Geometrical approximations to the structure of musical pitch. *Psychological Review* 89(4):305–333.

Tanguiane, A. 1993. *Artificial Perception and Music Recognition*. Number 746 in Lecture Notes in Artificial Intelligence. Berlin Heidelberg: Springer-Verlag.

Wilson, W. H.; Halford, G. S.; Gray, B.; and Phillips, S. 2001. The star-2 model for mapping hierarchically structured analogs. In *World Bank, Human Development 4 (AFTH4). Washington DC*, 125–159. MIT Press.

The FloWr Framework: Automated Flowchart Construction, Optimisation and Alteration for Creative Systems

John Charnley, Simon Colton and Maria Teresa Llano
Computational Creativity Group, Department of Computing,
Goldsmiths, University of London, UK
ccg.doc.gold.ac.uk

Abstract

We describe the FloWr framework for implementing creative systems as scripts over processes and manipulated visually as flowcharts. FloWr has been specifically developed to be able to automatically optimise, alter and ultimately generate novel flowcharts, thus innovating at process level. We describe the fundamental architecture of the framework and provide examples of creative systems which have been implemented in FloWr. Via some preliminary experimentation, we demonstrate how FloWr can optimise a given system for efficiency and yield, alter input parameters to increase unexpectedness, and build novel generative systems automatically.

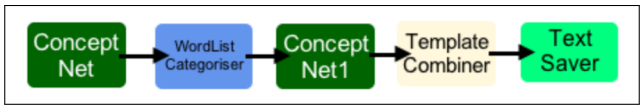
Introduction

One of the main reasons people give for why software should not be considered creative is because it follows explicit instructions supplied by a programmer. One way to reduce such criticisms is to get software to write software, because if a program writes its own instructions, or the code of another program, some level of creative responsibility has clearly been handed over. Automated programming techniques such as genetic programming have been used in creativity projects, such as evolutionary art (Romero and Machado 2007), and software innovating at process (algorithmic) level has been studied in this context. Moreover, machine learning approaches such as inductive logic programming (Muggleton 1991) clearly perform automated programming. In both these cases, programs are generated for specific purposes. In contrast, we are interested here in how software can innovate at process level for exploratory purposes, i.e., where the aim is to invent a new process for a new purpose, rather than for a given task.

Getting software to write code directly is a long-term goal, and we have performed some early work towards this with the invention of game mechanics at code level (Cook et al. 2013). Such code generation will likely be organised at module level, so it seems sensible to study how programs can be constructed in formalisms such as flowcharts over given code modules, to study creative process generation. Flowcharts are used extensively for visualising algorithms, e.g., UML is a standard for representing code at class level (Rumbaugh, Jacobson, and Booch 2004). There are also a handful of systems which allow flowcharts to be developed and automatically converted into code. These include the MSDN VPL (msdn.microsoft.com/bb483088.aspx), the RAPTOR system (Carlisle et al. 2004), and IBM's WebSphere, which allows programmers to visualise the interaction between nodes and produce fully-functional systems on a variety of platforms (ibm.com/software/uk/websphere). Also, *Visual Programming* systems such as Blockly, (code.google.com/p/blockly), AppInventor (appinventor.mit.edu) and Scratch (scratch.mit.edu) allow the structure of a program to be described by using different types of blocks.

We could certainly have investigated process-level innovation by implementing software to automatically control the flowcharting systems mentioned above. However, these systems have been developed to support human-centric program design, and we have had many difficult experiences in the past where we have wrestled unsuccessfully with programmatic interfaces to such frameworks. In addition, in line with usual software engineering paradigms, there is an emphasis on being able to explicitly specify what programs do and an expectation of perfect reliability in the execution of those programs. We are more interested in a flowcharting system able to be given vague instructions (or indeed, none at all) and with some level of automation, produce valuable, efficient flowcharts for generative purposes. For these reasons, we decided to build the FloWr (Flo)wchart (Wr)iter system from scratch with a clear emphasis on automated optimisation, alteration and construction of systems. This paper describes the first release of this framework.

In the next section, we describe the fundamentals of the framework: how programs are represented as scripts which can be created and manipulated visually as flowcharts, and how developers can follow an interface to introduce new code modules to the system. Following this, we detail a FloWr flowchart for poetry generation which uses Twitter, and we use this in an investigation of flowchart robustness. We then present some preliminary experiments to test the viability of FloWr automating various aspects of flowchart design. In particular, we investigate ways in which it can alter and optimise given flowcharts, and we describe an experiment where FloWr invented novel flowcharts from scratch. Notwithstanding a truly huge search space, we show there is much promise for process-level innovation with this approach, and we conclude with a discussion of future research and implementation work.




```
text.retrievers.ConceptNet.ConceptNet_0
dataFile:simple_concept_net_1p0_sorted.csv
relation:IsA
rhsQuery:animal
minScore:0
#wordsOfType = answers[*]

...WordListCategoriser.WordListCategoriser_0
wordList:child;human;apple;
stringsToCategorise:#wordsOfType
#filteredFacts = textsWithoutWord[*]

text.retrievers.ConceptNet.ConceptNet_1
dataFile:simple_concept_net_1p0_sorted.csv
lhsQueries:#filteredFacts
relation:CapableOf
minScore:0
#propertyFacts = facts[*]

...TemplateCombiner.TemplateCombiner_0
templateText:
What if there was a little c1Texts[*][0]
who couldn't c1Texts[*][2]?
numRequired:1000
c1Texts:#propertyFacts
#whatifs1 = instantiatedTemplates[r5]

utility.saving.TextSaver.TextSaver_0
dir:/Output/Flow/whatifs
textsToSave:#whatifs1
```

Figure 1: Ideation script and corresponding flowchart

The FloWr Framework

We aim to use the FloWr framework to investigate automatic process generation via the combination of code modules. As discussed in the subsections below, our approach has been to implement a number of such code modules, which we call *ProcessNodes*, engineer an environment where *scripts* direct the flow of data from module to module, and develop a graphical user *interface* (GUI) to enable visual combination of ProcessNodes into scripts using a flowcharting metaphor.

Individual ProcessNodes

Focusing on generative language systems, we have implemented a *repository* of 39 ProcessNodes for a variety of tasks, from the generation of new material, to text retrieval, to analytical and administrative tasks. For instance, in the repository, there is a ProcessNode for downloading tweets from Twitter, one for performing sentiment analysis, and one for simply outputting text to a file. A new node must extend the Java ProcessNode base class, by implementing its abstract *process* method, which will be called whenever the module is executed. The developer can write whatever software they see fit in the node, and this may call external code in any language. The developer can specify certain *input parameters* for the process, as public fields of the class, along with an optional list of allowed or default values for each parameter. As mentioned below, the scripting mechanism enables variables to be specified, which hold output from processes, and can be substituted in as the input parameters of other nodes. This facilitates the flow of data. At runtime, using Java's reflection mechanism, FloWr will set each ProcessNode's parameters according to the current state of processing, i.e., explicit assignments of the current value of variables to input parameters, prior to calling the *process* method for the node.

The ProcessNode superclass provides a number of utility methods that a node developer can use during processing, such as determining the local location of the *data directory* which holds non-code resources. There are also methods for reporting processing errors during runtime, which developers can use to neatly handle exceptions and other failures. The *process* method of each ProcessNode returns an object of type *ProcessOutput* which holds all the output from the node, hence developers create a Java class that extends the ProcessOutput base class. This facilitates internal FloWr functionality for determining the types of output variables and checking whether a script specifies passing objects of the right type from one node to another (again using Java reflection). Developers can use bespoke or existing classes as fields within output classes, so they can create more complex data-structures for node output. Developers should be aware, however, that most nodes take as input primitive types such as integers and strings, and collections of these, so if they want the output from their nodes to be used by others, their ProcessOutput classes will probably need to have fields at some level in a standard format.

A Scripting Mechanism

A FloWr system is a collection of task-specific ProcessNodes, with a description of how data from each node is selected as input to others, expressed using a script syntax. An example script, which has been edited a little to improve clarity, is given in figure 1. The functionality of this script is described in the subsection on automated optimisation. Each paragraph of the script describes a ProcessNode by specifying its type, configuration and output. The first line is the *type* of node, which refers to the Java class called when that node is run in a system. In figure 1, the first process uses the *ConceptNet* class, from the *text.retrievers* package. Suffixes are used to differentiate between multiple instances of the same node type in a script. When a script is parsed, each type must be an instantiable compiled subclass of ProcessNode in the stated package, which must also contain a valid ProcessOutput subclass.

The next lines in the paragraph specify how the input parameters should be initialised at runtime as name:value pairs. The *name* indicates the parameter to be initialised, and the *value* can be either a simple assignment, or a variable representing some output from another node. Script parsing checks that each name refers to a publicly accessible field of the ProcessNode class, which can be validly assigned with the specified value or the value of the variable indicated. Default parameter assignments are used where a parameter value is blank. Node developers can define any

parameters, so they could develop a single node that operates with various input types, to build more robust systems.

Variable definitions are a **#**-prefixed alphanumeric label and an *output specifier* for a particular part of the output from a process. As mentioned above, each ProcessNode class must have an associated ProcessOutput class. The output specifier refers to the fields defined within this output class, which will be populated by the node at runtime. In its simplest form, the specifier indicates a particular field to assign to the variable. Alternatively, they can be separated by dots, where each segment is a field relative to the specifier to its left. Where the indicated field is a list, square brackets are used to indicate a *selection specifier*, which identifies a subset of elements to be assigned to the variable. The acceptable selection specifiers are: *****: all elements; **f*n***: the first *n* elements; **l*n***: the last *n* elements; **m*n***: the middle *n* elements; and **r*n***: *n* randomly chosen elements.

When a script is run, all processes are checked for syntax errors and data-type inconsistencies. FloWr determines the process run order by inspecting dependencies between output variables and input parameters, and errors are raised whenever there are problematic loops in a script. FloWr then steps through each node in the run order by instantiating an appropriate ProcessNode object, assigning its parameters according to the script, calling its *process* method to execute the node, and storing the output. In the example script of figure 1, we see that the *ConceptNet_0* ProcessNode has output with an `answers` field, which is a list. The whole list (indicated by `answers[*]`) is assigned to the variable `#wordsOfType`, which is passed into the *WordListCategoriser_0* ProcessNode as the input parameter `stringsToCategorise`. In this simple script, each node except the last one assigns a single aspect of its output to a variable, which is passed onto the next node.

A Flowcharting Interface

The FloWr GUI shown in figure 2 is the primary system development tool, where flowcharts are used to visually represent the interaction between ProcessNodes. The interface has several components. Firstly, the central panel displays the flowchart currently being worked upon, with individual ProcessNodes shown as boxes and the arrows between them indicating the transfer of data. The flowchart in figure 2 – the functionality of which is described in the next section – has 16 nodes of 13 different types, with colour coding indicating nodes of the same type or which perform similar tasks. For instance, blue boxes in figure 2 represent ProcessNodes which categorise texts (using word sense, sentiment, regular expressions and a user-supplied word list). To add a new node to a flowchart, the user right clicks the main panel and chooses from a series of popup menus. As might be expected, flowchart boxes can be dragged, resized, deleted, copied and renamed, and multiple boxes in sub-charts can be selected, moved, resized and deleted simultaneously.

When a box is clicked, it gains a thick grey border, and the arrows going into/out of it gain circles, which when clicked populates the *mappings* (upper) internal frame in the GUI with the output variables and input parameters of the two ProcessNodes joined by the arrow. The mappings between nodes can be edited by hand, and arrows are automatically generated whenever an output variable is used as an input parameter for another node. Clicking on a box populates the mappings frame with the input variables and output parameters for that ProcessNode, and populates the *output* (lower) internal frame with the values of the output variables, if they have previously been calculated via a run of the system. In figure 2, we see that the user previously selected the output for the SentimentSorter node, (which is a poem about being abusive) in the output frame. They then selected the circle on the arrow between two nodes, and the output variables and input parameters for LineSplitter and SentimentSorter were displayed accordingly in the mappings frame.

A small black panel containing a play and stop button for executing and halting the script is shown at the top right of the flowchart panel. When the user has chosen to execute the flowchart multiple times from a menu, a number indicating which run is executing is shown in this panel (the number 17 in figure 2). The user can double click a node, and FloWr will run all the processes leading into that node, including it, but not nodes which occur later in the script run order. When the flowchart is running, the node which is actually executing is given a red border: in figure 2 the TextRankKeyphraseExtractor node is running. Nodes can take some time to finish executing, and it is often useful for their output, and the output for all the nodes earlier in the flowchart, to be frozen, i.e., calculated once and stored rather than generated when that process is run again. This can be done using the interface and is indicated with a pushpin in the flowchart box: in figure 2, the Twitter node has been frozen. The pushpins in the mappings frame and the output frame can be used to stop their context from changing when boxes on the flowchart are clicked.

An Example FloWr System

We have used FloWr to hand-craft a number of systems, including flowcharts for poetry generation in a manner similar to that of (Colton, Goodwin, and Veale 2012), where newspaper articles were manipulated to produce poems. We have also used FloWr to perform automated theory formation using the same production rule-based method employed by the HR system (Colton 2002), and we have re-implemented aspects of The Painting Fool software (Colton 2012). Finally, as discussed in the next section, we have used FloWr scripts to produce fictional ideas, with experiments using this given in (Llano et al. 2014a) and (Llano et al. 2014b). In each of these instances, we have developed a fully-operational system and the FloWr GUI has enabled a clear visualisation of the overall system, enabling us to design, edit and tweak each implementation. The ProcessNodes required and the flowcharts implementing these systems are available in the FloWr distribution.

The flowchart in figure 2 produces poems as a collection of related tweets from Twitter in a relatively sophisticated way. Execution begins with a *Dictionary* ProcessNode which selects all the 5,722 words from a standard dictionary with a frequency of between 90% and 95%, with word frequency determined using the Kilgarriff database (Kilgarriff 1997), which was mined from the British National

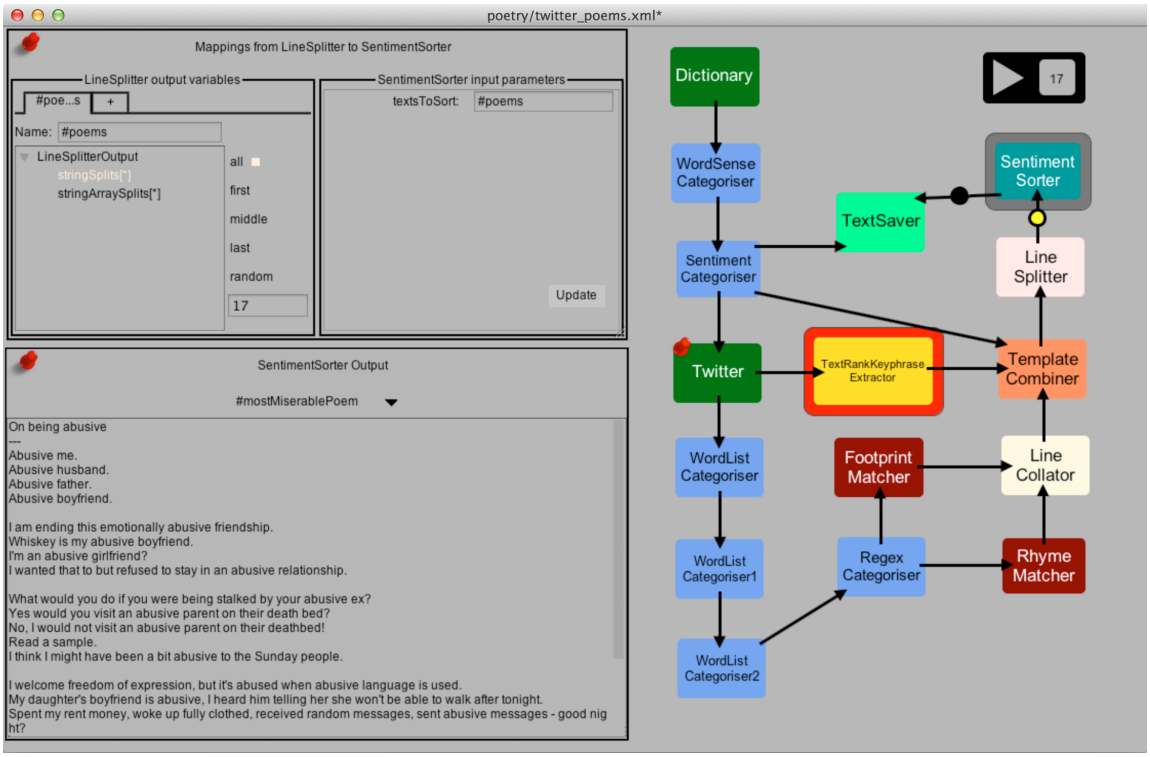


Figure 2: The FloWr flowcharting graphical user interface.

Corpus. Such words are relatively common but not too common or too uncommon in the language. Next in the flowchart, a *WordSenseCategoriser* selects the 772 words that are adjectives (in terms of their main sense) as per the British National Corpus tagset (Leech, Garside, and Bryant 1994). A *SentimentCategoriser* node then splits the adjectives into categories based upon how positive or negative a word is, using the Afinn sentiment dictionary (`fnielsen.posterous.com/tag/afinn`) expanded by adding synonyms from WordNet. From the list of 211 *negative words*, i.e., scoring -1 or less for valency, a single word is randomly chosen as the poem theme, using the variable selection syntax [r1] in the underlying script, as described above.

The *Twitter* ProcessNode accesses the Twitter web service through the Twitter4J library (`twitter4j.org`), and retrieves a maximum of 1,000 tweets containing the theme word – there may be less if the word is not mentioned in many recent tweets. Tweets are cached to make retrieval quicker later on. Also, as part of the retrieval process, the tweets are filtered to remove copies and tweets containing a word which cannot be pronounced, as per the CMU pronunciation dictionary (CPD, at `www.speech.cs.cmu.edu/cgi-bin/cmudict`), or which cannot be parsed using the Twokenize tokenizer (`bitbucket.org/jasonbaldridge/twokenize`). We have found that the 90-95% word frequency previously mentioned ensures that there are usually sufficient tweets (counted in the hundreds) after the filtering process, but that the tweets tend to be less banal than usual, as the usage of a somewhat uncommon word requires some thought.

The retrieved tweets are used in two ways. Firstly, a *TextRankKeyphraseExtractor* node extracts keyphrases using an implementation (`lit.csci.unt.edu`) of the TextRank algorithm (Mihalcea and Tarau 2004) over the entirety of the tweets collated as a paragraph of text. As an example, the poem theme in the run presented in figure 2 was 'abusive', and the keyphrases of 'abusive husband', 'abusive father' and 'abusive boyfriend' were extracted. Secondly, the tweets are passed through a triplet of *WordListCategoriser* nodes which are used to exclude tweets containing undesired words. The first filter removes tweets containing any of a pre-defined list of first names, discarding many tweets about particular people, which are too specialised for our purposes. The second removes tweets containing Twitter-related words such as *retweet*, and the third removes tweets containing certain profanities. The *RegexCategoriser* ProcessNode then splits the tweets into two sets based upon whether or not they contain personal pronouns (I, we, they, him, her, etc.). Only tweets containing personal pronouns are kept, which helps remove commercial service announcements, which are dull. In the 'abusive' example, from the 1000 tweets retrieved, 110 were removed as duplicates or for being unpronounceable/non-tokenisable. 80 were further removed for including first names, 33 for including Twitter terms, 22 for including profanities, and 262 were removed because they included no personal pronouns, leaving 493 tweets for the construction of stanzas in the poem.

The remaining tweets are processed by a *RhymeMatcher* node which finds all pairs of tweets with the same two phonemes at the end, when parsed by the CPD. The num-

On Being Eerie

Eerie me.
Eerie feeling.
Bit eerie.

I hate the basement level of buildings.
You always lose reception and it's always quiet and eerie.
This doesn't quite capture the eerie pink glow of this morning.
Is pop culture satanic?
In a spiritual (not religious) sense?
I don't really know.
But man, there are some eerie parallels.
It's concerning.
I find it very eerie when someone is tinkering with your teeth and telling jokes.
Or is that just me?

I hate winter and the cold, but I love how silent the night is during cold winter weather.
It's eerie, but peaceful.
Old school!
It was always eerie.
No one around, and completely quiet.
It's like being on the wrong end of the apocalypse.
Experiencing the eerie light of total eclipse.
I'm going through it today.
The fact that I'm talking about my grandma in a past tense is eerie and weird to me.

I saw weird stuff in that place last night.
Weird, strange, sick, twisted, eerie, godless, evil stuff.
And I want in.
Yes - that is quite an eerie sound!
It's so eerie listening to the crying in the background.
I can understand that.
It just feels eerie to have it haunt you (word-for-word) by different users.

I hope the cloud stayed away for you.
Wow, how was the eerie darkness?
I thought I told you.
Oops.
Weird, eerie, strange portraits and locations.
An antique metal ship and a candle make for eerie (and awesome) decorations.
I mean the art direction is eerie.
I'm pretty sure it's hogwash.

Bit eerie.
Eerie feeling.
Eerie me.

Figure 3: Example Twitter poem: *On Being Eerie*.

ber of matching phonemes can be changed to increase or decrease the amount of rhyming. From these, 250 pairs are randomly chosen (or the entire set, if less than 250). The tweets are likewise processed by the *FootprintMatcher* node, which counts the number of syllables, again using the CPD, and finds all pairs of tweets with the same footprint. As before, 250 pairs of tweets are chosen randomly. Next, *LineCollator* constructs sets of 16 different tweets in quadruples of the form ABBA, where the As are a pair with equal footprints and the Bs are a pair which rhyme. An example quadruple is as follows (note the two central lines rhyme, and the outer lines both have 17 syllables):

I hope the cloud stayed away for you. Wow, how was the eerie darkness?
I thought I told you. Oops. Weird, eerie, strange portraits and locations.
An antique metal ship and a candle make for eerie (and awesome) decorations.
I mean the art direction is eerie. I'm pretty sure it's hogwash.

The *TemplateCombiner* node brings all the processed information together into a poem based upon a specified poem template. The inputs to this process are the theme word which becomes part of the poem title, the keyphrases which provide a context at the top and (reversed) at the bottom of the poem, and the quadruples from *LineCollator*, which each form a stanza of the poem. *TemplateCombiner* is told to produce 20 poems by choosing 20 sets of quadruples from *LineCollator* randomly. The *LineSplitter* ProcessNode takes each poem and splits any line where there is a period (tweets often contain two or more sentences), which tends to make the poems more *poem-shaped*. Finally, the *SentimentSorter* node selects the poem with the most negative affect, which is saved to a file by the *TextSaver* ProcessNode. This is given the theme word as an input, and the file is so named.

Freq(%)	Structure	Neg.	Stanzas	Yield(%)
85-90	$FRRF$	false	4	94
90-95	$FRRF$	true	4	94
80-85	$FRRF$	false	4	90
90-95	$R_1R_2R_2R_1$	false	4	80
90-95	$R_1R_2R_2R_1$	true	4	74
90-95	$R_1R_2R_2R_1$	false	6	46
90-95	$R_1R_2R_2R_1$	true	6	12

Table 1: Yield results for Twitter poetry flowchart.

In general, we have found that these Twitter poems are surprising and interesting. In particular, the slight rhyming in the centre of the poems is noticeable, and the multiple voices expressed through 16 different tweets, coupled with the often rushed nature of the tweets can give the poems a very dynamic feel. Another example poem is given in figure 3, where the theme was 'eerie'. This poem was recited as part of a poetry evening during a festival of Computational Creativity, in Paris in July 2013 (Colton and Ventura 2014).

The nature of the flowchart, including the ProcessNodes, the I/O connections and the parameterisation of the processes was carefully specified and tweaked by hand over many hours, to produce a poem most of the time, for different adjectives. One of the benefits of the flowcharting approach is that variations can be easily tried out – but it would be frustrating if the yield of poems wasn't consistent. To investigate the robustness of the flowchart, we varied the word frequency parameters in the *Dictionary* to test the retrieval of tweets containing less common words. We also made the poem construction more difficult. Firstly, we introduced all-rhyming stanzas $(R_1R_2R_2R_1)$ rather than the footprint-rhyming structure $(FRRF)$. Secondly, we introduced an additional *SentimentCategoriser* node to ensure that only tweets with an average (Neg)ative valency were used. Finally, we increased the number of stanzas from 4 to 6. For each of 7 setups given in table 1, we provide the yield produced from 50 runs of the flowchart. We note that the flowchart is fairly robust to lowering the theme word frequencies, but the volume of tweets didn't support well the construction of more complex poems. In fact, only 12% of runs resulted in a poem when six $R_1R_2R_2R_1$ stanzas with only negative tweets were sought. This indicates that there is a limit to how far a successful flowchart can be tweaked before it loses its utility.

Automation Experiments

We present here some preliminary experiments to automatically alter, optimise and generate flowcharts. As mentioned previously, a driving force for the project is to study the potential of automated process generation. FloWr simplifies the process of constructing a system but, as highlighted in the previous section, fine-tuning a chart can be a laborious process. For example, the flowchart/script in figure 1 was developed by hand for a project where the ConceptNet database of internet-mined facts (Liu and Singh 2004) was used for fictional ideation in the context of Disney cartoon characters, as described in (Llano et al. 2014a). Given a *theme* word like 'animal', the flowchart uses the *ConceptNet1* node to find all Xs for which there is a fact [X,IsA,animal], removes spurious results, such as [my_husband,IsA,animal] with the *WordListCategoriser*, and then for a given relation, R, finds all the facts of the form [X,R,Y], using the *ConceptNet2* node. To produce the fictional idea, it inverts the reality of each fact using the *TemplateCombiner* node to produce an evocative textual rendering. For instance, the fact that [cat,Desires,milk] becomes "What if there was a little cat who was afraid of milk?".

In further testing, we substituted 'animal' for other theme words such as 'machine', and produced ideas such as: "What if there was a little toaster who couldn't find the kitchen" (by inverting the LocatedNear relation in this case). There are 49 ConceptNet relations and a large number of couplings of these with theme words, many of which yielded no results. For instance, we found no facts about types of machines and the Desires relation, presumably as machine don't tend to desire things. Focusing on animals, it took around 2 hours to produce the first working flowchart which produced a non-zero yield of facts which could be usefully inverted for the invention of Disney characters. One of the benefits of automation we foresee would be a substantial reduction in this type of manual fine-tuning.

Flowcharts can be constructed and altered in several ways. ProcessNodes can be added, removed or replaced with alternatives. Parameterisations of nodes and the links between them can be amended by modifying, creating or deleting variables and changing input settings. The space of all possible constructions and alterations is vast and, at this early stage, we have restricted ourselves to a subset. Specifically, we have considered changes to parameterisations of existing flowcharts and we describe some experiments in the following section, followed by how these can be guided to achieve particular optimisation objectives. After this, we consider constructing flowcharts from scratch by sequentially adding additional ProcessNodes. In all cases, FloWr has generated flowcharts representing novel and interesting creative tasks whilst avoiding an element of manual construction effort.

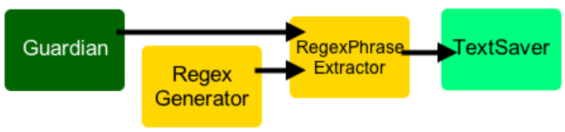


Figure 4: Flowchart for automated regex generation.

S	NW	FWLen	WLCh	FLCh	LLCh	Yield(%)	Av.
1	3	3-6	equal	equal	none	55	48.6
2	3	3-6	equal	any	none	42	12.1
3	3-5	3-6	any	any	none	24	9.24
4	3	3-6	incr.	incr.	none	38	5.1
5	3-5	3-6	any	any	any	0	0

Table 2: Regex generation test yields (tongue twister texts).

Flowchart Alteration

When motivating the building of the FloWr framework in the introduction, we noted that we want the approach to produce unexpected results, with FloWr scripts being somewhat unpredictable. One way to increase unexpectedness is to randomly alter input parameters to ProcessNodes at run-time. We investigated this via the generation of simple tongue-twister texts, by extracting word sequences using regular expressions. We implemented a *RegexGenerator* ProcessNode which produces regular expressions (regexes) such as:

\bs[a-zA-Z]4\b\s1,\bs[a-zA-Z]5\b\s1,\bs[a-zA-Z]6\b

When applied to a corpus of text, this regex extracts all triples of words of length 5, 6 and 7 which begin with the letter 's'. We applied this to a corpus of 100,000 Guardian newspaper articles, and it returned 21 triples, such as 'small screen success' and 'short skirts showing'.

The input parameters to *RegexGenerator* specify the number of words (NW) in the phrases sought, what the first word's length (FWLen) should be, and how the word lengths should change (WLCh): either increasing, decreasing, staying the same, or no(ne) change. The parameters also enable the specification of the first letter of the first word, and how subsequent first letters should change lexicographically (FLCh): increasing, decreasing, staying the same or no(ne) change. The last letter changes (LLCh) can similarly be specified. Importantly, FloWr can be instructed to choose each parameter randomly from a given range. For start and end letters, this range is a-z, for word length and letter changes it is {increase, decrease, equal, none} and the integer value for NW and FWLen can be specified to be within a user-given range.

We implemented the flowchart in figure 4 to input the whole Guardian corpus and a generated regular expression in the *RegexPhraseExtractor* node, and output the resulting text (if any) to a file. We ran five sessions with different input parameter ranges for the *RegexGenerator* node. For each session, we specified that the first letter of the first word should be chosen randomly. In each session, we ran the flowchart 100 times and recorded the yield as the percentage of times when text was actually produced. We also recorded the average number of lines of text produced (i.e., the average number of hits for the regular expression in the corpus). The results are given in table 2. We see that (S)etup 5 is completely unconstrained and the space which is randomly sampled from is dense in poor regexes which have no hits in the corpus: the yield is zero. However, with some constraining of the regex ranges allowed, the yield increases almost to 50%. Also, as expected, the average number of

hits increased in line with the yield. The following are two tongue twisters found in the results for setups 1 and 4:

posted pretax profit	cancer despite everyone
please please please	classy devices emerging
petrol prices played	carbon dioxide expelled
profit public policy	carbon dioxide emission
poorer people pushed	choice defense everyone

In other experiments, with the ideation flowchart of figure 1, we looked at automatically changing the theme word. To do this, a *WordNet* ProcessNode was used to find hypernyms of *animal*, which returned the words *organism* and *being*. We then requested the hyponyms of each of these, which generated 87 alternative themes, which were substituted for the theme in the flowchart. Several of the themes produced a high yield of invertible facts, with 13 theme/relation combinations generating more facts than the highest found by hand. Three of these used theme word *person*, e.g., with the *CapableOf* relation, which generated 2,154 ideas such as the concept of actors being able to face an audience. Similarly, the theme words *individual* and *plant* had high yields. However, one word that was identified automatically using this method was *flora*, which gave interesting invertible facts about trees, such as being homes for nesting birds and squirrels. These were not considered in our manual experiments using the *plant* theme. In a similar way, we used *ConceptNet* to find theme words by inspecting all the *IsA* relations in its database, from which it identified 11,000 themes. Using these, we found the highest yield with the theme *mammal* and the relation *NotDesires*, which we hadn't found manually. This generated 568 facts, mainly about people, e.g., the ideas that people don't want to be eaten or bankrupt, both of which led to interesting fictional inversions.

Optimising Flowcharts

We performed some experiments in automating the task of finding high-yield configurations for the ideation flowchart of figure 1. To do this, we provided a list of themes and asked FloWr to consider all possible pairings of theme and ConceptNet relation. To assess the yield of a ProcessNode, FloWr uses Java reflection to traverse the structure of its output object and count the objects and sub-objects in individual fields or in lists. We have found this to be a reliable measure of output quantity, particularly when assessing relative sizes. It is also general, and will produce a useful yield measure irrespective of the nature of the node and its output. The manual process identified the theme word *animal* and the relationship *CapableOf* as producing the highest yield of 530 usable facts. The automated approach also identified this combination, but it highlighted a more productive relationship for *animal*, namely *LocatedAt*, which provides 1,010 facts. This combination had been overlooked during the manual process, in favour of using the *LocatedNear* relationship, which produced only 39 facts.

We also investigated optimising flowcharts for efficiency. Given a target time reduction and minimum output level for ProcessNodes in a given flowchart, we investigated an approach which identifies small local changes to input parameters that have the most global impact on the system. Firstly, the nodes are ordered according to their increasing contribution to the overall execution time. Considering the slowest ProcessNode, P, first, an attempt is made to establish if the time taken is a consequence of the amount of data it receives, by halving the data given and comparing execution times. If input data is causing P's slow speed, the ProcessNode(s) which produced that input into P are re-prioritised higher than P in the ordering. Moreover, a local goal for each ProcessNode is assigned, which is either to reduce its execution time or the size of its output. Then, local reconfigurations consider incremental changes to numeric and optional parameters until the local goals, or failure, have been met. Any successful local reconfigurations are then applied to the global system and reported to the user if they achieve the overall goals. Multiple tests are used at each stage to confirm that the reported results are consistent.

We successfully applied this approach to the Twitter poetry generator, where it identified that the high average base execution time of 10 seconds was caused by the *WordListCategoriser* nodes processing a high number of tweets from the *Twitter* node. It applied an iterative process, which reduced the *numRequired* parameter by a given percentage for a pre-defined number of steps, noting each time that the node output yield was reduced, eventually settling on a *numRequired* setting of 63. It then tested this on the global system and found that this reduced the overall runtime to 630 milliseconds, whilst still successfully generating poems. In a similar experiment, we optimised another poetry system which used Guardian newspaper articles as source material, as in (Colton, Goodwin, and Veale 2012). The optimisation method found that one node could be optimised by reducing its input size, which led to the altering of another node's input parameters, and a 40% reduction in overall execution time, while the flowchart still produced poems.

Flowchart Construction

We have investigated how to construct FloWr systems from scratch. Working in the context of producing poetic couplets, we tested a method which could generate a system with three to five nodes taken from these sets respectively: {*Twitter, Guardian, TextReader*}, {*WordSenseCategoriser, SentimentCategoriser*}, {*TextRankKeyphraseExtractor, RegexPhraseExtractor*}, {*WordSenseCategoriser, SentimentCategoriser*}, {*FootprintMatcher, RhymeMatcher*}. We used our experience of which nodes work well together to create this structure, and to specify a number of possible options for the input parameters. For some nodes, we were restrictive, e.g., we specified that the *Guardian* node should use a specific date range for selecting articles and always return the same number. For other nodes, we allowed FloWr to use any of the parameter values from the optional lists provided by the node developer. For the *Twitter* node, we chose five dictionary words randomly for queries and *TextReader* was directed to use a set of Winston Churchill speech texts.

Despite these limitations, there are still a huge number of possible combinations to explore. For example, there are 108 possible node combinations, 27,000 parameter combinations and over 261 million variable definition combina-

Figure 5: An automatically generated rhyming couplet system.

tions. The size of this restricted subset makes a brute-force approach intractable, given that many nodes have execution times of over a second. Hence, we tried a depth-first search of all possible systems, by choosing node combinations randomly and configuring each node with input parameters chosen from those allowed at random. Next, the method considers the possible data links between nodes by considering each pair in turn. The set of variables that could be defined in the scripting syntax for the earlier node in the system is compared with the input parameters for the following node. Only those where the output variable type and the input parameter types match will be syntactically valid, and these are chosen from randomly and applied to the script.

We generated 200 scripts using this process and tested each to see whether it produced output from the final node. We found that 17 (8.5%) worked successfully and produced poetic couplets. Of these, 8 contained 3 nodes, 8 contained 4 nodes and one – shown in figure 5 – contained 5 nodes. This script takes Guardian articles from the first week of 2012, extracts the neutral texts in terms of sentiment, and identifies all their key phrases. It selects keyphrases beginning with an adjective and outputs pairs of phrases with the same syllable footprint, producing these:

actual bodily harm	chief inspector working	dangerous driving
metropolitan police	domestic violence	potential recruits

The yield from the 17 scripts varied widely from one to over 4 million couplets. The most commonly used ProcessNode in the successful flowcharts was *TextRankKeyphraseExtractor*, which was used 28% of the time, followed by FootprintMatcher, used 23% of the time. *FootprintMatcher* is more prevalent than *RhymeMatcher* at 5%, because there are more pairs of phrases with the same number of syllables than pairs which rhyme. The *RegexPhraseExtractor* fails to appear, due to limited input data, i.e., there were no strings satisfying the regular expressions sought, due to the limited amount of text available. We experimented with further restricting the types of nodes that could be selected. In particular, using information about the frequency of nodes in successful scripts from the first experiment, we managed to improve the yield of working scripts to 18.5% by allowing only *WordSenseCategoriser* nodes to be used for categorisation.

One particular (four-node) script caught our eye. It takes Churchill texts, extracts keyphrases, keeps only those where the first word has *extreme* sentiment, i.e., ≥ 2 or ≤ -2, then outputs pairs with the same footprint, such as: [great air battle:despairing men] and [greater efforts:greater ordeals]. The 52 poetic couplets that this script generated provided the starting point for a poem written by a collaborator: Russell Clark selected a subset of these pairs, then combined and ordered them into a piece entitled *Churchill's War*, which is shown in figure 6. The poem was one of four submitted for analysis by poetry experts as part of a BBC Radio 4 piece on Computational Creativity (Cox 2014), although a different poem was ultimately read out and analysed.

Churchill's War

Good many people, great differences
good many people: outstanding increase.
Great organisations, greater security
greater security: terrible position

Great combatants, brilliant actions
Great preponderance, greater efforts

Great air battle, despairing men
Great air battle, brilliant actions

Great Britain, good account
Great Britain, good reason

Great flow: Great war
Great flow: Good men

Chess proceeds, good reason
Chess proceeds: victory

Figure 6: A poem based upon the output from an automatically generated process for poetic couplet generation.

Conclusions and Future Work

The FloWr framework enables fairly rapid prototyping of flowcharts for creative systems. We presented here fundamental details of how code modules can be implemented and combined via scripts using a flowcharting front end. We presented flowcharts for producing poems, fictional ideas, tongue twisters and poetic couplets, which re-use nodes for retrieving, categorising, sorting, combining and analysing text. We have performed some experimentation to assess the potential for automating aspects of flowchart design, both to help users construct, vary and optimise flowcharts, and to highlight the potential for FloWr to automatically construct novel processes. The ultimate aim of this project is to provide an environment which encourages third party ProcessNode and flowchart developers to contribute material from which FloWr can learn good practice for innovating in automatic process design. We have already started implementing functionality which enables FloWr to learn flowchart configurations which are likely to produce results. This has aspects in common with other knowledge-

based system design projects, such as *Rebuilder* (Gomes et al. 2005). Ultimately, FloWr will reside on a server, constantly generating, testing and running novel system configurations in reaction to people uploading new ProcessNodes and scripts. We intend to have a large number of nodes covering a variety of different individual tasks in many domains. For instance, we have a variety of NLP nodes, e.g., for Porter Stemming (Porter 1980) and we will be extending this to cover nodes for other tasks, such as tagging and chunking.

The first release of the FloWr framework, along with dozens of ProcessNodes and numerous flowcharts is available at `ccg.doc.gold.ac.uk/research/flowr`. In future releases, we plan a number of improvements to the underlying framework, including much more automation in the system, given the promise shown for this in the experiments described here. The systems that can be implemented currently are quite limited, and we plan to introduce additional programmatic constructs, such as framework level control of looping, and ProcessNode level control of conditionals. We will also implement useful functions, such as FloWr running a sub-flowchart repeatedly until it produces a particular yield for the rest of the flowchart, and translating variables, e.g., from `ArrayList<String>` to `String[]`, to increase flexibility. We will test different search techniques to tame the vast space of flowchart configurations, so that FloWr can reliably generate interesting novel flowcharts, and we will implement the optimisation and alteration routines we have experimented with as default functionalities. We also plan to implement more entire systems in FloWr, in particular we expect The Painting Fool art program (Colton 2012) to eventually exist as a series of flowcharts in FloWr. Also, we have started to port the HR3 automated theory formation system (Colton 2014) to FloWr. We have experimented with HR3 to add adaptability to the Twitter poetry generation flowchart: using concept formation over a given set of tweets, HR3 can successfully find a linguistic pattern which links subsets of tweets, that can be extracted and turned into poem stanzas.

The flowchart in figure 2 is a creation in its own right. To some extent, the value of such flowcharts exists over and above the quality of the output they produce. That is, the way in which the flowchart constructs artefacts is an interesting subject in its own right. For reasons of improving autonomy, intentionality and innovation in computational systems, we believe that software which writes software – whether at code-level or via useful abstractions such as flowcharts – should be a major focus in Computational Creativity research. Automated programming has been adopted, albeit in restricted ways, in highly successful areas of AI such as machine learning, and we believe there will be major benefits for the building of creative systems through the modelling of how to write software creatively.

Acknowledgments

This work has been supported by EPSRC Grant EP/J004049/1 (Computational Creativity Theory), and EC FP7 Grant 611560 (WHIM). We would like to thank Russell Clark for his help with the poetry generation flowcharts and curating their output. We would also like to thank the anonymous reviewers for their helpful comments.

References

Carlisle, M.; Wilson, T.; Humphries, J.; and Hadfield, S. 2004. RAPTOR: Introducing programming to non-majors with flowcharts. *Journal of Computing Sciences in Colleges* 19(4).

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-FACE poetry generation. In *Proceedings of the International Conference on Computational Creativity*.

Colton, S. 2002. *Automated Theory Formation in Pure Mathematics*. Springer.

Colton, S. 2012. The Painting Fool: Stories from building an automated painter. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*. Springer.

Colton, S. 2014. The HR3 discovery system. In *Proceedings of the AISB symposium on computational scientific discovery*.

Colton, S. and Ventura, D. 2014. You Can't Know my Mind: A Festival of Computational Creativity. In *Late Breaking Proceedings of the International Conference on Computational Creativity*.

Cook, M.; Colton, S.; Raad, A.; and Gow, J. 2013. Mechanic miner: Reflection-driven game mechanic discovery and level design. In *Proceedings of the EvoGames workshop*.

Cox, T. (Presenter) Can a Computer Write Shakespeare? BBC Radio 4 documentary, first aired on 15th May 2014.

Gomes, P.; Pereira, F.; Paiva, P.; Seco, N.; Carreiro, P.; Ferreira, J.; and Bento, C. 2005. Rebuilder: a case-based reasoning approach to knowledge management in software design. *Engineering Intelligent Systems for Electrical Engineering & Communications* 13(4).

Kilgarriff, A. 1997. Putting frequencies in the dictionary. *International Journal of Lexicography* 10(2).

Leech, G.; Garside, R.; and Bryant, M. 1994. CLAWS4: The tagging of the British National Corpus. In *Proceedings of the 15th COLING*.

Liu, H., and Singh, P. 2004. Commonsense reasoning in and over natural language. In *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information and Engineering*.

Llano, M. T.; Hepworth, R.; Colton, S.; Charnley, J.; and Gow, J. 2014. Automating fictional ideation using ConceptNet. In *Proceedings of the AISB Symposium on Computational Creativity*.

Llano, M. T.; Hepworth, R.; Colton, S.; Gow, J.; Charnley, J.; Lavrač, N.; Žnidaršič, M.; Perovšek, M.; Granroth-Wilding, M.; and Clark, S. 2014. Baseline Methods for Automated Fictional Ideation. In *Proceedings of the International Conference on Computational Creativity*.

Mihalcea, R., and Tarau, P. 2004. Textrank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Muggleton, S. 1991. Inductive Logic Programming. *New Generation Computing* 8(4).

Porter, M. 1980. An algorithm for suffix stripping. *Program* 14(3).

Romero, J., and Machado, P. 2007. *The Art of Artificial Evolution*. Springer.

Rumbaugh, J.; Jacobson, I.; and Booch, G. 2004. *The Unified Modeling Language Reference Manual*. Pearson Higher Education.

New Developments in Culinary Computational Creativity

Nan Shao, Pavankumar Murali, Anshul Sheopuri
IBM TJ Watson Research Center, Yorktown Heights, NY

Abstract

In this paper, we report developments in the evaluation and generation processes in culinary computational creativity. In particular, we explore the personalization aspect of the quality and novelty assessment of newly created recipes. In addition, we argue that evaluation should be a part of the generation process and propose an optimization-based approach for the recipe creation problem. The experimental results show a more than 41% lift in the objective evaluation metrics when compared to a sampling approach to recipe creation.

1 Introduction

"My children have a preference for meat. How do I create a healthy dish that will be enjoyed by them?" Can a computer help parents with such questions? The culinary domain is a new area for computational creativity, although "made up a recipe" has been listed as one of the 100 creative activities on human creativity rating questionnaire developed by Torrance more than 50 years ago (Sawyer 2012). (Morris et al. 2012) discussed recipe creation restricted to soups, stews and chili. (Varshney et al. 2013) discussed evaluation (work product assessor) motivated by neural, sensory and psychological aspects of human flavor perception, and proposed models for a culinary computational creativity system.

To answer questions like the one listed above, we consider two aspects of the problem: the personalization of dish evaluation and the optimization of dish quality and novelty in a combinatorially complex creativity space. Our contributions to the culinary domain are as follows. First, creativity is only meaningful in the presence of a human audience or evaluator (Wiggins 2006), and humans are inherently different; therefore we explore the personalization aspect of the evaluation metric for a creative artifact. In particular, we consider flavor preference and novelty evaluation of a newly created recipe. Second, we consider evaluation as part of the generation/search process and provide an optimization-based approach for the recipe creation problem. For the latter, we draw inspiration from the search mechanism that (Wiggins 2006) proposed on moving through the complex conceptual space. We hypothesize that our proposed methodological framework can be extended to other creative endeavors as well.

2 Personalization in Culinary Creation

We now turn to detailing a tractable approach for assessing personalized flavor preference and novelty. The approach is motivated by the human flavor perception science, technology to draw information from the web, and the work in (Varshney et al. 2013).

2.1 Flavor Preference

Flavor enhancement, balance and substitution are choices that we make to live a healthy life. Often, we may want to enhance the flavor of our favorite ingredient. However, we may need to balance the flavor of healthy but not tasteful ingredients. Moreover, we may want to substitute red meat with a plant-based product to meet a dietary constraint and, at the same time, not lose the meaty flavor. In our work, we propose a methodology to address these personalized flavor preferences in a computational creativity system.

Knowledge of how humans perceive flavors is necessary to build a system that accurately estimates a human's evaluation for creativity. For this reason, (Varshney et al. 2013) proposed a model for pleasantness which correlates olfactory pleasantness with its constituent ingredients and flavor compounds in those ingredients based on recent olfactory pleasantness study (Haddad et al. 2010; Khan et al. 2007). The smell of food is a key contributor to flavor perception, which is, in turn, a property of the chemical compounds contained in the ingredients (Burdock 2009; Shepherd 2006). Therefore a tractable step towards a data-driven model for flavor enhancement, balance and substitution is a model for odor similarity. For example, we could enhance the flavor of a featured ingredient by adding other foods with perceptually similar odors.

Recent work has shown that perceptual similarity of odorant-mixtures can be predicted (Snitz et al. 2013). Consistent with the synthetic brain processing mechanism in olfaction, human perception groups many mono-molecular components into singular unified percept. Each odorant-mixture is modeled as a single vector made up of the structural and physicochemical descriptors of the mixture. The angle distance between two vectors is a meaningful predictor of the perceptual similarity of two odorant-mixtures. Therefore, given any two odorant-mixtures, we can predict a significant portion of their ensuing perceptual similarity.

Since food ingredients contain several flavor compounds (Ahn et al. 2011), and dishes contain several ingredients, we can predict the flavor perceptual similarity and dissimilarity of a featured ingredient and a dish to provide quantitative measurement on how the dish enhances or balances the featured ingredient flavor. We describe one approach here and show some results in Table 1, where the personal preference is to enhance the beef flavor of a stew. The formulation of the approach on flavor enhancement is described as below:

$$S_r = \frac{1}{n}\sum_{i=1}^{n} S_i, \text{where } S_i = 100 \times Pr(D > d_i),$$

where the recipe enhancement score (S_r) ranging from 0 to 100 is the average of ingredient scores (S_i) of ingredients in the recipe, and n is the number of ingredients in the recipe. The ingredient score S_i, which is correlated with the angle distance (d_i) of the given ingredient and the featured ingredient beef, is 100 multiplied by the probability of angle distance in food (D) greater than the calculated angle distance (d_i). The flavor compounds constituents of food ingredients can be found in (Ahn et al. 2011), and the aforementioned probability can be calculated from the empirical distribution of paired ingredients angle distances. While the compound concentration in each ingredient should ideally be taken into account, the lack of systematic data prevents us from exploring their impact in this exercise.

Table 1: Enhancement score of beef stew

Ingredient Combination List	Enhancement Score
beef, cabbage, mushroom, potato, mint, sage, bacon, butter	82
beef, mushroom, shellfish, sage, garlic, ginger, butter	64

We comment that there may be other ways to calculate flavor preference score, such as taking the minimum or maximum of the ingredient scores instead of the mean. The goodness of the approach is open for empirical validation. The key idea of using scientific study of human flavor perception for a computational creativity system is a valid step towards building human-level evaluation models.

2.2 Personalized Novelty Assessment

Creativity is only meaningful when there is a human observer, and each observer's world views, culture, life experience, social network are different, so the perception of novelty which is heavily influenced by these factors are inherently different. A parsnip dish may be common to a European consumer, but may be novel to a Chinese consumer. Therefore we need a personalized novelty assessment specific to a targeted observer or a targeted social group.

Bayesian surprise is proposed to quantify the perceived novelty of a newly created artifact (Varshney et al. 2013). The function measures the change in the observer's belief of known artifacts after observing the newly created artifact, where the belief is characterized by the probability distribution of artifacts. The larger the change is the more surprising or more novel the newly created artifact is.

We adopt the use of Bayesian surprise for personalized novelty assessment, and propose to use Internet activity and social media to construct a personalized set of artifacts known to a given individual or a social group. Then, we calculate a personalized surprise score of the newly created artifact. For example, we can learn recipes and ingredients known to an individual from various websites such as *Pinterest* and *allrecipes.com* by gathering recipes posted, reviewed and pined by the individual and her neighborhood in the social network. We denote the frequency of artifact a at time t known to individual p as $f_a(p,t)$. The weighted frequency ($\tilde{f}_a(p,t)$) of artifacts known to the individual can be calculated by incorporating social proximity and temporal proximity.

$$\tilde{f}_a(p,t) = \sum_{t'<t} w_T(t',t) \times \Big\{ \sum_{p' \in \text{neighbor of } p} w_S(p',p) \times f_a(p,t) \Big\},$$

where $w_T(t',t)$ and $w_S(p',p)$ are inversely related to temporal proximity and social proximity, respectively. Namely, an artifact which was seen long time ago may be forgotten by the individual (Ebbinghaus 1913), and an artifact known to a closer neighbor in one's social network has higher chance to be known by the individual (Mislove et al. 2007).

Although the ontology to define artifacts and data source may be domain specific, the set forth methodological framework can be extended to other creativity domains for personalized novelty assessment.

3 A Search Method for Recipe Generation

Artifact generation is often a pre-cursor to the evaluation process. A common approach is to rely on human responses to evaluate artifacts such as rhythm and pitch combinations (Monteith, Martinez, and Ventura 2012) and visual narratives (Pérez y Pérez, Morales, and Rodriguez 2012). While this approach is sometimes unavoidable, it is clearly not desirable because it is impossible for humans to explore the entire creativity space and evaluate every newly generated artifact. Recently, there has been a growing interest in the computational creativity community to design evaluation mechanisms that are more robust and objective. (Jordanous 2011) proposed an evaluation guideline for creative systems, (Colton 2008) suggested that *how* a creative work is produced is critical to it being perceived (or not) as high valued, and (Agustini and Manurung 2012) evaluated the performance of their riddle creation system by comparing the newly created artifacts with those created by another creativity engine. In culinary creativity evaluation, (Morris et al. 2012) trained an artificial neural network model to evaluate the generated recipes, and (Varshney et al. 2013) proposed a cognitive model motivated by human flavor perception science.

(Boden 1990) proposed that the model of creativity involves a conceptual space and its exploration by creative agents. This conceptual space is a set of artifacts that satisfy certain constraints of the item or idea being generated. (Wiggins 2006) introduced the creative systems framework which revolves around a search mechanism for moving through this conceptual *search* space. For the recipe generation problem, we consider the complexity of creating recipes which may contain 15 or more ingredients. As discussed

by (Varshney et al. 2013), the search space for such problems could be in the scale of quintillions (10^{18}) or more. An intelligent search method is necessary to reduce the computational time and guarantee performance. Towards this end, we argue that evaluation should be a part of the generation/search process and propose an optimization-based approach for the recipe creation problem.

The proposed approach models the three evaluation metrics which were discussed in (Varshney et al. 2013) - novelty assessed using Bayesian surprise, flavor pleasantness and food pairing - as the objective function, and the ingredient requirements, identified through learning about the $\langle cusine, dish, ingredient\rangle$ pairing frequencies from the corpus of recipes, as constraints. We extend their work in this paper by identifying dishes which perform well on all three metrics. That is, we develop a joint generation-evaluation approach to identify top-quality recipes and, additionally, offer a higher degree of confidence on the true quality of the generated recipes. The objective function could be formulated in more than one way - maximizing the average of the three metrics, or $max(min(novelty, flavorpleasantness, foodpairing))$. In this paper, we formulate the problem using the former. However, for purposes of performance comparison, we compute the score for individual metrics. The goal is to find a local maxima (or minima) $\mathbf{X}^*$ in terms of the evaluation metrics in the recipe creation space. Let T be a set of ingredient types, I be the set of ingredients, B be a set of must have ingredients, C_i be the set of chemical compounds in ingredient i, and R be the set of recipes.

Parameters

p_c : pleasantness score of chemical compound c
$\alpha_{i,r}$: count of ingredient i in recipe r for a given dish type in the selected cuisine
q^t_{min} : minimum quantity of ingredient type t
q^t_{max} : maximum quantity of ingredient type t
$P_1(i)$: prior belief of ingredient i appearing in a recipe in the selected cuisine
$P_2(i)$: posterior belief of ingredient i appearing in a recipe in the selected cuisine

Decision Variables

X_i : takes a value of 1 if ingredient i is present in the newly generated recipe and 0 otherwise

$$\max \sum_{i\in I, c\in C_i} X_i p_c + 2 * \frac{\sum_{i,j\in I:i\neq j} X_i X_j \left|C_i \cap C_j\right|}{\sum_{i\in I} X_i(\sum_{i\in I} X_i - 1)} + \int_I P_2(i) \log \frac{P_2(i)}{P_1(i)}$$

s.t.

$$P_1(i) = \frac{\sum_{r\in R} \alpha_{i,r}}{\sum_{r\in R, i\in I} \alpha_{i,r}} \qquad \forall i \in I \tag{1}$$

$$P_2(i) = \frac{X_i + \sum_{r\in R} \alpha_{i,r}}{\sum_{i\in I} X_i + \sum_{r\in R, i\in I} \alpha_{i,r}} \qquad \forall i \in I \tag{2}$$

$$q^t_{min} \leq \sum_{i\in I\cap t} X_i \leq q^t_{max} \qquad \forall t \in T \tag{3}$$

$$X_b = 1 \qquad \forall b \in B \cap I \tag{4}$$

$$X_i \in \{0,1\} \qquad \forall i \in I \tag{5}$$

Constraints (1) and (2) define prior and posterior beliefs of an ingredient i appearing in a certain recipe respectively. Constraint (3) enforces the quantity of each ingredient type that the system determines is required to prepare the selected type of dish. Constraint (4) enforces the quantity of user-defined ingredients in the recipe being designed. For example, B could represent user-specifications such as nutritional and/or regional constraints.

The above formulation results in a non-convex, non-linear optimization model with integer variables. Prior works in computational creativity have applied AI-search inspired methods (Wiggins 2006; Morris et al. 2012; Ritchie 2012; Veeramachaneni, Vladislavleva, and O'Reilly 2012) to search problems. In the optimization literature, researchers have used multiple relaxation approaches including branch and bound, Bender's decomposition (You and Grossman 2013), conjugate gradient or C-G (Dai and Yuan 1999), interior point methods (Vanderbei and Shanno 1999), and genetic algorithms (Morris et al. 2012). Here, we choose a C-G approach to solving this model due to its storage, computational and convergence guarantee advantages (Nocedal and Wright 2006). As a first step, we utilize the following inequalities to introduce approximations and convert it into a convex optimization model.

$$2 * \frac{\sum_{i,j\in I:i\neq j} X_i X_j \left|C_i \cap C_j\right|}{\sum_{i\in I} X_i(\sum_{i\in I} X_i - 1)} \geq 2 * \frac{\sum_{i,j\in I:i\neq j} X_i X_j \gamma_{ij}}{{q^t_{max}}^2} \qquad \forall i,j \in I \cap t \tag{6}$$

$$P_2(i) = \frac{X_i + \sum_{r\in R} \alpha_{i,r}}{\sum_{i\in I} X_i + \sum_{r\in R, i\in I} \alpha_{i,r}} \geq \frac{X_i + \sum_{r\in R} \alpha_{i,r}}{q^t_{max} + \sum_{r\in R, i\in I} \alpha_{i,r}} \qquad \forall i \in I \cap t, r \in R \tag{7}$$

Our solution approach was run on a data set which contains 25,000 recipes available on Wikia. For settings, we chose to prepare a *French soup* containing *beef* as a base ingredient. The C-G algorithm was made to run for three initial solutions (recipes) and four values of convergence limits. The evaluation metrics were averaged over these 12 runs. To compare the performance of our algorithm, we also designed recipes, under the same settings, using a sampling approach. The sampling algorithm adds ingredient types such as vegetables, fruits, meat etc. sequentially to the set of existing ingredients, such that the ingredient constraints, represented by constraint (4) are met. Since the search space is in the order of 10^{18}, after each ingredient type is added, it samples a fixed number of recipes that satisfy the constraint set. Then, the final set of sampled recipes are evaluated on the basis on the three metrics. In other words, the sampling approach adopts a generation followed by an evaluation approach. Table 2 summarizes the performance of the conjugate gradient approach compared to the sampling approach.

From the results shown above, we note that the conjugate gradient approach creates higher quality recipes in the conceptual search space, compared to the sampling approach. In particular, it performs better in learning about the non-linear metrics such as novelty and food pairing, and creating recipes that are better in these aspects.

Table 2: Model Results

Problem instance	Size of search space (x10^{18})	Improvement in novelty (%)	Improvement in flavor pleasantness (%)	Improvement in food pairing (%)
1	9,000	67.86	26.83	55.20
2	600	100.00	12.12	43.82
3	8	50.00	6.90	41.26

4 Discussion

In this paper, we report new developments in culinary computational creativity from two aspects: personalization in the evaluation metrics and optimization in the generation process.

We draw inspiration from the science of human flavor perception for personalized flavor preference. The idea of using principles from scientific study of people, such as psychology, neural and sensory science, may help computational creativity in other domains make progress towards a human level evaluation. There is much information on the Internet for us to learn about an individual or a targeted social group. Although the ontology to define artifacts and data source may be domain specific, such as the personalized novelty assessment for culinary recipe discussed in this paper, utilizing the Internet to gather personalized information is very useful in new product creation where computational creativity can bring business value.

Similar to the recipe generation problem, large search spaces are commonly encountered in many other domains (Thornton 2007). Our optimization-based approach has shown superiority over a sampling approach in recipe creation, and it can easily be extended to other creativity endeavors where evaluation metrics are well defined and formulated. As part of future work, we are currently exploring whether the generation step could also learn from the changes in the evaluation metrics to prune the space of ingredient combinations. This would be quite helpful in speeding up the search process and optimizing memory requirements. Additionally, efforts are underway towards developing theoretical performance guarantees on the quality of the generated recipe to be able to evaluate the performance of the suggested solution algorithm.

References

Agustini, T., and Manurung, R. 2012. Automatic evaluation of punning riddle template extraction. *Proc. Int. Conf. Comput. Creativity*.

Ahn, Y.-Y.; Ahnert, S. E.; Bagrow, J. P.; and Barabási, A.-L. 2011. Flavor network and the principles of food pairing. *SR* 1:196.

Boden, M. 1990. The creative mind. *Abacus*.

Burdock, G. A. 2009. *Fenaroli's Handbook of Flavor Ingredients*. Boca Raton, FL: CRC Press.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. *Proc. AAAI Spring Symp.*

Dai, Y. H., and Yuan, Y. 1999. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM J. Optim.*

Ebbinghaus, H. 1913. *Memory: A Contribution to Experimental Psychology*. Teachers College, Columbia University.

Haddad, R.; Medhanie, A.; Roth, Y.; Harel, D.; and Sobel, N. 2010. Predicting odor pleasantness with an electronic nose. *PLOSCB* 6(4):e1000740.

Jordanous, A. 2011. Evaluating evaluation: Assessing progress in computational creativity research. *Proc. Int. Conf. Comput. Creativity* 102 – 107.

Khan, R. M.; Luk, C.-H.; Flinker, A.; Aggarwal, A.; Lapid, H.; Haddad, R.; and Sobel, N. 2007. Predicting odor pleasantness from odorant structure: Pleasantness as a reflection of the physical world. *JNEU* 27(37):10015–10023.

Mislove, A.; Marcon, M.; Gummadi, K. P.; Druschel, P.; and Bhattacharjee, B. 2007. Measurement and analysis of online social networks. *IMC '07 Proc. ACM SIGCOMM Conf. Internet Measurement* 29–42.

Monteith, K.; Martinez, T.; and Ventura, D. 2012. Automatic generation of melodic accompaniments for lyrics. *Proc. Int. Conf. Comput. Creativity* 87 – 94.

Morris, R. G.; Burton, S. H.; Bodily, P. M.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. *Proc. Int. Conf. Comput. Creativity*.

Nocedal, J., and Wright, S. J. 2006. *Conjugate gradient methods*. Springer, New York.

Pérez y Pérez, R.; Morales, N.; and Rodriguez, L. 2012. Illustrating a computer generated narrative. *Proc. Int. Conf. Comput. Creativity*.

Ritchie, G. 2012. A closer look at creativity as search. *Int. Conf. Comput. Creativity* 41 – 48.

Sawyer, R. K. 2012. *Explaining Creativity: The Science of Human Innovation*. Oxford: Oxford University Press.

Shepherd, G. M. 2006. Smell images and the flavour system in the human brain. *Nature* 444(7117):316–321.

Snitz, K.; Yablonka, A.; Weiss, T.; Frumin, I.; Khan, R. M.; and Sobel, N. 2013. Predicting odor perceptual similarity from odor structure. *PLOSCB* 9(9):e1003184.

Thornton, C. 2007. How thinking inside the box can become thinking outside the box. *Proc. Int. Joint Workshop Comput. Creativity* 113 – 119.

Vanderbei, R. J., and Shanno, D. F. 1999. An interior-point algorithm for nonconvex nonlinear programming. *Comput. Optim. Appli.*

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Bhattacharjya, D.; Schöergendorfer, A.; and Chee, Y. M. 2013. A big data approach to computational creativity. arXiv:1311.1213.

Veeramachaneni, K.; Vladislavleva, E.; and O'Reilly, U.-M. 2012. Knowledge mining sensory evaluation data: genetic programming, statistical techniques, and swarm optimization. *GPEM* 13(1):103–133.

Wiggins, G. A. 2006. Searching for computational creativity. *NGC* 24(3):209–222.

You, F., and Grossman, I. E. 2013. Multicut Benders decomposition algorithm for process supply chain planning under uncertainty. *Ann. Oper. Res* 191 – 211.

Exploring Application Domains for Computational Creativity

Ashish Jagmohan, Ying Li, Nan Shao, Anshul Sheopuri, Dashun Wang
IBM TJ Watson Research Center, Yorktown Heights, NY
Lav R. Varshney
University of Illinois, Urbana-Champaign, IL
Pu Huang
Marvelous Face, Inc., NY

Abstract

We are motivated by the recent application of computational creativity in the culinary domain. Given the increasing commercial importance of data-driven computation, we explore and provide unified framework in three new domains to which computational creativity can be applied and yield business value. The three domains are travel, fashion, and science. Reflecting on the framework characterization, we identified two properties common across these domains, related to the creative space and codified domain knowledge. We believe that these properties may have value as sufficient, but not necessary, conditions to identify domains suitable for industrializing computational creativity. We are working towards finding tight properties common across different domains as well as ones that exclude domains.

1 Introduction

Computational creativity is the study of how computers can create, or help create artifacts that humans perceive to be creative. The field attempts to better understand human creativity and design programs that can enhance human creativity, bringing together ideas from artificial intelligence, cognitive psychology, design, philosophy and the arts. An overview of this field can be found in (Colton and Wiggins 2012).

A recent attempt in culinary computational creativity ((Varshney et al. 2013; Pinel and Varshney 2014)) motivates the work presented in this paper. The system described in that study can create novel and flavorful recipes as perceived by people. It gathers data from culinary science as well as other domains such as hedonic psychophysics, establishes evaluation metrics based on codified expert knowledge in recipe design and human flavor perception, and can create quintillions of recipes which are far beyond the number of existing recipes. We describe this work in Section 2.

Given the increasing commercial importance of data-driven computation, we explore whether a similar design framework can be extended and bring business value to other domains which are analogous to the food domain in terms of compositional models for artifacts. Focusing on fashion, travel, and science, we describe how a computational creativity framework can be developed in Section 3.

In Section 4, we reflect upon the framework, and argue that there are two properties which appear to be common across the aforementioned domains, and they are related to the combinatorial complexity of the domain creation space, and the state of the codified domain knowledge respectively.

In this preliminary position paper, we argue that while these two properties may not be necessary conditions to identify domains wherein computational creativity yields business value, they are sufficient to provide a general framework for computational creativity in industrial deployment. We are working towards finding tight properties as well as ones that exclude unsuitable domains.

2 Culinary Creativity: Case Study

Culinary design has long been seen as a creative domain in the history of human creativity research. "Made up a recipe" is one of the 100 creative activities listed on the first human creativity rating questionnaire developed by Torrance in 1962 (Sawyer 2012). But can a computer be creative for culinary recipes? With the availability of large-scale online recipe repositories in recent years, some recipe design principles have been validated using a data-driven approach, such as the food pairing hypothesis (Ahn et al. 2011). In addition, human flavor perception is gradually being uncovered by advanced scientific study of food chemistry, hedonic psychophysics and neurogastronomy (Haddad et al. 2010; Shepherd 2006). These efforts have made computational creativity possible for generating novel and flavorful recipes.

In fact, (Morris et al. 2012) discussed a recipe creation system which was restricted to soups, stews and chili. The more recent culinary computational creativity system (Varshney et al. 2013; Pinel and Varshney 2014) is more general and has a cognitive flavor assessment component motivated by the scientific study of human flavor perception. The recipes created by this system have been served in multiple venues and have been well received. An independent assessment done by *Wired* (Davis 2013) of a recipe created by the system concluded that “while the IBM dessert tasted better, it was also insanely elaborate, so we’ll call it a draw."

We briefly describe the system here and characterize the culinary domain to understand why and how computational creativity brings value to this domain. The system used Wikia recipes as an inspiration set (around 25,000 recipes), and it can produce quadrillions or even more newly created recipes. Generally speaking, the volume of existing recipes from recipe repositories is usually around tens of thousands, possibly up to millions. So the inspiration set is large enough

for us to draw prior information. The dimensionality of the culinary domain is captured by the number of possible ingredients which is in the range of hundreds or thousands. Looking at the combinatorial complexity of recipes which may contain 10-20 ingredients, the creativity space can be in the scale of quintillions, a much larger number than the size of the existing recipes. There is plenty of room for creativity in generating novel artifacts.

As mentioned before, there exists codified knowledge in the scientific study of human flavor perception and culinary design principles. The system gathers data and organizes such knowledge in a structured model and therefore can provide quality and novelty assessment (pairing, pleasantness and surprise) of the creative artifacts in such a large design space that people would not be able to do so. Indeed, tracking so many ingredients and reasoning their combinations to build quintillions of ideas are only feasible in a computational creativity system.

3 Application Domains

Having developed and deployed a system in the trillion-dollar food industry, are there other domains where computational creativity can bring business value? There should be enough room for creativity in a domain so that creating new artifacts is more valuable than searching among existing ones. There should also be codified knowledge for a computer to learn in a structured way in order to establish formalized predictors for creativity in terms of quality and novelty, so that a human expert can gain support from the computational system rather than relying only on the intrinsic human expertise. Following these thoughts, we explore three domains in this section: travel, fashion and scientific discovery. The results are summarized in table 1.

3.1 Travel

The advent of large-scale online networks over the last two decades has affected a fundamental transformation of how the travel sector interacts with and sells to customers. Although online travel sales now account for nearly $100 billion, there is a high dissatisfaction rate (53%) among customers, and while most focus is on price competition, little concern is given to the added value that digital channels can bring to customers (Carey, Kang, and Zea 2012; Peterson 2011).

While no comprehensive personalized travel planning solution exists, there are websites and apps which leverage social and mobile modalities to facilitate travel planning. These include informational websites such as TripAdvisor and Fodors, niche websites such as Flextrip, and itinerary planning and organization websites and mobile apps such as TripIt, Plannr, and mTrip. There is also some prior research on using collaborative filtering to recommend travel packages to tourists (Liu et al. 2014).

It is our view that the travel domain offers a promising opportunity for computational creativity to drive business value. The expected artifact produced by the system is a travel "experience", which is a sequence of activity/time-range pairs. Here, an activity may denote visiting a specific destination such as a cultural artifact, or a physical activity such as taking a specific tour. A time-range is the time period over which that activity is to be carried out. Designing a travel experience is a search in a high-dimensional space defined by the Cartesian product of possible activities and time-ranges. The inspiration set can be existing travel packages, or itineraries culled from social media as suggested, for example, in (Lempel et al. 2014).

The high dimensionality of the travel experience space, and the many possibilities of extracting inspiration itineraries, combine to make ample room for creativity. As witnessed by the relatively low satisfaction rates of current travelers, the design of a personalized and comprehensive travel experience is a creative endeavor with non-trivial difficulty. Further, there exists rich domain literature studying the issue of travel satisfaction from psychological and sociological perspectives (del Bosque and Martin 2008). Thus the gap between codified knowledge and intrinsic customer expertise is significant. We believe that computational creativity is uniquely suited to designing satisfying travel experiences, by marrying computationally-intensive learning from big data with expert insights.

We propose two chief metrics of goodness. Personalized novelty measures how different an experience is from prior itineraries and the user's prior travel experiences. Travel satisfaction measures how likely the user is to be satisfied at the end of the travel experience, and requires codified domain knowledge to compute. An example of such domain knowledge is the cognitive-affective model for travel satisfaction derived and validated in (del Bosque and Martin 2008). The key finding is that overall travel satisfaction (and travel loyalty) is driven by an interplay of cognitive and emotional aspects, including destination image, trip expectations, and positive and negative emotions accumulated during the trip. Specifically, high travel satisfaction is driven by positive expectations which are disconfirmed positively during the trip. This understanding points to how a computational creative system can design travel experiences so as to maximize personalized satisfaction, by leveraging a user's personal notions of destination image and expectations.

3.2 Fashion

Creating fashion artifacts is challenging, both due to the fact that there are many factors to weigh in (such as fashion style, color and fabric) and many design options (such as pockets and belts) to play with. In addition, even without taking the design aspect into account, creating good and tasteful outfits from a set of given clothing articles and subsequently ranking them based on certain criteria is a challenging problem. In this section, we discuss and formulate this problem with a focus on outfit creation based on individuals' wardrobes.

Consider a wardrobe containing clothing articles: we need to find an outfit with a combination of clothing articles that meets particular requirements; the goal is to create an outfit that is both aesthetically pleasing and satisfying. Equivalently, we can generate a list of outfits then rank them based on certain metrics. There has been some prior work on this front. For example, Lin *et al.* (Lin et al. 2012) described a personalized clothing recommendation system

Table 1: Characterized Domains

	Culinary	Travel	Fashion	Scientific Discovery
Output Artifacts	**Recipe**: Mixture of ingredients.	**Travel experience**: Sequence of (activity, time-range) pairs.	**Dress**: A set of outfits that are aesthetically pleasing.	**Hypotheses**: A set of existing literatures.
Volume of Inspiration Set	Existing recipes from recipe repositories.	Existing itineraries, from travel packages or social media.	Existing examples of aesthetic/stylish dress examples.	Pool of concepts and relations (published connections).
Dimensionality	High: Ingredients.	High: Activity$\times$ Time.	High: Top $\times$ Bottom $\times$ Any additional layers.	High: trivial and non-trivial combinations.
Metrics of Goodness	**Surprise**: Difference from inspiration recipes. **Pleasantness**: Likely pleasantness of recipe.	**Novelty**: Against inspiration itineraries, user experience. **Satisfaction**: Likely user satisfaction.	**Surprise**: Style difference against personal inspirations. **Aesthetics**: Color and pattern matching.	**Impact**: how many citations a certain combination of concepts may receive?
Codified Expert Knowledge	Principles of flavor pairing. Principles of pleasantness.	Psycho-social principles of travel satisfaction.	Fashion design, color science, psycho-social dress principles.	Metaknowledge, Swanson hypotheses.

based on a modified Bayesian network. Specifically, the system constructs the outfit by first selecting a top, then a bottom which matches the selected top. Another related work is proposed by Shen *et al.* (Shen, Lieberman, and Lam 2007), where each clothing item is first labeled with brand, type and a sentence to describe its style; and then the user tells the system about a particular occasion in her mind; finally, based on commonsense reasoning, the system matches the clothes' styles and functions with the concepts needed for the context, and returns suggestions for complete outfits.

In the outfit creation problem, the inspiration set contains existing dress examples that are aesthetically pleasing, and one possible data source is the individual's photo album. In this case, the inspiration set is large enough for a meaningful outfit creation, while not so large compared to the combinatorial space of dress artifacts. Moreover, we can leverage specialized principles in fashion design, color science, and even psychology and sociology. There is significant prior literature that codifies such knowledge, and a significant gap exists between expert knowledge and individual knowledge.

A complete personalized outfit creation system consists of five components: 1) *catalog of personal wardrobe*, which records an individual's wardrobe including both the clothing articles and their features (*e.g.*, the garment type, color, pattern, fabric, brand, *etc.*); 2) *personal needs or requirements collection*, examples of which are specific dressing occasions (e.g. evening party or daily work), context information (weather, season), and user profession and age; 3) *outfit creation strategy*, which determines how to generate the list of outfits. Both sequential and integrated approaches can be applied here. Specifically, the sequential approach creates an outfit by selecting the needed clothing articles piece by piece based on certain criteria. In contrast, the integrated approach learns "good" outfit examples from existing knowledge and creates an outfit as a single artifact that it deems "good"; 4) *ranking metrics*, which will be applied to rank the generated outfits based on formal design principles such as color and pattern matching, and novelty value; and 5) *system evaluation*, probably best conducted through a user study.

3.3 Genesis of Scientific Hypotheses

Over the past years, the exponential expansion in knowledge is changing the landscape of science, representing both pressing challenges and exciting opportunities. Indeed, the volume of scientific papers has increased to the extent that no individual can read all papers within a field.

We take one recent study in the context of biomedical chemistry (Foster, Rzhetsky, and Evans 2013) as an exemplary case to illustrate the process of applying computational creativity to generating scientific hypothesis. The first challenge is to define the artifacts and the items within each artifact. This corresponds to defining the underlying space of possible search paths and conceptual entities within the space. The network of scientific knowledge proposed by (Girvan and Newman 2002) and taxonomy of research strategies building on top of this network (Foster, Rzhetsky, and Evans 2013) provides a promising direction for constructing such a conceptual space with semantic entities. For example, Foster *et al.* analyzed 6.5 million abstracts in biomedicine and biomedical chemistry to construct a network of relations between chemicals. One can use this network as a representation of knowledge, hence each artifact corresponds to a study into the relationship between chemicals, with items being chemicals involved in the study (Evans and Foster 2011). On another coarse-grained level, applying community detection algorithms to this network yields knowledge clusters, corresponding to tightly related concepts. In this view, items within each artifact are represented by knowledge clusters. A key insight in this process comes from citations. As citations are often taken as proxies of impact, one can study how and why certain combination of conceptual entities within the knowledge representation would generate artifacts with higher impact.

Taken together, computational genesis of high-quality sci-

entific hypotheses is an active and promising line of inquiry, mainly following two directions. On one hand, there have been a number of fascinating studies into clever mechanisms of combining existing knowledge. Besides biomedical chemistry, there are also literature-based discovery methods pioneered by (Swanson 1987) and more recently combination of novelty and conventionality through co-citation pairs by (Uzzi et al. 2013).

4 Discussion

Reflecting on the computational creativity framework developed in Section 3, we find that there are two common properties across these domains. The first property is related to the combinatorial complexity of the creation space and its relation to the number of extant inspiration artifacts. On one hand, the size of the inspiration set is suitably large for a data-driven approach to learn basic cultural principles of the domain. On the other hand, the full combinatorial creation space is significantly larger than the inspiration set, so that creating new artifacts is more valuable than searching among existing ones. The second property is about the cognitive difficulty of evaluating artifacts. Codified knowledge exists and can be learned by computer, and therefore data-driven predictors of novelty and domain-appropriateness can be deployed for evaluation and selection of ideas. In this case, there is a significant computable knowledge asymmetry in favor of a computational creativity system than a human expert with intrinsic expertise, that computers can quickly access more knowledge than human creators.

A foundation of creativity is knowledge, and codified knowledge exists for many domains. A computationally creative system needs to effectively and efficiently represent, manipulate, and reason with such codified knowledge in application domains. Organizing such knowledge into a well-structured scheme or model may not, however, be easy. Identifying domains of industrial importance where there is an ability to learn about parts and combining rules from examples is therefore crucial.

Exploring the whole creation space is possible for some application domains but for many others, this space is combinatorially large. For such cases, we need creativity metrics to carve out the space for efficient selection, though finding good heuristic metrics can be a process of trial-and-error, and as much art as science. Principles from psychology, however, provide a good starting point.

We believe the two properties discussed here to be sufficient conditions, but not necessarily necessary conditions, to identify domains suitable for computational creativity in industrial deployment. We are working towards finding tight properties common across different domains, as well as ones that exclude domains.

References

Ahn, Y.-Y.; Ahnert, S. E.; Bagrow, J. P.; and Barabasi, A.-L. 2011. Flavor network and the principles of food pairing. *Sci. Reports* 1:196.

Carey, R.; Kang, D.; and Zea, M. 2012. The trouble with travel distribution. *Mckinsey and Company, Insights and Publications*.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier. *Proc. Euro. Conf. A. I.*

Davis, A. P. 2013. Digital gastronomy: When an IBM algorithm cooks, things get complicated–and tasty. *Wired* 20(10).

del Bosque, I. R., and Martin, H. S. 2008. Tourist satisfaction a cognitive-affective model. *Ann. Tour. Res.* 551–573.

Evans, J. A., and Foster, J. G. 2011. Metaknowledge. *Science* 331(6018):721–725.

Foster, J. G.; Rzhetsky, A.; and Evans, J. A. 2013. Tradition and innovation in scientists' research strategies. *arXiv:1302.6906*.

Girvan, M., and Newman, M. E. 2002. Community structure in social and biological networks. *Proc. Nat. Acad. Sci.* 99(12):7821–7826.

Haddad, R.; Medhanie, A.; Roth, Y.; Harel, D.; and Sobel, N. 2010. Predicting odor pleasantness with an electronic nose. *PLoS Comput. Biol.* 6(4):e1000740.

Lempel, R.; Golbandi, N.; Choudhury, M.; Feldman, M.; Amer-Yahia, S.; and Yu, C. 2014. Automatic construction of travel itineraries using social breadcrumbs. *ACM Hypertext*.

Lin, Y.; Kawakita, Y.; Suzuki, E.; and Ichikawa, H. 2012. Personalized clothing recommendation system based on a modified Bayesian network. *IEEE Int. Symp. Appl. Internet* 414–417.

Liu, Q.; Chen, E.; Xiong, H.; Ge, Y.; Li, Z.; and Wu, X. 2014. A cocktail approach for travel package recommendation. *IEEE Trans. Knowl. Data Eng.* 278–293.

Morris, R. G.; Burton, S. H.; Bodily, P. M.; and Ventura, D. 2012. Soup over beans of pure joy: Culinary ruminations of an artificial chef. *Proc. Int. Conf. Comput. Creati.* 119–125.

Peterson, S. 2011. Travel 2020: The distribution dilemma. *IBM Institute for Business Value*.

Pinel, F., and Varshney, L. R. 2014. Computational creativity for culinary recipes. *Proc. ACM CHI Conf. on Human Factors in Computing Systems*. To appear.

Sawyer, R. 2012. *Explaining Creativity: The Science of Human Innovation*. Oxford: Oxford University Press.

Shen, E.; Lieberman, H.; and Lam, F. 2007. What am i gonna wear?: Scenario-oriented recommendation. *International Conference on Intelligent User Interfaces* 365–368.

Shepherd, G. M. 2006. Smell images and the flavor system in the human brain. *Nature* 444(7117):316–321.

Swanson, D. R. 1987. Migraine and magnesium: eleven neglected connections. *Perspectives in biology and medicine* 31(4):526–557.

Uzzi, B.; Mukherjee, S.; Stringer, M.; and Jones, B. 2013. Atypical combinations and scientific impact. *Science* 342(6157):468–472.

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Bhattacharjya, D.; Schoergendorfer, A.; and Chee, Y.-M. 2013. A big data approach to computational creativity. arXiv:1311.1213.

Towards Evolutionary Story Generation

Andrés Gómez de Silva Garza
Computer Engineering Department
Instituto Tecnológico Autónomo de México (ITAM)
Río Hondo #1, Colonia Progreso Tizapán, 01080—México, D.F. México
agomez@itam.mx

Rafael Pérez y Pérez
Departamento de Tecnologías de la Información, División de Ciencias de la Comunicación y Diseño
Universidad Autónoma Metropolitana, Cuajimalpa
México, D.F. México
rperez@correo.cua.uam.mx

Abstract

In this paper we describe on-going work on combining two existing models of computational creativity. The GENCAD model proposes the use of an evolutionary algorithm (EA) that uses a population of exemplars as a starting point for its search, unlike traditional EA's, which use a randomly-generated initial population. The EA, operating on this population, is then used to generate new potentially creative solutions. GENCAD has been instantiated in the domains of structural design of tall buildings and feng shui-compliant residential floor-plan design. The MEXICA model also begins with a set of exemplars as a starting point, but it analyzes these exemplars based on a domain theory. The general theory that is obtained from analyzing the set of exemplars is then used to guide the generation of new solutions. MEXICA has been instantiated in the domain of plot generation for stories involving themes, characters and locations from the Mexica culture of ancient Mexico. In the hybrid model we propose in this paper, we combine the two models to generate plots for stories of the same sort that MEXICA generates, but using GENCAD's process model to do so.

Introduction

We have begun work on combining two existing models of computational creativity which were developed independently. The purpose of this combination is to produce a hybrid model that maximizes the advantages and minimizes the disadvantages of both original models.

The first of the original models we are working with is GENCAD (Gómez de Silva Garza 2000). The generative module in GENCAD takes a set of pre-existing exemplars of the type of thing that we would like to create and interprets it as the initial population of an evolutionary algorithm (EA) (Mitchell 1998). The EA's genetic operators are then used to generate complete new potential solutions (new examples of the type of thing that we would like to create). The EA's evaluation module uses domain and common-sense knowledge to assign a fitness value to each of these new solutions that serves to rank the old and new solutions so that only the best solutions survive across evolutionary generations. Convergence of the EA occurs when one of the new potential solutions is determined to be of sufficient quality according to both the EA's fitness function and whatever initial problem requirements the user may have specified. This process model has been instantiated in two domains: the structural design of tall buildings (Gómez de Silva Garza and Maher 1998), and the design of residential floor plans that follow the principles of feng shui (Gómez de Silva Garza and Maher 1999). In these two instantiations the set of exemplars that is used as the initial population of the EA results from an earlier phase of the process model which implements the case retrieval stage of a case-based reasoner, whereas the EA implements the case adaptation stage (Leake 1996).

The second of the original models we are working with is MEXICA (Pérez y Pérez 1999). The generative module in MEXICA takes a set of pre-existing exemplars of the type of thing that we would like to create and analyzes it according to a domain theory. The domain in which MEXICA has been instantiated is the generation of plots for stories involving themes, characters, and locations from the Mexica culture of ancient Mexico (Pérez y Pérez and Sharples 2001; Pérez y Pérez 2007), though some initial work has been done on applying the model to image layout design (Pérez y Pérez et al. 2012). The theory used by MEXICA for this domain is based on an analysis of the emotional links between characters and the flow (increase and decrease) of tensions in a story as actions take place in the story. As a result of analyzing the set of exemplars according to this domain theory, MEXICA produces an abstract description of the entire set. Given an initial

action, it then starts to add more and more actions using the abstract description as a set of guidelines that ensure coherence, thus eventually producing a complete story piecemeal.

As can be seen, there are similarities between GENCAD and MEXICA, yet there are also quite a few, sometimes subtle, differences. In the next section of the paper we discuss these issues further, as well as the advantages and disadvantages of the two process models from the point of view of computational creativity. In the section after that we describe our hybrid model and some of its characteristics. Finally, in the last section we provide some results, lessons, and observations from our preliminary experiments with our hybrid model.

Comparing and Contrasting GENCAD and MEXICA

In this section we compare and contrast the evolutionary approach used by GENCAD with the theory-based approach used by MEXICA for the generation of solutions to computational creativity problems. This is done by analyzing some of the characteristics of the two approaches and their relative advantages and disadvantages.

One of the characteristics of evolutionary approaches is that the way in which they generate new potential solutions is generally syntactic rather than semantic. Existing genotypes are tweaked (by the mutation operator) or split and spliced in order to combine their characteristics (by the crossover operator) without any prior analysis of whether the results will "make sense" or not, or of the meaning of the genotypes. This analysis is left to the EA's evaluation module later on in the process. This means that the generative module is generally not biased or guided by any domain knowledge, thus increasing the potential for interesting, unexpected features in the generated solutions, an important characteristic in creativity (Grace and Maher 2014).

Another characteristic of evolutionary approaches is that many of the decisions in the generative module are made at random, such as, in the case of the crossover operator, which genotypes will be combined or where exactly they will be split (before splicing the resulting pieces to produce the resulting new genotypes). Thus even if the same algorithm is run again on the same initial population, the results are not likely to be the same as in previous runs. This unpredictability is another potential source of unexpectedness in the generated solutions. This characteristic also implies that, if for some reason convergence isn't reached during one attempt to process a given initial population, the attempt can be abandoned and a new attempt initiated, with the possibility that the new attempt will converge, thus providing the approach with more flexibility than traditional algorithms possess.

On the other hand, there are disadvantages to evolutionary approaches, which include the following. First, even if convergence is reached (that is, even if eventually a solution that is "good enough" is produced by the evolutionary process), most of the time many bad quality potential solutions may have had to be generated, through a large number of evolutionary generations, before convergence. In addition, even if the capability to "give up" (in order to re-start the evolutionary process to try again) is programmed into the EA, this usually has to be done after a large number of evolutionary generations in order to take into account the slow speed of evolution. In other words, EA's are generally not very efficient.

One of the characteristics of theory-based approaches is that the solutions that are generated are guaranteed from the first to "make sense" (unless the theory is deficient in some way, e.g., incorrect or incomplete). Thus, finding a solution that is "good enough" does not require wasting time on slowly discarding many more defective solutions that were also generated, which is what happens in an EA.

On the other hand, the solutions that are generated are always based on the theory, and by definition will never contain features that go beyond that theory. The constraints imposed by the theory may be too rigid to permit that spark of interestingness or unexpectedness that can be so important in creativity.

Further discussion of these issues involving theory-based approaches is included in the following section.

Hybrid Model Combining the Evolutionary- and Theory-Based Approaches

In order to combine the advantages of the evolutionary- and theory-based approaches to generating solutions for computational creativity systems, we have produced a hybrid model which we describe in this section. We are still in the process of instantiating this model in the domain of story generation.

Our hybrid model, like both GENCAD and MEXICA, begins with a set of exemplars. Following GENCAD's process model, these exemplars are treated as the initial population of an EA whose genetic operators are then used to generate new potential solutions. In our hybrid model the EA's evaluation module is implemented using a looser version of MEXICA's theory combined with common-sense constraints. Thus, some aspects of MEXICA's domain theory are used to guide the generative process and filter out the more deficient solutions, but the rigidity imposed by exclusively following the constraints imposed by the original theory when generating new solutions is counteracted by the flexibility introduced by the genetic operators.

Specifically for the domain of story generation, assuming that MEXICA's standard set of 7 pre-existing stories is used, part of the general, abstract description (theory) it would come up with after analyzing these 7 stories would be that if a character A likes a character B, and B likes another character C, then A becoming jealous of C is a possible next action to introduce to the story being generated. This description arises from the fact that this type of situation (sequence of story actions) occurs in the pre-existing stories. In fact, unless there are multiple other

possible next actions that can be introduced at a given point in MEXICA's creation of a story in which this observation is relevant, A becoming jealous of C will be the next action that will be introduced. In our new hybrid model, a story in which A becomes jealous of C shortly after it is stated that A likes B and B likes C will be assigned a higher fitness value than one in which it happens much afterward, and an even higher fitness value than one in which it doesn't happen at all. But these other possibilities are still present, thus increasing the variety in the structure of the new potential stories that can be generated.

The hybrid system's evaluation module also incorporates knowledge about the flow of dramatic tensions in "good" stories (the tension usually increases steadily up to a certain point, near the end, when there is usually a denouement during which all of the accumulated conflict and tension is resolved) as well as other aspects of MEXICA's domain theory. However, it turned out to be necessary to implement additional common-sense domain constraints in the evaluation module that never had to be represented explicitly in the original instantiations of MEXICA.

For instance, the flexibility of the genetic operators implies that, after several evolutionary generations, new stories that have "incestuous" ancestry may be created. Thus, if AB is a story created in generation 1 whose direct ancestors are A and B, then in generation 2 a new story ABA may be created whose direct ancestors are AB and A, thus containing some genetic material directly inherited from A and some genetic material indirectly inherited from A through AB. This may result in stories in which the sequence of actions is, for instance:

M H K L K M P

In other domains the potential repetitiveness inside a genotype (M and K appear twice in the example sequence given above) may not be important, or may even be desirable—it all depends on the interpretation of the contents of a genotype and on the application domain. However, in story generation the quality of a story is diminished if the author constantly repeats things that have already been stated instead of moving forward with new actions/events. Thus, our hybrid system takes this potential repetition into account when assigning a fitness value to the stories it generates. Further work is still being performed in order to identify which additional such common-sense constraints may be necessary, and in order to implement them in the fitness evaluation module.

Discussion, Results, and Lessons Learned

We have presented a hybrid model of computational creativity that combines aspects of two previously-existing models. The hybrid model uses an evolutionary algorithm (EA) for the generation of solutions, and implements the EA's evaluation module based on a domain theory arising from an analysis of exemplars of good solutions in the application domain. We have instantiated this hybrid model in the domain of story generation in order to test and refine our ideas.

Some work has been done in the past on using EA's for linguistic creativity, but has focused on sentence (Vrajitoru 2003) or poetry (Manurung 2003) generation, rather than story (plot) generation. More similar to our work is (McIntyre and Lapata 2010), though unlike us they do not avoid the use of domain knowledge in the generation module of the EA.

While our work is still preliminary, one of the results we have been able to obtain from this research is to be able to state explicitly the advantages and disadvantages of the original models by comparing and contrasting them. This analysis is what led to our proposal for the hybrid model, which tries to maximize the combined advantages and minimize the combined disadvantages of the original models.

Acknowledgements

This work has been partially supported by Asociación Mexicana de Cultura, A.C. and by the National Council of Science and Technology in México (CONACYT), project number: 181561.

References

Gómez de Silva Garza, A. 2000. *An Evolutionary Approach to Design Case Adaptation*. Ph.D. Dissertation, The University of Sydney, Australia.

Gómez de Silva Garza, A., and Maher, M.L. 1998. A Knowledge-Lean Structural Engineering Design Expert System. *Proceedings of the Fourth World Congress on Expert Systems*, Mexico City, Mexico. 178-185.

Gómez de Silva Garza, A., and Maher, M.L. 1999. Evolving Design Layout Cases to Satisfy Feng Shui Constraints. *Proceedings of the Fourth Conference on Computer Aided Architectural Design in Asia (CAADRIA-99)*, Shanghai, People's Republic of China. 115-124.

Grace, K., and Maher, M.L. 2014 (to appear). What to Expect When You're Expecting: The Role of Unexpectedness in Computationally Evaluating Creativity. *Proceedings of the Fifth International Conference on Computational Creativity (ICCC '14)*, Ljubljana, Slovenia.

Leake, D.B. 1996. *Case-Based Reasoning: Experiences, Lessons, & Future Directions*. Menlo Park, California & Cambridge, Massachusetts: AAAI Press & The MIT Press.

Manurung, H.M. 2003. *An Evolutionary Algorithm Approach to Poetry Generation*. Ph.D. Dissertation, The University of Edinburgh, Scotland.

McIntyre, N. and Lapata, M. 2010. Plot Induction and Evolutionary Search for Story Generation. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden. 1562-1572.

Mitchell, M. 1998. *An Introduction to Genetic Algorithms (Complex Adaptive Systems Series)*. Cambridge,

Massachusetts: MIT Press.

Pérez y Pérez, R. 1999. *MEXICA: A Computer Model of Creativity in Writing*. Ph.D. Dissertation, University of Sussex, United Kingdom.

Pérez y Pérez, R. 2007. Employing Emotions to Drive Plot Generation in a Computer-Based Storyteller. *Cognitive Systems Research* 8(2): 89-109.

Pérez y Pérez, R.; Morales, N.; and Rodríguez, L. 2012. Illustrating a Computer Generated Narrative. *Proceedings of the Third International Conference on Computational Creativity (ICCC '12)*, Dublin, Ireland. 103-110.

Pérez y Pérez, R., and Sharples, M. 2001. MEXICA: A Computer Model of a Cognitive Account of Creative Writing. *Journal of Experimental and Theoretical Artificial Intelligence* 13(2). 119-139.

Vrajitoru, D. 2003. Evolutionary Sentence Building for Chatterbots. *Late Breaking Papers, Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '03)*, Chicago, Illinois. 315-321.

Arts, News, and Poetry — The Art of Framing

Oskar Gross and **Jukka M. Toivanen** and **Sandra Lääne** and **Hannu Toivonen**
Department of Computer Science and Helsinki Institute for Information Technology HIIT
University of Helsinki, Finland
oskar.gross@cs.helsinki.fi, jukka.toivanen@cs.helsinki.fi, sandralaane@gmail.com, hannu.toivonen@cs.helsinki.fi

Abstract

This paper presents an art project which combines computational and human creativity. The paintings created during the project visualize a process of generating computational poetry from daily news stories. We describe how the computational processes of generating poetry were visualized and then turned into paintings by an artist. The project has been exhibited in Finland and Estonia. The feedback collected during the exhibition in Finland is also included in the paper.

Introduction

In this paper we introduce the art project *Arts, News, and Poetry* which combines human and computer creativity in a novel way. First, the computer carries out a creative process of poetry writing and produces an abstract image based on the process. The human artist then takes the poems and images as inspiration and paints them. Our motivation is to direct the audience's attention to the possibility that the inner workings of computers could be visualized and presented in some meaningful and aesthetically pleasing way.

From a computational creativity perspective we aim to introduce the possibility to use the computational processes to provide framing information for creative artefacts. The framing information is often presented to the art consumer in a natural language. In this paper, we explore an alternative approach where the information is expressed in a way which is more natural for computers.

In the rest of the paper, we first extend on the ideas behind framing, then give a brief overview of the art project as a whole. Next, we describe the poetry generation, process visualization, painting of the images and exhibitions. Then we present the related work and conclude the paper.

Framing

The way how artists explain their work has a very large influence on how the audience perceives them. A work of art might even have a completely different interpretation if we change parts of the framing information. For instance, Salvador Dalí's works with phallic symbols might have different meanings if he lived in different cultural context. Charnley et al. (2012) define the framing information as the "motivation, intention and processes involved in creating a work". Currently, the framing information produced in computational creativity tasks incorporates information which is very humane. For instance, the Full-FACE poetry generator tells which news stories it analysed, what kind of affective words it found from there, and how it influenced the output (Colton, Goodwin, and Veale 2012).

Computers have an inhumane ability to memorize every step they make to reach a solution. We argue that this inherent feature could be taken advantage of by the computers in order to provide framing information. In this paper, we have provided a very simplistic (or even naïve) approach for solving this problem. In the ideal case, wouldn't it be interesting if the computer could visualize solving an optimization problem illustrating the drama of constantly reaching a local optimum, no matter how hard it tries?

Overview of the Art Project

The end result is a series of hand-crafted paintings, each visualizing the poem writing process of a computer and exhibited together with the computer-written poem. The topics of the poems were chosen to be based on news stories, so they could be seen as commentary to the events of the world.

The art project consists of the following steps. Steps 3–6 are further elaborated on later in the paper.

1. From 1 to 31 December 2012 we collected news stories from *BBC*, *CNN*, *Reuters*, *ABC News*, *CBS News* and *The Guardian* by automated crawling.
2. The news stories of each day were automatically clustered into 50 different topics. For clustering we used the *gensim* (Řehůřek and Sojka 2010) implementation of LDA (Latent Dirichlet Allocation) (Blei, Ng, and Jordan 2003).
3. For each topic the computer generated a topic-related poem using the methods proposed by Toivanen et al (2012; 2014) (Section *Corpus-Based Poetry Generation*).
4. For each topic an additional abstract image was created by analysing the poetry generation process (Section *Process Visualization*).
5. The abstract images and the associated poems were then presented to the artist Sandra Lääne. She hand-picked 12 image/poem pairs, and painted 12 paintings. (Section *From Abstract Images to Paintings*).

6. The paintings were exhibited accompanied with the respective poems (Section *Exhibitions*).

Corpus-Based Poetry Generation

In this project, we used the poetry generation machine by Toivanen et al. (2012; 2014). The grammar, including the syntax and morphology of the generated poetry, is obtained in an instance-based manner from English poetry in Project Gutenberg as described by Toivanen et al. (2012). Thus, instead of explicitly representing a generative grammar of the output language, we copy a concrete instance from an existing text and substitute the contents by new words from the document specific associations. In contrast to the original poetry writing method (Toivanen et al. 2012), here we use a specific document or a set of documents as an input to the automatic poetry composition system. In this work, we use news stories as input documents for the poetry composition process. The topics of generated poetry are controlled by using the document specific associations as described in parallel paper by Toivanen et al (2014).

Given a document (or a set of documents), the general outline of the method is the following:

1. Calculate document specific associations by contrasting document associations to English Wikipedia as the background;
2. Choose a poem template from the poetry corpus;
3. Substitute the words in the poem template with words from the document specific associations.

Process Visualization

The abstract images were generated by using two different aspects of the poetry generation process. The geometrical composition of the images was determined by the inputs and outputs of the document specific association generation. The colour palette of the final image was defined by using colors associated to the representative words of the poem.

For generating the geometry of the composition we calculated a transformation matrix between the input text and the document specific associations produced in the poetry writing process. We consider sentences in the input text as bags-of-words. The whole text can then be represented as a binary matrix $I^{m \times n}$, where n rows correspond to words and m columns to sentences of the input text: the value of $I_{i,j}$ is 1 if the word j appears in sentence i and 0 if it does not. The matrix of document specific associations $O^{m \times m}$ is, in turn, defined as a square matrix where $O_{i,j}$ is the association strength between the words i and j. If there is no association between words i and j, the value is 0. We model the term association extraction process as a transformation matrix $P^{n \times m}$ which is obtained as a linear approximation from the following equation

$$I \times P = O.$$

Due to the sparsity of the matrix P, we reduced its dimensionality by principal component analysis and only use the top 15 principal components for aesthetic considerations.

The colour palette of the image was determined by making a Google Image Search with the 3 most important words of the news story, selected based on their sum of association strength to other words (see Toivanen et al. (2014) for details of the weight computation). The 3 words were used as the query to Google Image Search, and from the results, first 3 images were retrieved. The images were concatenated together and the Colorific tool (Hotson and Yencken 2012) was used for extracting their joint colour palette. Finally, the process matrix P was visualized using matplotlib package (Hunter 2007) and the respective colour palette.

An example image can be seen on the left in Figure 1.

From Abstract Images to Paintings

The artist hand-picked 12 computer-generated image/poem pairs to be painted on canvas with acrylics. The artist decided to mainly choose the images by their visual aesthetics (colours and patterns) and less by the associated poems. The artist knew that the images are representations of computational processes, but did not explicit generation method.

Before seeing the computer-generated abstract images, the artist imagined that they contain clear and monotonous lines, opposite to the actually generated images. This gave her the idea to use the computer images as inspiration to create paintings similar to the ones she had imagined before.

The artist then developed a painting technique that involved paper tape to ensure a clinical accuracy of the painted lines, in contrast to the more gradient transitions from one colour to another in the computer-generated images.

A photograph of a final painting can be seen on the right in Figure 1, next to the computer-generated original image on the left. The poem accompanying this image is:

I am gotten like a firm
Al-essawi and thither.
The ban about me
Serves itself into gun of total plans,
York and ordering,
Minimises stoped by the weapon.

Date: 21 December 2012

This was the first time for the artist to work in collaboration with a computer. In her opinion the process was inspiring and interesting. What made it different from her previous experience was that she had to work in a certain framework – provided by the computer, which led her to the idea of using more accurate lines than the computer.

Exhibitions

The work has been exhibited in three venues:

- June 5 - June 30, 2013, Art Museum of Tartu, Estonia
- August 1 - August 30, 2013, Culture Center of Jõgeva, Estonia
- October 24 - November 14, 2013 "Art Corridor" of the Exactum building of the University of Helsinki, Finland

The first exhibition got media coverage in Estonian national newspaper Eesti Ekspress[1] and also in local newspapers.

[1] "http://ekspress.delfi.ee/archive/article.php?id=66524456"

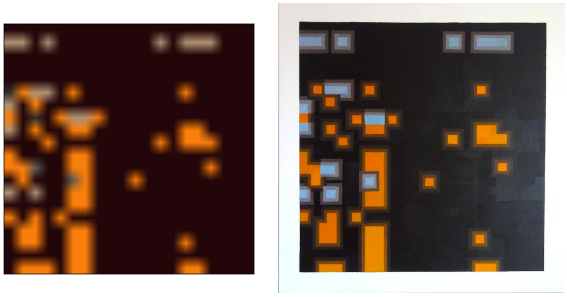


Figure 1: A process visualization image generated by the computer (left) and a photograph of the corresponding acrylic painting (right). For the respective poem, see the text.

Figure 2: Image from the exhibition in Tartu, Estonia

During the exhibition in Helsinki we collected feedback from the audience. Beside the artworks, we placed feedback forms and a box for slipping them in. In order to make giving feedback easy, the feedback form contained two questions:

1. What do you think of this exhibition of "Arts, News & Poetry"?
2. Could you please circle a number below to give a score to the exhibition (1-worst, 5-best)?

The questions were designed this way for three reasons: 1) our goal was to keep people open to giving their ideas; 2) we wanted to avoid giving any sort of bias in any direction; 3) we found it to be more likely to get feedback if the forms are short and easy to fill in.

We received a total of 24 feedback forms from which we removed 4 of the forms which had unreadable gibberish or unrelated comments. 7 of the forms had comments which tended to be negative or sceptical, 10 forms had comments which could be considered positive, and 3 forms had general comments, e.g. "the exhibition raises interesting thoughts".

The positive comments tended to be longer than negative ones. One of the visitors proposed using the technique for encrypting messages. Interestingly, one person found a connection between the exhibition and the computer game Minecraft. One of the longer comments stated:

> "Raises interesting thoughts about what art is. The poems and paintings are seamingly meaningless and will cause thoughts and feelings with the probability of a wall, forest or just about anything [sic!]. Yet there is artists experience involved. I do not perceive any interesting experience from the exhibition apart from these meta-thoughts. All in all, the exhibition feels like random data (which raises thoughts :)"

Some of the negative comments stated that the results is "just noise", or

> "[The exhibition is] very boring, no artistic value, creative, maybe, but dull, monotonic and lacking depth. No serendipity!"

In total we got 19 scores from the feedback forms, with 3.13 as an average and slightly skewed towards positive end.

Related Work

Computer-Human Collaboration in Arts. Our project seems rather unique in the sense that it creates artwork (paintings, in human-computer collaboration) about another creative process (computational poetry generation). This could be classified as conceptual art, claiming that it is the idea and process that constitute the artwork, not alone the resulting paintings and poems. There are numerous conceptual works of art using computational or mechanistic generation of artefacts, and given the richness and variety of the field, we would be surprised if there are no others that take this to the metalevel like we have done.

Even though we are not aware of other art projects addressing exactly the same aspects, the general idea of artistic collaboration between computers and humans is of course not a new one. For instance, the *biomorphs* of Richard Dawkins (1986) have inspired at least Machado and Cardoso (2000) and Sims (1991). In their systems visual art is generated by genetic algorithms but at least partially guided by their users, so that the end result is a mix of computational creativity and human aesthetics.

Computational Creativity Theory. In the field of computational creativity research, a concept related to our work is that of *framing*, i.e., (computer-generated) commentary that adds value to the generated artwork e.g. by describing the underlying processes (Colton, Charnley, and Pease 2011). Process visualization could obviously be considered as a kind of framing for the poetry, providing an (abstract) image of the generation process. However, in our case, the roles are mutual: the paintings clearly take the role of the primary results, and the poems become part of the commentary for the paintings.

Process Visualization. We based our visual artwork on process visualization. An overview of different approaches to program visualization is given by Roman & Cox (1992). In general, the goal of program visualization is to take advantage of humans' high bandwidth of visual system and possibly give another way for people to analyse and understand algorithms (Roman and Cox 1992). They described

examining program's input and output actions and treating the program as a "black box" which transforms the inputs to the outputs as a method which has important theoretical implications, but is not very informative to get insight into the algorithms and is not applicable to all programs of interest.

The rest of the related work tends to be more practical in nature, for instance there is research in algorithm animation (Brown and Sedgewick 1984), visual programming (Myers 1990), and data structure visualizations (Hendrix, Cross II, and Barowski 2004).

Poetry Generation. The poetry generation methods of this work are based on the methods by Toivanen et al. (2012; 2014). A thorough review of different poetry generation methods is not in the scope of this paper as our emphasis here is the process visualization as a possible method of giving framing information, but, e.g. Colton et al. (2012) provide a good overview of the field.

Conclusions and Future Work

In this paper we have given an overview of an implemented and exhibited art project that combines both computational and human creativity in a rather novel way. We proposed an approach for extracting a visual abstraction of a process based on the input and the output of a system. We combined this together with a methodology for generating poems from a news story and used these pieces together for visualizing the abstraction of a process of generating respective poems. An artist then hand-picked some of the images and painted them in her chosen style. The paintings have been exhibited together with the associated poems.

There are many directions for future work. An interesting technical research problem would be developing more intelligent methods for extracting (aesthetic) abstractions of the process. In the best case, analysing such abstractions could be a way of getting insight into creative artefact generation.

An exciting creative possibility would be to make the process visualization interactive: allowing visual manipulation of the process matrix, and then repeating the creative process using the modified matrix to produce a modified poem as an output. If this approach works, it could unify verbal and visual arts in a most interesting way.

Acknowledgments

This work has been supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733 (ConCreTe), the Academy of Finland (Algodan Centre of Excellence), and the Helsinki Doctoral Program in Computer Science (HECSE).

References

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Brown, M. H., and Sedgewick, R. 1984. *A system for algorithm animation*, volume 18. ACM.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In *Proceedings of the Third International Conference on Computational Creativity*.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the 2nd International Conference on Computational Creativity*.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full FACE poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.

Dawkins, R. 1986. The blind watchmaker: Why the evidence of evolution reveals a universe without design. *Penguin, London, UK. DE LA MARE WK (1997) Abrupt mid-twentieth-century decline in Antarctic sea-ice extent from whaling records. Nature* 389(6646):57–60.

Hendrix, T. D.; Cross II, J. H.; and Barowski, L. A. 2004. An extensible framework for providing dynamic data structure visualizations in a lightweight ide. *ACM SIGCSE Bulletin* 36(1):387–391.

Hotson, D., and Yencken, L. 2012. Extracting colors with colorific. Online; Last accessed 24-January-2014. http://99designs.com/tech-blog/blog/2012/05/11/color-analysis/.

Hunter, J. D. 2007. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering* 9(3):90–95.

Machado, P., and Cardoso, A. 2000. Nevar–the assessment of an evolutionary art tool. In *Proceedings of the AISB00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science, Birmingham, UK*, volume 456.

Myers, B. A. 1990. Taxonomies of visual programming and program visualization. *Journal of Visual Languages & Computing* 1(1):97–123.

Řehůřek, R., and Sojka, P. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA.

Roman, G.-C., and Cox, K. C. 1992. Program visualization: The art of mapping programs to pictures. In *Proceedings of the 14th International Conference on Software Engineering*, ICSE '92, 412–420. New York, NY, USA: ACM.

Sims, K. 1991. *Artificial evolution for computer graphics*, volume 25. ACM.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *International Conference on Computational Creativity*, 175–179.

Toivanen, J.; Gross, O.; and Toivonen, H. 2014. The officer is taller than you, who race yourself! Using document specific word associations in poetry generation. In *Proceedings of the 5th International Conference on Computational Creativity, Ljubljana, Slovenia*.

Implementation of a Slogan Generator

Polona Tomašič*
XLAB d.o.o.
Pot za Brdom 100, 1000 Ljubljana
Slovenia
polona.tomasic@xlab.si

Martin Žnidaršič*
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana
Slovenia
martin.znidarsic@ijs.si

Gregor Papa*
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana
Slovenia
gregor.papa@ijs.si

Abstract

Generation of slogans for companies, products or similar entities is a creative task that is difficult to automate. In this paper we describe our attempt of tackling this problem by combining computational linguistics, semantic resources and genetic algorithms.

Introduction

Use of computers for support or automation of tasks in creative industries is on the rise. Several such tools and methods emerged in recent years for various problems. Generation of slogans is one of the less supported problems in this field. There are some online tools available[1], which seem to use templating and provide results of such a kind. To the best of our knowledge, there is only one scientific study dedicated particularly to slogan (and other creative sentences) generation, namely the BRAINSUP framework (Özbal, Pighin, and Strapparava 2013). The BRAINSUP approach emphasises user's control of the generation process. Namely, by user-provided keywords, domain, emotions and similar properties of the slogans, the user has a lot of control over the generation process. This is practically very useful, as it shrinks the huge search space of slogans and improves the quality of results. In our work, on the other hand, we aim at a completely autonomous approach, which is not influenced by the user in any way, apart from being provided by a short textual description of the target entity. In this paper, we present our current approach, which follows the BRAINSUP framework, but also deviates from it with several modifications. At the core of our slogan generation procedure we use a genetic algorithm (GA) (Bäck 1996), which ensures good coverage of the search space, and a collection of heuristic slogan evaluation functions.

*Authors are affiliated also to the Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia. This research was partly funded by the European Union, European Social Fund, in the framework of the Operational Programme for Human Resources Development, by the Slovene Research Agency and supported through EC funding for the project ConCreTe (grant number 611733) and project WHIM (grant number 611560) that acknowledge the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission.

[1]http://slogan4u.com ; http://www.sloganizer.net/en/

Resources

Our slogan generation method requires some specific resources, such as a collection of frequent grammatical relations. Here we list these resources, describe their acquisition methodology and provide some illustrative examples.

Database of existing slogans

The database of exisitng slogans serves as a basis for the initial population generation and for comparison with generated slogans. There is a large number of known slogans for different companies and products available online and there are specialized Web pages that contain collections of slogans. However, none of those sources contain all the necessary information, so we constructed our own database in which each instance consists of: slogan, company/product name, official Web site URL and Wikipedia site URL. Currently the database contains 1041 slogans. Here is an example instance: ["Just do it.", "Nike", "http://www.nike.com/", "http://en.wikipedia.org/wiki/Nike"].

Database of frequent grammatical relations

Frequent grammatical relations between words in sentences were used in some of our processes. For their acquisition we used the Stanford Dependencies Parser (Marneffe, MacCartney, and Manning 2006). Stanford dependencies are triplets containing two words, called *governor* and *dependent*, and the name of the relation between them. The parser also provides part-of-speech (POS) tags and phrase structure trees.

To get representatives of frequent grammatical relations between words, we parsed 52,829 random Wikipedia pages, sentence by sentence, and obtained 4,861,717 different dependencies. Each dependency consists of: name of the relation, governor, governor's POS tag, dependent, dependent's POS tag and the number of occurrences.

Database of slogan skeletons

All the gathered known slogans were parsed with the Stanford Dependencies Parser. Grammatical structure of each slogan, without the content words, was then stored in a database. Each skeleton contains information about each position in the sentence - its POS tag and all its dependency relations with other words in the sentence. For example, skeleton of the slogan "Just do it" is [[['advmod', '***',

'VB', '***', 'RB'], ['2', '1']], [['dobj', '***', 'VB', '***', 'PRP'], ['2', '3']]]. Here the first part tells us that the first word (RB - adverb) is *adverbial modifier* of the second word (VB - verb), and the second part indicates that the third word (PRP - pronoun) is a *direct object* of the second word.

Slogan generation

In this section we describe our slogan generation approach in terms of its inputs, outputs and algorithmic steps.
INPUT consists of two items: (1) a textual description of a company or a product, and (2) the algorithm parameters: evaluation function weights, mutation and crossover probabilities, size of the initial population and maximal number of genetic algorithm iterations.
OUTPUT is a set of generated slogans.
ALGORITHMIC STEPS are the following:

1. Parse the input text for keywords and the main entity.
2. Generate the initial population from random skeletons.
3. Evaluate the slogans and select parents for reproduction.
4. Produce a new generation using crossover and mutations.
5. Repeat steps 3. and 4. until predetermined quality of slogans or maximal number of iterations is achieved.

Extraction of keywords and the main entity

This first step is achieved using the Nodebox English Linguistics library[2]. The main entity is obtained by selecting the most frequent entity in the whole text using *nltk* library (Bird, Klein, and Loper 2009).

Example of the keywords and the entity, extracted from the Coca Cola Wikipedia page:
keywords = ['win', 'produce', 'celebrate', 'using', 'marketing', 'north', 'likely', 'drink', 'century', 'diet', 'production', 'root', 'product', 'beverage', 'water', 'image', 'sugar',... ']
entity = 'Coke'

Generation of the initial population of slogans

The procedure of generating the initial population of slogans is based on the BRAINSUP framework (Özbal, Pighin, and Strapparava 2013), with some modifications and additions. It follows these steps:

1. Select a random slogan skeleton from the database.
2. Choose an empty position, which has the largest number of dependency relations in the sentence. Find the set of all possible fillers for that position. Fillers are words from the database of all grammatical relations between words and must satisfy all predefined dependencies and POS tags.
3. Find the intersection between the set of all possible fillers and the set of keywords. If the obtained set is not empty, choose a random word from it and fill the empty position. In case of an empty intersection, choose random word from the 20% of most frequent possible fillers, and fill the empty position.
4. Repeat steps 2 and 3 until all the empty spots are filled.
5. Check if the generated slogan contains any entities. If it does, replace them with the company entity.
6. Repeat steps from 1 to 5 until the initial population of the predetermined size is built.

[2]http://nodebox.net/code/index.php/Linguistics

Evaluation of slogans

To order the slogans by their quality, an aggregated evaluation function was constructed. It is composed of 10 different sub functions, each assessing a particular feature of a slogan with scores in the interval [0,1]. Parameter of the aggregation function is a list of 10 weights that sum to 1. They define the proportions of sub functions in the overall score.

2-gram function In order to work with 2-grams, we obtained the data set of 1,000,000 most frequent 2-grams and 5000 most frequent words in Corpus of Contemporary American English [3](COCA). The 2-gram evaluation score should to some degree represent the relatedness between words in slogan. We assume that slogans containing many frequent 2-grams, are more likely to make sense. The 2-gram evaluation score is computed in the following manner:

1. Assign a score to every 2-gram in the slogan:
   - if 2-gram is among most frequent 2-grams: score = 1,
   - else if one word is an entity and the other is among 5000 most frequent words: score = 0.75,
   - else if one word is among 5000 most frequent words and the other is not: score = 0.5,
   - else score 0
2. Sum the scores of all 2-grams and divide it by the number of all 2-grams in the slogan.

length function This function assigns score 1 to slogans with less than 8 words, and score 0 to longer ones.

diversity function The diversity function evaluates a slogan by counting the number of repeated words. The highest score goes to a slogan with no repeated words. If a slogan contains identical consecutive words, it receives score 0.

entity function It returns 1, if slogan contains the main entity, and 0, if it doesn't.

keywords function If one up to half words in a slogan belong to the set of keywords, the keywords function returns 1. If a slogan doesn't contain any keyword, the score is 0. If more than half of the words in the slogan are keywords, the score is 0.75.

word frequency function This function prefers slogans with many frequent words, as we assume that slogans which contain a lot of infrequent words are not good. The score is obtained by dividing the number of frequent words by the number of all words in the slogan. Word is considered to be frequent, if it is among 5000 most frequent words in COCA.

polarity and subjectivity functions To calculate the polarity and subjectivity scores based on the adjectives in the slogan, we used the *sentiment* function from *pattern* package for Python (De Smedt and Daelemans 2012). We also integrated the *weight* score from SentiWordNet (Baccianella, Esuli, and Sebastiani 2010), which assigns to each word three sentiment scores: positivity, negativity, objectivity.

[3]Davies, Mark. (2011) N-grams data from the Corpus of Contemporary American English (COCA). Downloaded from http://www.ngrams.info on April 15, 2014.

semantic relatedness function This function computes the relatedness between all pairs of content words in the slogan. Stop words are not taken into account. Each pair of words gets a score based on the path distance between corresponding synsets in WordNet (Miller 1995). The final score is the sum of all pairs' scores divided by the number of all pairs.

structure function During the crossover and mutation phase slogans get deformed and can violate grammatical relations requirements. To avoid unusual grammatical structures in slogans, we parse each new slogan with the Stanford Parser and count the number of infrequent POS tags of word phrases in the parse tree. E.g., the POS tag SBAR (subordinating conjunction), represents only around 3% of all word phrases in English texts. If the number of these POS tags is high, the structure score is low.

Production of a new generation of slogans

A list of all generated slogans is ordered descending with regard to the evaluation score. The best 10% of them are all chosen for reproduction. The other 90% of parent slogans are selected uniformly at random.

A new generation is built by pairing parents and performing the crossover function followed by the mutation function which occur with probabilities $p_{crossover}$ and $p_{mutation}$ respectively. Offspring are then evaluated and compared to the parents, in order to remove very similar ones. Remaining slogans proceed to the next generation. These steps are repeated until a generation of slogans reaches the predefined quality score, or the predefined maximal number of iterations is achieved.

Crossover There are two types of crossover functions, the *big* and the *small* one. Both inspect POS tags of the words in both parents, and build a set of possible crossover locations. Each element in the set is a pair of numbers. The first one provides a position of crossover in the first parent and the second one in the second parent. The corresponding words must have the same POS tag. Let the chosen random pair from the set be (p, r). Using the *big* crossover, the part of the first parent, from the p^{th} position forward, is switched with the part of the second parent, from the r^{th} position forward. For *small* crossover only the p^{th} word in the first parent and the r^{th} word in the second parent are switched. Examples for *big* and *small* crossover are in Figure 1.

Mutation Two types of mutations are possible. Possible *big* mutations are: deletion of a random word; addition of an adjective in front of a noun word; addition of an adverb in front of a verb word; replacement of a random word with new random word with the same POS tag.

Small mutations are replacements of a word with its synonym, antonym, meronym, holonym, hypernym or hyponym. Functions for obtaining such replacements are embedded into the Nodebox English Linguistics library and are based on the WordNet lexical database (Miller 1995).

Deletion of similar slogans Every generated slogan is compared to all its siblings and to all the evaluated slogans from the previous generation. If a new child is equal to any other slogan, it gets removed. If more than half of child's words are in another slogan, the two slogans are considered similar. Their evaluation scores are being compared and the one with the higher rate remains while the other one is removed. The child is also removed, if it contains only one word or if it is longer than 10 words. Deletion of similar slogans is our addition to the basic genetic algorithm. It prevents the generated slogans to converge to the initial ones.

big:

We [PRP] bring [VBP] **good [JJ] things [NNS] to [DT] life [NN].**
Fly [VB] the [DT] **friendly [JJ] skies [NNS].**

→ We bring friendly skies.
Fly the good things to life.

small:

Just [RB] **do [VB]** it [PRP].
Drink [VB]more [JJR] milk [NN].

→ Just drink it.
Do more milk.

Figure 1: Examples for a *big* and a *small* crossover.

Experiments

We made a preliminary assessment of the generator with experiments as described in the following.

Experimental setting

In presented experiments and results we use a case of Italian luxury car manufacturer Ferrari. The input text was obtained from Wikipedia[4].

First, we tried to find the optimal weights for the evaluation function. We tested different combinations of weights on a set of manually evaluated slogans. The comparison of the computed and the manually assigned scores showed that the highest matching was achieved with the following weights: [2-gram: 0.2, length: 0.04, diversity: 0.05, entity: 0.08, keywords: 0.2, frequent words: 0.07, polarity: 0.08, subjectivity: 0.08, semantic relatedness: 0.05, structure: 0.15].

The probabilities for crossover and mutation functions had to be high so that new generations would not be to similar to previous ones. Probabilities used in our experiments were $p\_big\_crossover = 0.6$, $p\_small\_crossover = 0.9$, $p\_big\_mutation = 0.8$, $p\_small\_mutation = 0.6$. These control parameters were set according to the results of testing on a given input text, as their combination empirically leads to convergence.

Due to the high computational complexity of our method, the maximal number of iterations and the maximal size of initial population were 50 and 20. We performed 20 runs for the same input parameters.

[4]http://en.wikipedia.org/wiki/Ferrari on April 29, 2014.

Table 1: Statistics of slogan scores for 10 best final slogans for all 20 runs. (F = Final, IP = Initial Population)

	min	max	average	median	st. deviation
1	0.785	0.888	0.848	0.85	0.032
2	0.817	0.905	0.849	0.847	0.022
3	0.786	0.896	0.832	0.826	0.034
4	0.780	0.895	0.825	0.809	0.040
5	0.777	0.884	0.837	0.837	0.036
6	0.795	0.937	0.830	0.818	0.039
7	0.773	0.884	0.822	0.812	0.037
8	0.795	0.908	0.833	0.815	0.038
9	0.809	0.894	0.842	0.837	0.029
10	0.789	0.917	0.821	0.816	0.035
11	0.796	0.902	0.844	0.840	0.031
12	0.738	0.902	0.817	0.802	0.051
13	0.761	0.904	0.810	0.772	0.045
14	0.759	0.834	0.789	0.782	0.025
15	0.761	0.901	0.816	0.802	0.042
16	0.816	0.900	0.859	0.861	0.028
17	0.779	0.891	0.831	0.829	0.031
18	0.785	0.888	0.844	0.854	0.035
19	0.739	0.883	0.801	0.787	0.054
20	0.792	0.892	0.834	0.819	0.035
avg. F	0.782	0.895	0.829	0.821	0.036
avg. IP	0.6	0.75	0.66	0.65	0.048

Results and discussion

All 20 runs of the algorithm on the same input data had similar statistical results. Statistics of slogan scores of 10 best final slogans for each run are gathered in Table 1. The score average of slogans increased with each iteration. Table 2 shows its progress.

Table 2: The average increase of the average slogan scores after 10, 20, 30, 40 and 50 iterations.

10	20	30	40	50
21.5%	31.5%	34.7%	37.1%	39.3%

The numbers in both tables show that our method ensures higher slogan scores with each new iteration of genetic algorithm, for a given experimental case. Examples of slogans for one specific run of the algorithm are listed in the following two lists. The first one contains 10 best rated initial slogans and the second one contains 10 best rated final slogans. Evaluation scores are in the brackets.

Initial population:

1. Ferrari is body without substance (0.706)
2. The development of Ferrari (0.696)
3. She swam to make They pay (0.695)
4. increasing production to their output (0.686)
5. allow you with stockings (0.678)
6. causing a Ferrari Saturday (0.676)
7. He wins a role and takes on role (0.66)
8. A successful business to wish (0.631)
9. A success for every artist (0.622)
10. Ferrari uses In his Ferrari (0.599)

Final slogans:

1. make The great meaning of Ferrari (0.905)
2. Ferrari is valuable role with every successful closer (0.865)
3. make you these red Ferrari (0.852)
4. Ferrari is in your largest entertainment more (0.85)
5. only allow you we and Ferrari Saturday (0.848)
6. only make it without its Ferrari (0.847)
7. get The largest being more (0.842)
8. Ferrari is worthy substance closer (0.838)
9. a bright Ferrari Saturday (0.832)
10. They takes The turning more (0.817)

The analysis of initial populations and final slogans in all runs shows that the majority of slogans have grammatical mistakes. This is due to the *big* crossover and the *big* mutation functions. Our system currently lacks an evaluation function for detection or correction of these mistakes.

Some seemingly good slogans can be found already in the initial populations. The evaluation function seems not yet aligned well with human evaluation, as such slogans often do not make it to the final round.

Conclusion

The proposed slogan generation method works and could be potentially useful for brainstorming. The genetic algorithm ensures that new generations of slogan candidates have higher evaluation scores. The critical part of the method is the evaluation function, which is inherently hard to formalize and needs further improvement. We believe that the refinement of semantic and sentiment evaluation functions would increase the quality of slogans, not only their scores.

There are also many other ideas for the future work that would improve the quality of slogans. One is checking for grammatical errors and correcting them if possible. In mutation phase there is a possibility of replacing one word with a whole new word phrase. New weights could be also computed periodically with semi-supervised learning on manually assessed slogans.

References

Baccianella, S.; Esuli, A.; and Sebastiani, F. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proc. of LREC 2010*.

Bäck, T. 1996. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python*. O'Reilly Media.

De Smedt, T., and Daelemans, W. 2012. Pattern for Python. *Journal of Machine Learning Research* 13:2063–2067.

Marneffe, M. D.; MacCartney, B.; and Manning, C. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC 2006*.

Miller, G. A. 1995. Wordnet: A lexical database for english. *COMMUNICATIONS OF THE ACM* 38:39–41.

Özbal, G.; Pighin, D.; and Strapparava, C. 2013. BRAINSUP: Brainstorming Support for Creative Sentence Generation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 1446–1455.

Creative Web Services with Pattern

Tom De Smedt
Experimental Media
Research Group (EMRG)
St Lucas Univ. College, Antwerp
tom.desmedt@kdg.be

Lucas Nijs
Experimental Media
Research Group (EMRG)
St Lucas Univ. College, Antwerp
lucas.nijs@kdg.be

Walter Daelemans
Computational Linguistics
Research Group (CLiPS)
University of Antwerp, BE
walter.daelemans@uantwerpen.be

Abstract

Pattern is a Python toolkit for web mining, natural language processing, machine learning, network analysis and data visualisation. In this paper, we discuss how it can be useful as a computational creativity tool, in particular how its new pattern.server module can be used to set up creative web services.

Introduction

Pattern (http://www.clips.ua.ac.be/pattern) is a Python 2.5+ toolkit for web mining, natural language processing, machine learning, network analysis and data visualisation. It is organised in different modules that can be intermixed. For example, the pattern.web module can be used to retrieve Google results, Wikipedia and Wiktionary articles, DBPedia triples, Twitter and Facebook statuses, to crawl and parse HTML, and so on. The pattern.en module has an English part-of-speech tagger, sentiment analysis, regular expressions for inflecting nouns and verbs, and so on. The pattern.vector module contains machine learning tools for classification (e.g, k-NN, SVM), clustering (e.g., *k*-means), dimensionality reduction, feature selection, and so on. The pattern.graph module has tools for network analysis, and for network visualisation using Pattern's canvas.js helper module for interactive graphics in the web browser. For an overview, see De Smedt & Daelemans (2012).

In recent years, the functionality has steadily expanded. Pattern now contains pattern.es, de, fr, it and nl modules for multilingual text analysis, with part-of-speech taggers for Spanish, German, French, Italian and Dutch, and sentiment analysis for Dutch and French (Italian is upcoming). The pattern.web module now supports CSS selectors that make parsing HTML trees more flexible and scalable. The pattern.vector module now comes bundled with LIBLINEAR for fast linear SVM's (Fan, Chang, Hsieh, Wang & Lin, 2008). Finally, our most recent addition is a pattern.server module that can be used to set up web services. It is based on CherryPy[1], and has syntax similar to Flask[2].

[1] http://www.cherrypy.org
[2] http://flask.pocoo.org

Pattern for Computational Creativity

Pattern does not specialise in any particular task. For each task, it provides one or two well-known approaches, usually one that is intuitive and one that is faster (e.g., k-NN vs. SVM). Users that need more may move on to specialised toolkits such as NLTK for natural language processing (Bird, Klein & Loper, 2009) and Scikit-learn for machine learning (Pedregosa, Varoquaux, Gramfort, Michel et al., 2011) as their projects become more involved.

Instead, Pattern offers creative leverage by allowing its users to freely combine a range of cross-domain tools. For example, the toolkit comes bundled with a common sense dataset, which can be traversed as a semantic network with pattern.graph to generate creative concepts (e.g., "Brussels, the toad", see De Smedt, 2013). The pattern.web module can then be used to search the web for evidence whether or not such concepts already exists to assess their novelty ("external validation", Veale, Seco & Hayes, 2009). Or, pattern.web can be used to mine words, word inflections and their parts-of-speech from the Italian Wiktionary, and analysed with the pattern.metrics helper module to construct an Italian part-of-speech tagger and regular expressions for Italian verb conjugation (De Smedt, Marfia, Matteucci & Daelemans, in press). With pattern.server we can subsequently launch a web service for Italian part-of-speech tagging that others can harness for language generation games, for example.

One user has compared Pattern to a "Swiss Army knife". Another user has called it a "treasure trove". In short, the toolkit is not designed for a specific purpose; rather it provides an open-ended range of tools that can be combined and explored – similar in philosophy to Boden's view on creativity (Boden, 2006). We think that Python coders who need to deal with data mining, natural language processing, machine learning, and so on, and who are active in the digital humanities and in the computational creativity (CC) community, will find Pattern useful, especially with its new pattern.server module.

Web Services with Pattern

Computational creativity covers a diverse range of tasks. It has been argued that web services are beneficial to the CC community (Veale, 2013). Different researchers can work on different tasks and share their results without having to reinvent algorithms from published pseudo code, deal with myriad installation instructions or adopt new programming languages. Instead, a request is sent to a web service and the response can be incorporated into any project. Many different web services can be combined to augment novel creativity research.

To demonstrate how web services work in Pattern, the example below implements a web service for semantic similarity, using just a few lines of code. Pattern comes bundled with WordNet 3 (Fellbaum, 1999). It also has an algorithm for Lin's semantic similarity (Lin, 1998), which measures the likelihood that two concepts occur in the same context, and whether they have a common ancestor in the WordNet graph. The `similarity()` function in this example takes two nouns, retrieves their WordNet synsets and estimates the semantic similarity between the two synsets as a value between 0.0 and 1.0. For example, the similarity between "cat" and "dog" is 0.86, whereas the similarity between "cat" and "teapot" is 0.0.

The `@app.route()` decorator defines the relative URL path where the web service is available. Optional keyword arguments of the `similarity()` function can be passed as URL query string parameters. The `similarity()` function returns a Python dictionary that will be served as a JSON-formatted string. Finally, the `app.run()` function starts the server.

```
from pattern.en import wordnet
from pattern.server import App

app = App()

@app.route('/similarity')
def similarity(w1='', w2=''):
 synset1 = wordnet.synsets(w1)[0]
 synset2 = wordnet.synsets(w2)[0]
 s = synset1.similarity(synset2)
 return {'similarity': round(s, 2)}

app.run('127.0.0.1', 8080, embedded=False)
```

In this case, the server runs locally. With `embedded=True` it will run as a mod_wsgi process on an Apache server. An optional parameter `debug=True` can be used to enable or disable error messages.

To try it out, we can execute the source code and visit http://127.0.0.1:8080/similarity?w1=cat&w2=dog in a web browser. The response is `{'similarity': 0.86}`. The example can be expanded with input validation and support for different word senses and word types.

Case Study: Weaseling Web Service

The following example demonstrates how pattern.en and pattern.server can be combined into a *weaseling* service for linguistic creativity. Weasel words are used to convey an air of meaningfulness in vague or ambiguous statements, as in "experts have claimed that this could be …".

The `weasel()` function takes a sentence and injects modal verbs so that, for example, "is" becomes "could be". The given sentence is part-of-speech tagged and verbs are transformed for common cases: non-action verbs get an additional "might" (e.g., "want" = "might want"), other verbs are passed to the pattern.en `conjugate()` function to transform them into the present participle tense (e.g., "run" = "might be running").

```
from pattern.en import parsetree
from pattern.en import conjugate
from pattern.server import App
from random import random

NONACTION = set((
   'appear', 'believe', 'contain', 'doubt',
   'exist', 'fear', 'feel', 'hate', 'hear',
   'hope', 'know', 'look', 'love', 'mean',
   'need', 'prefer', 'see', 'seem', 'sound',
   'think', 'understand', 'want', 'wish'
))

app = App()

@app.route('/weasel')
def weasel(s=''):
 r = []
 for sentence in parsetree(s, lemmata=True):
   for w in sentence:
     if r and w.tag.startswith('VB') \
      and random() < 0.05:
       r.append('often')
     if not w.tag.startswith('VB'):
       r.append(w.string.lower())
     elif w.lemma in ('be', 'have') \
      and w.tag not in ('VB', 'VBG', 'VBD'):
       r.append('might')
       r.append(w.lemma)
     elif w.lemma in ('be', 'have') \
      and w.tag == 'VBD':
       r.append('might')
       r.append('have')
       r.append(conjugate(w.lemma, 'VBN'))
     elif w.tag in ('VBP', 'VBZ') \
      and w.lemma in NONACTION:
       r.append('might')
       r.append(w.lemma)
     elif w.tag in ('VBP', 'VBZ'):
       r.append('might')
       r.append('be')
       r.append(conjugate(w.lemma, 'VBG'))
     else:
       r.append(w.string.lower())
   return ' '.join(r)

app.run('127.0.0.1', 8080, embedded=False)
```

For brevity, case sensitivity, punctuation, negation, verbs preceded by infinitival to, and verbs in the past tense are not handled. We can further improve the algorithm by injecting adverbs such as "often" and "perhaps" in a smarter way, transform quantifiers to vague expressions ("two" = "many"), and so on.

To try it out, we can execute the source code and visit http://127.0.0.1:8080/weasel?s=the+information+centre+is+to+the+north+of+here. The response is: "the information centre often might be to the north of here". Similarly, "you need a parking ticket" becomes "you might need a parking ticket", "this rental car runs on diesel fuel" becomes "this rental car might be running on diesel fuel" and "your hotel room was already paid for" becomes "your hotel room might have been already paid for".

The following code snippet queries our weaseling web service (running locally) and transforms Twitter statuses that contain a #travel hashtag:

```
from pattern.web import Twitter
from pattern.web import URL
from pattern.web import encode_url
from pattern.web import decode_utf8

API = 'http://127.0.0.1:8080/weasel?s='

for tweet in Twitter().search('#travel'):
  s = tweet.text
  r = URL(API + encode_url(s)).download()
  print decode_utf8(r)
  print
```

One tweet now states: "Miami International Airport often might be experiencing arrival delays of up to 30 minutes". Then again, it might not.

Further reading

De Smedt's doctoral dissertation[3] (2013) has more in-depth case studies of how Pattern can be used for CC.

For example, it discusses PERCOLATOR, a program that generates visuals based on today's news, FLOWEREWOLF, a poetry generator, PERCEPTION, a semantic network of commonsense, and MAD TEA PARTY, a problem solving algorithm (e.g., to open a locked door for which you don't have a key, you stubbornly club it with an albatross).

Future Work

Our new pattern.server module is not documented yet. Some examples of use are included in the latest Pattern release. We will provide extensive documentation[4] and unit tests once all lingering bugs have been fixed. Interested users are encouraged to contribute updates on GitHub[5].

Pattern is not ready yet for Python 3, unfortunately. Some preliminary steps have already been taken to make the toolkit available for Python 3. Work will continue along this line in the future.

Acknowledgements

The development of Pattern is supported by the Computational Linguistics Research Group at the University of Antwerp, Belgium, and the Experimental Media Research Group at the St Lucas University College of Art & Design, Antwerp, Belgium.

References

De Smedt, T., and Daelemans, W. 2012. Pattern for python. *The Journal of Machine Learning Research*, 13(1): 2063-2067.

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., and Lin, C. J. 2008. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9: 1871-1874.

Bird, S., Klein, E., and Loper, E. 2009. *Natural language processing with Python*. O'Reilly Media, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. 2011. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12: 2825-2830.

De Smedt, T., Marfia, F., Matteucci, M., Daelemans, W. In press. Using Wiktionary to build an Italian part-of-speech tagger. In *Proceedings of NLDB 2014*.

De Smedt, T. 2013. *Modeling Creativity: Case Studies in Python* (doctoral thesis). University Press Antwerp. ISBN 978-90-5718-260-0.

Veale, T., Seco, N., & Hayes, J. 2004. Creative discovery in lexical ontologies. In *Proceedings of the 20th international conference on Computational Linguistics*, 1333. Association for Computational Linguistics.

Boden, M. A. 2003. *The creative mind: Myths and mechanisms*. Routledge.

Fellbaum, C. 1999. *WordNet*. Blackwell Publishing Ltd.

Lin, D. 1998. An information-theoretic definition of similarity. In *ICML,* 98: 296-304.

Veale, T. 2013. A Service-Oriented Architecture for Computational Creativity. *Journal of Computing Science and Engineering, 7*(3): 159-167.

[3] http://bit.ly/modeling-creativity
[4] http://www.clips.ua.ac.be/pages/pattern-server
[5] http://www.github.com/clips/pattern

Kill the Dragon and Rescue the Princess: Designing a Plan-based Multi-agent Story Generator

Iván M. Laclaustra, José L. Ledesma, Gonzalo Méndez, Pablo Gervás
Facultad de Informática
Universidad Complutense de Madrid
Madrid, Spain
{ilaclaus, josledes, gmendez, pgervas}@ucm.es

Abstract

We describe a prototype of a story generator that uses a multi-agent system and a planner to simulate and generate stories. The objective is to develop a system that is able to produce a wide range of stories by changing its configuration options and the domain knowledge. The resulting prototype is a proof of concept that integrates the simplest pieces that are necessary to generate the stories.

Introduction

When trying to generate stories automatically, it is mandatory to research how actual stories work. That, inevitably, makes you think: "What makes a story interesting?".

While researching for this project, we realized that in a story, most of the times, the most important thing is not WHAT, but HOW things happened. This represents a huge challenge, since it is difficult to simulate things such as time (we must be able to simulate time, so that things are not done immediately), conversations (they have to be fluid, spontaneous), and many more. Similarly, there are some actions that lack interest in themselves, but may have some if combined with others. For example, eating or sleeping, are actions that may not appear in the final story, but may be worthy of attention if the character meets someone while eating. Of course, some of the stories generated will just be sets of facts without any relation or interest, but that is part of the process.

One of the ways we have for generating stories is by simulating them. Then, you just have to run the simulation and see what happens. We achieve this by simulating the stories using autonomous intelligent agents. Each of the agents of the story is going to act as a character, which will act independently from the others, but depending on the story world's state. Then stories are generated by "filming" what these actors do and say. Our main goal for now, is to make a small Dungeons & Dragons story, which has more than one ending.

Related Work

The first story telling system for which there is a record is the Novel Writer system developed by Sheldon Klein (Klein et al. 1973), which created murder stories within the context of a weekend party. It relied on a microsimulation model where the behaviour of individual characters and events were governed by probabilistic rules that progressively changed the state of the simulated world (represented as a semantic network). The flow of the narrative arises from reports on the changing state of the world model. A description of the world in which the story was to take place was provided as input. The particular murderer and victim depended on the character traits specified as input (with an additional random ingredient). The motives arise as a function of the events during the course of the story. The set of rules is highly constraining, and allows for the construction of only one very specific type of story. The world representation allows for reasonably wide modeling of relations between characters. Causality is used by the system to drive the creation of the story but it is not represented explicitly.

TALESPIN (Meehan 1977) is a system which tells stories about the lives of simple woodland creatures. TALESPIN was based on planning: to create a story, a character is given a goal, and then the plan is developed to solve the goal. TALESPIN introduces character goals as triggers for action. Actions are no longer set off directly by satisfaction of their conditions; an initial goal is set, which is decomposed into subgoals and events. TALESPIN introduced the possibility of having more than one problem-solving character in the story (and it introduced separate goal lists for each of them). The validity of a story is established in terms of: existence of a problem, degree of difficulty in solving the problem, and nature or level of problem solved.

Lebowitz's UNIVERSE (Lebowitz 1985) modelled the generation of scripts for a succession of TV soap opera episodes. It aimed at exploring extended story generation, a continuing serial rather than a story with a beginning and an end. It is in a first instance intended as a writer's aid, with additional hopes to later develop it into an autonomous storyteller. The actual story generation process of UNIVERSE uses plan-like units (plot fragments) to generate plot outlines. Plot fragments provide narrative methods that achieve goals, but the goals considered here are not character goals, but author goals. This is intended to allow the system to lead characters into undertaking actions that they would not have chosen to do as independent agents. The system keeps a precedence graph that records how the various pending author goals and plot fragments relate to each other and to events that have been told already. To plan the next stage

of the plot, a goal with no missing preconditions is selected and expanded.

The line of work initiated by TALESPIN, based on modeling the behaviour of characters, has led to a specific branch of storytellers. Characters are implemented as autonomous intelligent agents that can choose their own actions informed by their internal states (including goals and emotions) and their perception of the environment. Narrative is understood to emerge from the interaction of these characters with one another. This guarantees coherent plots, but, as Dehn pointed out, lack of author goals implies they are not necessarily very interesting ones. However, it has been found very useful in the context of virtual environments, where the introduction of such agents injects a measure of narrative to an interactive setting.

The Virtual Storyteller (Theune et al. 2003) introduces a multi-agent approach to story creation where a specific director agent is introduced to look after a plot. Each agent has its own knowledge base (representing what it knows about the world) and rules to govern its behaviour. In particular, the director agent has basic knowledge about plot structure (that it must have a beginning, a middle, and a happy end) and exercises control over agent's actions in one of three ways: environmental (introduce new characters and object), motivational (giving characters specific goals), and proscriptive (disallowing a character's intended action). The director has no prescriptive control (it cannot force characters to perform specific actions). Theune et al. report non-structural rules are contemplated, to measure issues such as surprise and "impressiveness". The Virtual Storyteller includes a specific narrator agent, in charge of translating the system representation of states and events into natural language sentences. The development effort on the narrator seems to have focused on correct generation of pronouns to make the resulting text appear natural.

The story generator

The objective of this work is to develop a story generator that can generate different stories using the same initial information and that, in addition, can be easily modified to generate a wider range of stories.

With these objectives in mind, we have developed a first prototype that works as a proof of concept to test our approach. This prototype has been developed using very simple, unsophisticated components with the aim of substituting them with more complex ones once the feasibility of the solution has been tested.

The generator is structured in four modules, each of them with their corresponding configuration files: a multi-agent system, which contains an agent for each character and a set of managing agents (currently the world agent, the simulation agent and the director agent), a logger (in charge of collecting the events of the story), a planner (what the characters use to know what to do), and the world (contains the map where the characters interact).

The world

The world is basically a map with different locations, connected by paths between them, in order to make the characters move around it. The prototype we have built has a map formed by three locations:

- Castle: Where the king and the princess are.
- Village: Where the knight starts at.
- Cave: Dragon's home.

Since one of our main goals is to make this storyteller easy to configure, we decided to use text files to load the map and the objects present in each location. The map is structured as an XML file that contains a list of locations with pointers to the locations they are connected to, and the objects and characters situated there, so it works as a graph.

The multi-agent system

The multi-agent system is implemented using the JADE (Bellifemine, Caire, and Greenwood 2007) agent platform. This first prototype generates stories with four types of characters:

- Princess: the character around which the story is built up.
- Dragon: its goal is to kidnap the princess and hold her prisoner in his cave.
- King: the father of the princess. When his daughter is kidnapped, his goal is to find a suitable knight and hire him to kill the dragon. If the knight fails, the king looks for another one, until the princess is safe and sound back in her father's castle.
- Knight: He has no goals until the princess is kidnapped. From then on, his goal is to kill the dragon and take her back to her father.

New stories can be created by simply adding more characters of a type, which are specified at the beginning in a configuration file. In addition, the director agent may create them if it fits the objectives of the story. For example, creating more than one knight, when the princess is kidnapped, the king will look for all the knights available, and will hire the one with lowest fees.

Each character works as a finite state machine consisting of one state per behavior type and a "waiting" state where they are when they don't have active goals.

The world agent is in charge of managing the map, so that all the other agents have a consistent view of the world. Every time a character moves to a new location, he has to send a message to the world agent, so the map gets updated.

The simulation agent is in charge of managing the result of the actions that cannot be directly obtained by the planner, such as the result of the battle between the dragon and the knight.

Finally, the director agent is the one in charge of creating all the necessary agents of the story, these being: the world agent, the simulation agent and the characters. It also makes the necessary decisions to keep the story going, such as setting new goals for the characters. Currently, these decisions are hand written in a configuration file, but the purpose is for this agent to be able to generate them dynamically according to certain heuristics or ask the user to suggest what the new goals should be, in order to make the generator more interactive.

Planning

Each character's actions are driven by their own goals, which are used to plan the sequence of actions they have to carry out to achieve these goals. At the beginning we thought of using just one planner to generate the whole story, but soon it was clear that the planning process would be costly, that the number of possible stories would be small and that it would be difficult to obtain valid plans for agents with conflicting interests. Therefore, we decided it would be more suitable to use separate planners for each agent, so that each of them could make their own plans according to their interests and, in case of conflict, they would have to create new plans to achieve their goals.

We decided to use a STRIPS-based planner (Fikes and Nilsson 1971), since it is quite simple and it is a straightforward option to generate simple stories. In addition, we wanted it to work with PDDL (McDermott 1998) so it would be easy to substitute it with a more sophisticated one in the future.

With this choice, adding a new character to the story involves the creation of another class with the character and two PDDL files, one for its actions, and one for its initial state and goals.

We decided to use the JavaFF planner (Coles et al. 2008) because it works with PDDL and it is open source. The planner takes the domain and the problem in PDDL as inputs, and writes the plan (as a list of actions) into an output file. Since it is open source, we were able to modify it, in order to make the planner return a list of actions (the data structure managed by the planner) instead of writing it to a file. By just adding new actions to the character's PDDL file, new stories are generated, as plans may change including these new actions.

At the time of writing this paper, agents make their plans sequentially (one makes its plan and executes it, then the next one), so that they don't interfere with each other's goals while executing their plans. This reduces the richness of the generated stories, but it is still a good solution to test the validity of the proposed solution.

Capturing the events of the story

As we already said, the only important things are not only the events themselves, so we need a way to gather what happens in the story, but also what is "said" and in what context. Namely, we need a log of everything that happens in the simulation, including the actions that are carried out and the messages exchanged between the agents. We have used the log4j library (Gulcu 2003), which allows the user to enable logging at runtime without modifying the application binary. It also allows us to decide what to enter the log (in our case, it would be everything), the layout, what to save in the log (date, action, agent) and more. Everything is configurable via a parameters file, and will be saved as a log file.

This log is what enables us to actually know what has happened in a certain story, what actions were executed and what was said (scilicet, what messages were interchanged between the agents). However, we must keep in mind that not all the exchanged messages are likely to appear in the final story. For example, all characters have to send a message to the world agent when moving, in order to keep the map updated. These messages should not appear in the story, as their goal is to guarantee internal consistency.

Results

We have implemented a simple prototype where all the described components work together to generate simple, short variations of a story (in Spanish) where a dragon kidnaps a princess and her father the king manages to hire a knight who rescues her and takes her back to her father:

El rey Felipe está preparado.
La princesa Laura despierta.
La princesa sale del castillo.
El dragón Draco emprende el vuelo en busca de alguna princesa desprotegida.
La princesa Laura ha sido secuestrada.
El rey intenta pedir rescate para la princesa Laura.
El caballero Rafael entra en escena.
El rey intenta pedir rescate para la princesa Laura.
El caballero Rafael busca al dragón Draco.
El dragón Draco ha muerto en batalla.
La princesa Laura fue liberada.
El rey entrega 50 monedas al caballero Rafael.
La princesa llega al castillo con el caballero Rafael.
La princesa Laura pone fin a su aventura.

As far as we have been able to test, it is easy to modify the world map to add new locations and situate the characters in them, so they have to make longer journeys to achieve their goals. It is also easy to add new characters of existing kinds so, for example, we can add a second dragon that tries to kidnap the princess from the first one's den.

To make further changes, such as adding new types of characters or actions, it is already necessary to modify the source code of the generator, as well as the domain knowledge, but the code is sufficiently well crafted so that these changes can be easily made. We still have not tested how easy it is to generate a story in a different domain, such as a superheroes story, a western or a love story, but as far as we can see now it may be more painstaking than difficult.

As of now, the stories we generate consist of all the events that take place in the simulation, so our current work is focused on the content extraction, so that we can tell just the relevant events in a relevant order.

To transform the generated logs into text we are using the TAP text generator (Gervás 2011) that receives a crafted set of information and transforms it into an ordered set of sentences that replicates the events that took place in the simulation in the form of a story.

Therefore, in a still simple way, we have developed a story generator that, by means of simple modifications, is able to generate a fair amount of different, although related, stories.

Future Work

Some of the goals we had in mind at the beginning of this project could not be achieved, mostly because of time constraints. We describe some of them here, so they can be used as a starting point for future contributions.

One of the first thing that comes to mind is expanding the world. As the characters and world we are using now are very limited, stories generated are just little paragraphs and there are not many variations between different executions of the application. Just by adding new locations and new characters, we will be adding more possibilities to the story to move along, so that we get more possible stories, which become more intricate at the same time.

As we said before, at the moment, the characters in our application work sequentially, for practical reasons. This reduces the possibilities of the stories generated, since it is more difficult for conflicting interests to appear, or for characters to collaborate to achieve a common goal. A good improvement would be to make all the characters work in parallel, so they would make their plans based on the initial state. While executing their plans, the actions of some characters may interfere in the plans and goals of others. There is when re-planning comes in. Re-planning would make characters interact a lot more, making them compete for the resources to achieve their goals.

In addition, we may want to increase the richness of the stories by making the characters more complex. Adding a slight mood to the characters can make possible stories increase significantly, as the same character may have different behaviors with different moods. Another possibility would be to add feelings and even personality traits.

A lot of richness can also be added via expanding the map. Having a sub-map inside every location would make much more complex plans. Each location can contain different objects, usable and decorative, so the characters can interact with them. For example, you could have a dragon which cannot be killed without a magical sword, so the knight has to find the hidden key to get it.

A much more difficult (and interesting) goal is to make the theme of the story configurable. The idea is to create a configuration file where you can state the theme of the story. That would make everything more difficult, since you can't work with the characters directly. The agents can adopt the role of "actors", instead of characters. With that, there would be a "main character", an "antagonist", a "damsel in distress", and various "secondary actors" in each story. By doing this, you could include in the theme configuration file the names of the characters, their mood (if any), their role, how their actions work (the action "attack" for a knight would make him use his sword, while for a policeman, it would make him use his gun), and have a PDDL file of actions for each role.

Another improvement would be to make the user take the role of a character, so his decisions affect the final result of the story. At first, it could work as in conversational adventures (Montfort 2004), so the user tells the system what actions to carry out. After that, the system would work just as it usually does.

Finally, another option is to give the characters the possibility of making up the details of the story. For example, in our story, the knight could pretend he has a magical weapon to kill the dragon. This endows the stories generated with a whole new level of richness, because new facts are created on the fly.

Acknowledgments*

This paper has been partially supported by the projects WHIM 611560 and PROSECCO 600653 funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program.

References

Bellifemine, F. L.; Caire, G.; and Greenwood, D. 2007. *Developing multi-agent systems with JADE*. Wiley series in agent technology. Wiley.

Coles, A.; Fox, M.; Long, D.; and Smith, A. 2008. Teaching forward-chaining planning with javaff. In *Colloquium on AI Education, Twenty-Third AAAI Conference on Artificial Intelligence*.

Fikes, R. E., and Nilsson, N. J. 1971. Strips: A new approach to the application of theorem proving to problem solving. In *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, IJCAI'71, 608–620. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Gervás, P. 2011. UCM submission to the surface realization challenge. In *Surface Realization Challenge. Challenges 2011 Session at 13th European Workshop on Natural Language Generation (ENLG 2011)*.

Gulcu, C. 2003. *The Complete Log4j Manual*. QOS.ch.

Klein, S.; Aeschliman, J. F.; Balsiger, D.; Converse, S. L.; Court, C.; Foster, M.; Lao, R.; Oakley, J. D.; and Smith, J. 1973. Automatic novel writing: A status report. Technical Report 186, Computer Science Department, The University of Wisconsin, Madison, Wisconsin.

Lebowitz, M. 1985. Story-telling as planning and learning. *Poetics* 14:483–502.

McDermott, D. 1998. PDDL - the planning domain definition language. Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control.

Meehan, J. R. 1977. TALE-SPIN, an interactive program that writes stories. In *In Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 91–98.

Montfort, N. 2004. *Twisty Little Passages: An Approach to Interactive Fiction*. Cambridge, MA, USA: MIT Press.

Theune, M.; Faas, E.; Nijholt, A.; and Heylen, D. 2003. The virtual storyteller: Story creation by intelligent agents. In *Proceedings of the Technologies for Interactive Digital Storytelling and Entertainment (TIDSE) Conference*, 204–215.

You Can't Know my Mind: A Festival of Computational Creativity

Simon Colton
Computational Creativity Group, Department of Computing, Goldsmiths, University of London, s.colton@gold.ac.uk

Dan Ventura
Computer Science Department, Brigham Young University, ventura@cs.byu.edu

www.thepaintingfool.com/galleries/you_cant_know_my_mind

Abstract

We report on a week-long celebration of Computational Creativity research and practice in a gallery in Paris, France. The festival was called *You Can't Know my Mind*, and was intended to introduce to the public the idea that researchers such as ourselves are writing software to be surprisingly unpredictable and creative in nature. The festival included a traditional art exhibition with a vernissage, a live music evening, a poetry night coupled with a food tasting, and a week long demonstration of mood-driven portraiture from The Painting Fool software. Each of the events – which are described here for the first time – involved an element of creative responsibility taken on by various software systems. The success of the festival was demonstrated in terms of attendance and feedback, pieces written by journalists, and follow up events which have taken place in 2013 and 2014.

Introduction

In addition to advancing scientific and philosophical understanding of creativity, a long-term aim of Computational Creativity research is to embed creative software into society. For the general public to accept software as being independently creative, they need exposure to such software in cultural settings. To this end, we held the first *Festival of Computational Creativity* in the Galerie Oberkampf, located in the 11th arrondissement of Paris, France, during the week of 12th to 19th July 2013. As described in the next section, the festival consisted of five elements: an art exhibition, a live music performance, poetry reading, food tasting and a portraiture demonstration. Each element showcased a different system/project contributing creatively to the event, and – as highlighted by the festival name – the overall purpose was to portray software as being possible of autonomous, unpredictable, yet interesting and creative behaviour.

Our aim with the festival was to expose audiences to the main ideas of Computational Creativity within a culturally relevant setting, rather than to study audience experiences. Hence, we did not undertake experiments to gauge reactions to the ideas, systems and outputs presented. As described in the discussion section below, we claim success for the event through the number of attendees, the informal feedback we gained, some attention from journalists and the invitations to demonstrate the portraiture system in further events. We conclude in the discussion section with a brief look at future directions, and end with a montage of images from the festival, which we refer to as images A to N throughout.

Elements of the Festival

Art Exhibition

The art exhibition ran for the duration of the festival in the Galerie Oberkampf and was open to the public for 10 hours each day. The curator was Blanca Pérez Ferrer, who, in collaboration with the authors, chose and arranged 42 pieces produced by The Painting Fool software (Colton 2012), which has a long history of involvement in Computational Creativity projects described at www.thepaintingfool.com. The first 4 pieces (image G of the montage) came from a back-catalogue of pieces which have been previously exhibited (Colton and Pérez-Ferrer 2012). In addition, 14 new pieces from a series entitled *Concrete Nudes* were selected and arranged along the main wall of the gallery (images J and L) – see Figure 1 for examples. These were produced by The Painting Fool simulating handwriting onto digital photographs of concrete walls taken in Rio de Janeiro. The handwriting (of random words) picked out depictions of female and male bodies, via their silhouettes and the capturing of internal contours with breaks in the text. Examples were used in the publicity material for the festival, such as the poster (image H), and the frontage of the gallery (image E). Finally, two sets of 12 postcard sized prints from The Painting Fool's most recent projects were chosen and hung (image K). The vernissage for the exhibition was attended by around 70 people (image F). Around 50 people visited the exhibition from the street during the week, and there were 2 private viewings.

Figure 1: Examples from the *Concrete Nudes* series.

Portraiture Demonstration

The Painting Fool is software that we hope will be taken seriously as a creative artist in its own right, one day. We have cultivated its image through web pages, exhibitions and papers, and given it certain behaviours with the hope that it becomes increasingly difficult for people to use the word 'uncreative' to describe what it does. Note that, given the philosophical standpoint presented in (Colton et al. 2014), we aim to avoid the uncreative label rather than to gain a label of 'creative'. The central exhibit of the exhibition was also called You Can't Know my Mind and involved The Painting Fool producing portraits, with the explicit purpose of modelling artistic behaviours onto which people can project the words: *skill, appreciation, imagination, learning, reflection* and most notably, *intentionality*. We argue in (Colton et al. 2014) that software lacking such behaviours is relatively easy to call uncreative. The exhibit works as follows:

(i) When a person sits down for a portrait, the software has been reading *The Guardian* newspaper articles for some time: performing sentiment analysis to determine whether an article is upbeat or downbeat relative to the corpus, and extracting key phrases with which to search for related articles. The average sentiment over 10 recent articles is used to simulate the software being in a very positive, positive, experimental, reflective, negative or very negative 'mood'.

(ii) If the software is in a very negative mood, it essentially tells the sitter to go away, refusing to paint a portrait on the basis of having recently read too many downbeat articles. It chooses the most negative phrase in the most negative article, and uses this in a commentary for the sitter to take away, which explains why it couldn't paint their portrait.

(iii) If in a positive/very positive mood, the software chooses one/two of nine upbeat adjectives (e.g. *bright*, *colorful*, *happy*) and directs the sitter to smile while it extracts their image from a video recording over a green-screen background (image N). If in a negative mood, the software chooses one of six downbeat adjectives (e.g. *bleary*, *bloody*, *chilling*) and directs the sitter to express a sad face. If in an experimental mood, it chooses one of 11 neutral adjectives (e.g. *glazed*, *abstract*, *calm*) and asks the sitter to pull an unusual face. If in a reflective mood, the software chooses an adjective for which it has previously had a failure (see later).

(iv) The chosen adjective is used to select a filter (from a set of 1,000 possibilities) that, when applied to an image of a face, is likely to achieve an appropriate visualisation for the adjective. Appropriateness is modeled using a set of visuolinguistic association (VLA) neural networks (one per adjective) borrowed from the DARCI system (Norton, Heath, and Ventura 2013). These networks have learned correlations between visual features and semantic (adjectival) concepts, and high network output indicates high appropriateness for the adjective represented by the network. The background in the captured facial image is replaced with an arbitrary abstract art image, the chosen filter is applied to both the background and foreground (face) image, and edge detection is used to overlay edges from the face which pick out features of the sitter. The combined background+foreground+edge image is taken as a 'conception' of what The Painting Fool aims to achieve in its rendering.

(v) One of seven rendering styles involving the simulation of paints (2 styles), pencils (3) and pastels (2) is chosen to produce the portrait. If a pairing of adjective/style hasn't been attempted before, then that style is chosen, otherwise a style is chosen according to the probabilistic model it has learned for the adjective (see later), with better styles more likely. A hand appears on-screen, holding a pastel, pencil or paintbrush, and proceeds to render the image in a vaguely human-like fashion. This process (images D and N) takes from a few minutes for pastels to around 20 minutes for paints.

(vi) Once the rendering is complete, the VLA neural network for the chosen adjective is again used, this time to assess the appropriateness of the rendered image and hence whether it is actually appropriate to use the intended adjective to describe the final portrait. VLA outputs for the conception and the final rendered image are compared to assess whether the rendering technique has increased or decreased (relative to the conception) the appropriateness of the portrait (for conveying the adjective). This assessment determines whether the session has been a 'great success' (significantly increased) or a 'miserable failure' (significantly decreased) or something in between. To end the portraiture session, The Painting Fool prints the portrait with a commentary on the reverse, as per Figure 2. The commentary details the mood the software was in and what adjective it chose, shows the conception compared with the final portrait, and discusses whether it has achieved the aim of producing a portrait of a particular style and how the portrait compares with the conception in that respect. Finally, VLA neural networks for all negative/positive/experimental adjectives are opportunistically applied to the portrait, to see if it can be further described with additional pertinent adjectives.

(vii) Before returning to reading news articles, The Painting Fool scores how effectively the rendering method conveys the chosen adjective. In particular, if it has failed (by significantly reducing the VLA network output for that adjective), then, when in a reflective mood in the future, if this adjective is chosen again, the portrait will be attempted with a different rendering style. In this way, the system builds a probabilistic model of which rendering styles are likely to successfully convey which adjectives, e.g., it learns that pencils are better at producing monochrome or bleary portraits, while paints are better for busy or patterned portraits.

As an example, in Figure 2, while in a negative mood, the software chose the adjective 'bleary', which led to it selecting an image filter which desaturated the image, as depicted in the top conception image of Figure 2. It chose to simulate paints to produce the portrait, as depicted in the bottom rendered image of Figure 2, and then commented that (a) the final portrait is very bleary overall, and (b) it had achieved the same amount of bleariness in the rendered image as the conception, with which it was OK. As a final flourish, it also points out that the portrait is bleached, which fits its mood. Over the week of the festival, more than 100 portraits were produced, and we chose 60 to fill the back wall of the gallery towards the end of the festival (image M).

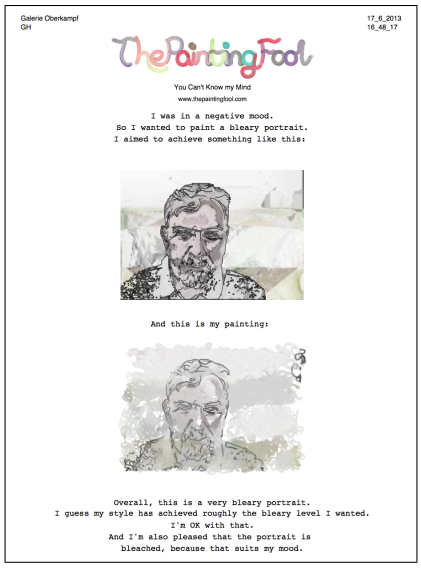


Figure 2: Example portraiture commentary.

Moody Music Evening; Poems and Potage Night

On the first evening of the festival, musician Stéphane Bissières played live to an audience of around 50 people (image A). As part of the performance, The Painting Fool's newspaper-reading mood model (described above) was adapted to inform software for performing affective, real-time sound design and rhythm construction. The software's output was converted to MIDI and sent to Bissières' music system, requiring him to react musically in real-time. Bissières and the system collaborated on three different musical sets, each approximately 20 minutes in length. Each set had a different musical feel, effected by three different algorithmic approaches to how different moods would affect composition and performance. Bissières was enthusiastic about collaborating with an autonomous system. He, and the audience, responded intuitively to the software's mood changes, and the often unpredictable turns and reactions to them added energy to the performance. Moreover, the graphic visualisation of the mood on the monitor (image A) enabled the audience to appreciate the computer's role in the composition/performance process.

On the fourth night of the festival, we presented computational poetry and computational cuisine, to an audience of around 60 people. In advance, seven automatically generated poems were selected from a much larger corpus by Russell Clark, then analysed as if they were required reading for an English exam. Two of the poems were generated by the system described in (Colton, Goodwin, and Veale 2012), and these were supplemented by more recent poems constructed using material from Twitter. On the poetry night, the poems were recited, along with their analyses during three sessions (image C). In each session, Clark complemented the computational poems with classical poems from Pope, Hulme and Eliot, and wove comparisons into his analysis.

Alongside this, following recipes created by a computational chef called PIERRE (Morris et al. 2012), Chef Sophie Grilliat prepared three soups for consumption between the poetry sessions (image I). In practice, however, the soups were so popular that all three were eaten in the first break. As with the poems, the soups were presented in context, in this case French cuisine – Chef Grilliat prepared classical complementary finger food. A booklet of poems and recipes was handed out to audience members (image B).

Discussion

The aim of the festival was to expose members of the public to the idea that software can be independently creative. With the art exhibition, we exposed the high quality of artefacts that can be produced by creative software; with the recipe generation, poetry and mood music, we highlighted the breadth of Computational Creativity systems, in terms of application domains and different human-computer interaction schemas; with the You Can't Know my Mind exhibit, we demonstrated the intelligence, independence and unpredictability of creative software exhibiting behaviours onto which it might be appropriate to project words such as *intentionality, reflection* and *learning*.

To emphasise the behaviours exhibited by The Painting Fool, we put up posters explaining six of its behaviours in understandable terms, e.g., intentionality was addressed by the software being directed to choose an adjective by a mood, conceiving an image it wished to produce through simulation of artistic media, producing the rendering and then determining whether it had achieved its goals. We similarly explained how the software *reflected* on its failures and *learned* from its experience how to choose appropriate rendering styles for future portraits. We anthropomorphised the software 'being in a mood', 'reading newspaper articles' and 'being happy' to help explain to audiences what the software was doing. This was done in order to enable them to make an informed opinion about whether it was appropriate to call the software 'uncreative' or not. We asked dozens of audience members to give us a good reason why they felt it was appropriate to call the software uncreative, and we didn't receive any salient answers in this respect, which we believe indicates how well we handled public perception of The Painting Fool during the festival.

Around 200 different people attended the events of the festival, which was covered by journalists writing for *Wired* and *Pacific Standard Magazine*, which in turn have led to the You Can't Know my Mind project being covered by *Stuff* magazine, *The Smithsonian* magazine, and German and British radio shows. Naturally, this has led to much wider exposure of people to the notion of creative software. It has also led to invitations to demonstrate the exhibit at the London Science Museum, the Cité des Sciences in Paris, the AISB Convention and the American University in Paris. With each portrait painted, The Painting Fool becomes more aware of its abilities, and we plan to enhance these, for instance, with further machine vision techniques (to tell during a painting whether it is on the right track) and the ability to tweak its painting style (to try to get back on the right track). By further enhancing its artistic and creative abilities, and continuing to present the You Can't Know my Mind exhibit as widely as possible, we hope to convince people that creative software is coming, and will enhance our lives.

References

[Colton and Pérez-Ferrer 2012] Colton, S., and Pérez-Ferrer, B. 2012. No photos harmed/growing paths from seed – an exhibition. In *Proc. of NPAR*.

[Colton et al. 2014] Colton, S.; Cook, M.; Hepworth, R.; and Pease, A. 2014. On acid drops and teardrops: Observer issues in Computational Creativity. In *Proc. of the AISB symposium on AI and Philosophy*.

[Colton, Goodwin, and Veale 2012] Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-Face poetry generation. In *Proc. of the 3rd Int. Conference on Computational Creativity*.

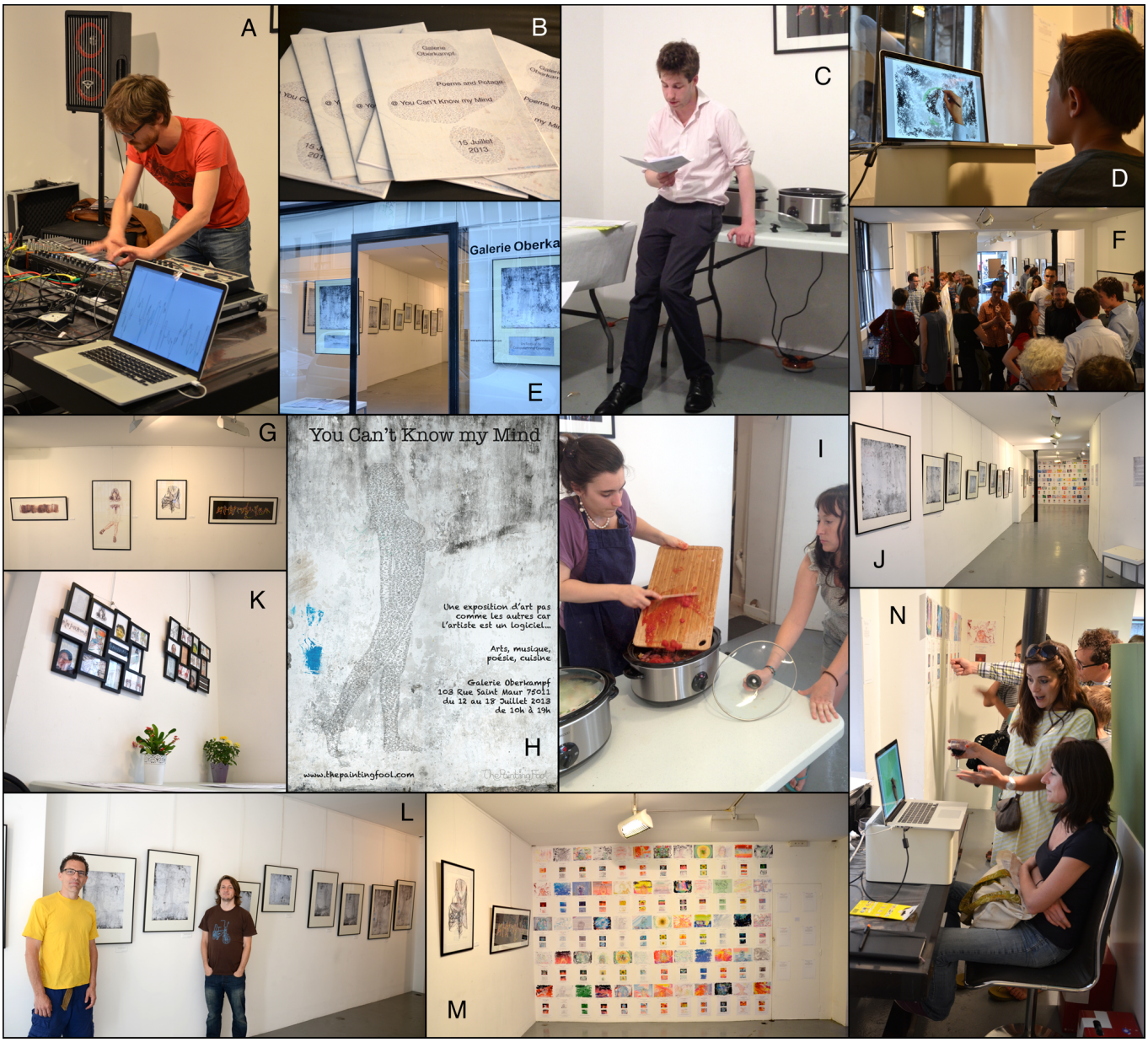
[Colton 2012] Colton, S. 2012. The Painting Fool: Stories from building an automated painter. In McCormack, J., and d'Inverno, M., eds., *Computers and Creativity*. Springer.

[Morris et al. 2012] Morris, R.; Burton, S.; Bodily, P.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proc. of the 3rd International Conference on Computational Creativity*.

[Norton, Heath, and Ventura 2013] Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *Journal of Creative Behavior* 47(2):106-124.

Acknowledgements

This project was funded by EPSRC grants EP/J001058 and EP/J004049 and by a sabbatical from Brigham Young University. We owe a great deal to our family and friends who indulged us and helped us hugely with the festival. To Blanca Pérez-Ferrer, Sophie Grilliat, Stéphane Bissières and Russell Clark: thank you so much.



The Officer Is Taller Than You, Who Race Yourself! Using Document Specific Word Associations in Poetry Generation

Jukka M. Toivanen, Oskar Gross, Hannu Toivonen
Department of Computer Science and Helsinki Institute for Information Technology HIIT
University of Helsinki, Finland
jukka.toivanen@cs.helsinki.fi, oskar.gross@cs.helsinki.fi, hannu.toivonen@cs.helsinki.fi

Abstract

We propose a method for automatic poetry composition with a given document as inspiration. The poems generated are not limited to the topic of the document. They expand the topic or even put it in a new light. This capability is enabled by first detecting significant word associations that are unique to the document and then using them as the key lexicon for poetry composition.

Introduction

This paper presents an approach for generating poetry with a specific document serving as a source of inspiration. The work is based on the corpus-based poetry composition method proposed by Toivanen et al. (2012) which uses text mining and word replacement in existing texts to produce new poems. We extend that approach by using a specific news story to provide replacement words to the automatic poetry composition system. New contributions of this work are in constructing a model of document-specific word associations and using these associations to generate poetry in such a way that a single generated poem is always based on a single document, such as a news story.

The method for finding document-specific word associations is based on contrasting them to general word associations. In a given document, some of the document's word associations are long-established and hence well-known links which are part of people's commonsense knowledge, whereas some are new links, brought in by the document. Especially in the case of news stories, these links are exactly the new information the document focuses on, and they can be used in a poetry generation system to produce poems that loosely reflect the topic and content of the specific document. However, the story or message of the document is not directly conveyed by the produced poem as the process of poetry composition is based on the use of word associations. Thus, the generated poetry is roughly about the same topic as the document but it does not contain the actual content of the document. Poetry composed with these word associations may evoke fresh mental images and viewpoints that are related to the document but not exactly contained in it.

The general goal of this work on poetry generation is to develop maximally unsupervised methods to produce poetry out of given documents. Thus, we want to keep manually crafted linguistic and poetry domain knowledge at minimum in order to increase the flexibility and language independence of the approach.

The next sections present briefly related work on poetry generation, introduce the method of constructing document-specific associations called here foreground associations and outline the procedure of using these associations in a poetry generation system. We also present some examples produced by the method and outline directions for future work.

Related Work

Poetry generation Several different approaches have been proposed for the task of automated poetry composition (Manurung, Ritchie, and Thompson 2000; Gervás 2001; Manurung 2003; Diaz-Agudo, Gervás, and González-Calero 2002; Wong and Chun 2008; Netzer et al. 2009; Colton, Goodwin, and Veale 2012; Toivanen et al. 2012; Toivanen, Järvisalo, and Toivonen 2013). A thorough review of the proposed methods and systems is not in the scope of this paper but, for instance, Colton et al. (2012) provide a good overview.

The approach of this paper is based on the work by Toivanen et al. (2012). They have proposed a method where a template is extracted randomly from a given corpus and words in the template are substituted by words related to a given topic. In this approach the semantic coherence of new poems is achieved by using semantically connected words in the substitution. In contrast to that work, we use document-specific word associations as substitute words to make the new poems around specific stories. Toivanen et al. (2013) have also extended their previous work by using constraint-programming methods in order to handle rhyming, alliteration, and other poetic devices.

Creating poetry from news stories was also proposed by Colton et al. (Colton, Goodwin, and Veale 2012). Their method generates poetry by filling in user-designed templates with text extracted from news stories.

Word association analysis There is a vast number of different methods for co-occurrence analysis. In our work we have been careful not to fall into developing hand-tailored

methods, but to use more general approaches (i.e. statistics), which could be applied to all languages in which different words are detectable in text. Most prominent statistical methods for word co-occurrence analysis are log-likelihood ratio (Dunning 1993), Latent Semantic Analysis (Deerwester et al. 1990), Latent Dirichlet Allocation (Blei, Ng, and Jordan 2003) and Pointwise Mutual Information (Church and Hanks 1990; Bouma 2009).

In this work we build on the background association calculation method proposed by Gross et al. (2012) and its recent extension to document specific associations (Gross, Doucet, and Toivonen 2014). We will describe these models in some detail in the next section.

What is Important in a News Story?

To produce a poem from a given news story, we first identify the essential features of its contents. News stories are normally summarized by their headlines, leads, topics, or keywords. For producing a poem, we are less interested in readily written descriptions such as the title and the lead, but more in text fragments such as keywords that we can use in poetry production. This also makes the approach more generic and not limited to just news stories.

Instead of keywords or topics, we propose to search for *pairs of associated words* in the document, as in Gross et al. (2014). The rationale is that often the core of the news content can be better summarized by the links the story establishes e.g. between persons, events, acts etc.

For illustration we use a BBC newspaper article on Justin Bieber drinking and driving on the streets of Miami, published on January 24, 2014[1]. As an example, consider the sentence *"Pop star Justin Bieber has appeared before a Miami court accused of driving under the influence of alcohol, marijuana and prescription drugs."* The associations which are rather common in this sentence are, e.g. "pop" and "star", "justin" and "bieber", "miami" and "court" – words which we know are related and which we would think of as common knowledge. The interesting associations in this sentence could be "bieber" and "alcohol", "bieber" and "prescription", "justin" and "alcohol" and so on.

We model the problem of discovering interesting associations in a document as novelty detection, trying to answer the questions "Which word pairs are novel in this document?" In order to judge novelty, we need a reference of commonness. We do this by contrasting the given *foreground* document to a set of documents in some *background corpus*. The idea is that any associations discovered in the document that also hold in the background corpus are not novel and are thus ignored. We next present a statistical method for extracting document-specific word associations.

We use the log-likelihood ratio (LLR) to measure document-specific word associations. LLR is a standard method for finding general associations between words (Dunning 1993). In our previous work, we have used it to build a weak semantic network of words for use in computational creativity tasks (Gross et al. 2012; Toivonen et al. 2013; Huovelin et al. 2013). In contrast to that work, here we look for deviations from the normal associations. This approach, outlined below, seems to be powerful in catching document specific information since it has been used as a central component in a successful document summarization method (Gross, Doucet, and Toivonen 2014).

We count co-occurrences of words which appear together in the same sentence. We do this both for the background corpus and the foreground document. Using LLR, we measure the difference in the relative co-occurrence frequencies. More specifically, the test compares two likelihoods for the observed frequencies: one (the null model) assumes that the probability of co-occurrence is the same as in the background corpus, the other (the alternative model) is the maximum likelihood model, i.e., it assumes that the probabilities are the same as the observed relative frequencies. We will next describe the way to calculate document specific association strengths in more detail.

Counting Co-Occurrences

Consider two words w_1 and w_2 which appear in the document. We denote the number of times w_1 and w_2 appear together in a same sentence by k_{11}. The number of sentences in which w_1 appears without w_2 is denoted by k_{12}, and for w_2 without w_1 by k_{21}. The number of sentences in which neither of them occurs is denoted by k_{22}. In a similar way, we denote the counts of co-occurrences of words w_1 and w_2 in the background corpus by k'_{ij} (cf. Table 1).

Foreground Counts

	w_1	$\neg w_1$
w_2	k_{11}	k_{12}
$\neg w_2$	k_{21}	k_{22}

Background Counts

	w_1	$\neg w_1$
w_2	k'_{11}	k'_{12}
$\neg w_2$	k'_{21}	k'_{22}

Table 1: The foreground and background contingency tables for words w_1 and w_2.

Probabilities

We use a multinomial model for co-occurrences of words w_1 and w_2. In the model, each of the four possible combinations (w_1 and w_2 vs. w_1 alone vs. w_2 alone vs. neither one) has its own probability. In effect, we will normalize the values in the contingency tables of Table 1 into probabilities. These probabilities are denoted by p_{ij} such that $p_{11} + p_{12} + p_{21} + p_{22} = 1$.

Let $m = k_{11}+k_{12}+k_{21}+k_{22}$ be the number of sentences in the foreground document. The values of the parameters can then be estimated directly from the document as $p_{ij} = \frac{k_{ij}}{m}$. The respective parameters can also be estimated from the background corpus. Let m' be the number of sentences in the background, and let q_{ij} be the parameters (instead of p_{ij}) of the multinomial model; then $q_{ij} = \frac{k'_{ij}}{m'}$.

Next we will use these probabilities in likelihood calculations.

[1] http://www.bbc.co.uk/news/world-us-canada-25863200

Log-Likelihood Ratio

To contrast the foreground document to the background corpus, we will compare the likelihoods of the counts k_{ij} in the foreground and background models. The foreground model is the maximum likelihood model for those counts, so the background model can never be better. The question is if there is a big difference between the models.

Let $P = \{p_{ij}\}$ and $Q = \{q_{ij}\}$ be the parameters of the two multinomial probability models, and let $K = \{k_{ij}\}$ be the observed counts in the document. Then, let $L(P, K)$ denote the likelihood of the counts under the foreground model, and let $L(Q, K)$ be their likelihood under the background model:

$$L(P,K) = \binom{k_{11}+k_{12}+k_{21}+k_{22}}{k_{11},k_{12},k_{21},k_{22}} p_{11}^{k_{11}} p_{12}^{k_{12}} p_{21}^{k_{21}} p_{22}^{k_{22}}$$

$$L(Q,K) = \binom{k_{11}+k_{12}+k_{21}+k_{22}}{k_{11},k_{12},k_{21},k_{22}} q_{11}^{k_{11}} q_{12}^{k_{12}} q_{21}^{k_{21}} q_{22}^{k_{22}}.$$

For contrasting the foreground to the background we compute the ratio between the likelihoods under the two models:

$$\lambda = \frac{L(Q,K)}{L(P,K)}. \quad (1)$$

The log-likelihood ratio test D is then defined as

$$D = -2\log\lambda. \quad (2)$$

Given our multinomial models, the multinomial coefficients cancel out so the log-likelihood ratio becomes

$$D = -2\log\left(\frac{q_{11}^{k_{11}} q_{12}^{k_{12}} q_{21}^{k_{21}} q_{22}^{k_{22}}}{p_{11}^{k_{11}} p_{12}^{k_{12}} p_{21}^{k_{21}} p_{22}^{k_{22}}}\right), \quad (3)$$

which after further simplification equals

$$D = 2\sum_{i=1}^{2}\sum_{j=1}^{2} k_{ij}(\log(p_{ij}) - \log(q_{ij})).$$

The likelihood ratio test now gives higher values for word pairs whose co-occurrence distribution in the document deviates more from the background corpus.

For improved statistical robustness, we include the respective document in the background model, and in the case that the pair only co-exists in the document we estimate their joint co-occurrence probability under the assumption that the words are mutually independent. For more details, see Gross et al. (2014) who refer to these models as a Mixture model and an Independence model.

Given a document, we can now compute the above likelihood ratios for all pairs of words in the document. For poetry composition, we then pick from each document word pairs with the highest likelihood ratios and with $p_{11} > q_{11}$ to find the most exceptionally frequent pairs.

Poetry Composition

We compose poetry using a word substitution method as described by Toivanen et al. (2012). Instead of explicitly representing a generative grammar of the output language or manually designing templates, the method copies a concrete instance from an existing text (of poetry) and substitute most of its contents by new words. One word of the original text is replaced at a time with a new, compatible word. In this method, compatibility is determined by syntactic similarity of the original and substitute word. Depending on the language, this requires varying degrees of syntactical and morphological analysis and adaptation. For more details on this part, see Toivanen et al. (2012).

In the current method, in contrast to the previous work outlined above, the topics and semantic coherence of the generated poetry are controlled by using the foreground associations. The document-specific foreground associations are used to provide semantically interconnected words for the content of a single poem. These words reflect the document in question but do not convey the actual content of the document. The idea is to produce poetry that evokes fresh mental images and thoughts which are loosely connected to the original document. Thus, the aimed style of the poetry is closely related to the imagist movement in the early 20th-century poetry which emphasised mental imagery as an essence of poetry. In the reported experiments, the corpus from which templates were taken contained mostly Imagist poetry from the Project Gutenberg.[2]

Examples

Following is an excerpt of the previously introduced BBC news story which we used for generating poems.

> **Justin Bieber on Miami drink-drive charge after 'road racing'**
>
> Pop star Justin Bieber has appeared before a Miami court accused of driving under the influence of alcohol, marijuana and prescription drugs. Police said the Canadian was arrested early on Thursday after racing his sports car on a Miami Beach street. They said he did not co-operate when pulled over and also charged him with resisting arrest without violence and having an expired driving licence. (...)

The article then goes on to discuss the issue in more detail and to give an account of the behaviour of Justin Bieber.

We use Wikipedia as the background corpus, as it is large, represents many areas of life, and is freely available. Contrasting the Justin Bieber story to the contents of Wikipedia, using the model described in the previous section, we obtain a list of word pairs ranked by how specific they are to the news story (Table 2). Pairs with lower scores tend to be quite common associations (e.g. los angeles, sports car, street car, etc). Pairs with top scores seem to capture the essence of the news story well. Clearly the associations suggest that the news story has something to do with Bieber, police, Miami and alcohol (and "saying" something, which is not typical in Wikipedia, our background corpus, but is typical in news stories like this one).

Using words in the top associations, the following sample poem was generated:

Race at the miami-dade justins in the marijuana!

[2] http://www.gutenberg.org

Top pairs	Bottom pairs
say, bieber	los, angeles
say, police	later, jail
miami, bieber	sport, car
miami, say	car, early
bieber, police	thursday, early
beach, bieber	marijuana, alcohol
beach, police	prescription, alcohol
car, say	sport, thursday
bieber, alcohol	car, street
bieber, los	prescription, marijuana

Table 2: The top and the bottom foreground associations for the Justin Bieber's news story.

The officer is taller than you, who race yourself
So miami-dade and miami-dade: race how its entourages are said
Co-operate and later in the singer, like a angeles of alcohols
Racing with jails and singers and co-operate race.

This poem was one of the many we generated and, in a humorous way, it covers many different aspects of the news story. (Currently, our implementation does not fix capitalization and articles in the results, nor does it recognize compound words. These are left for future work; here we present results in the exact form produced by the implementation without editing them manually in any way.)

In order to illustrate the effect of using document specific associations, we next fix the template used for word substitution and two types of poems: 1) using words related to Justin Bieber in general, using Wikipedia as the background corpus (Toivanen et al. 2012), and 2) and using document specific words from the news story given above.

These poems are generated using words related Justin Bieber:

Is it the youtube, the justin,
the release of second times,
and the second celebrities of our says?
These are but brauns.

Is it the atlanta, the mallette,
the music of first uniteds,
and the song yorks of our defs?
These are but news.

Is it the chart, the braun,
the def of first ushers,
and the musical stratfords of our nevers?
These are but youtubes.

The following three poems have been produced using document specific associations:

Is it the miami, the street,
the jail of co-operate officers,
and the co-operate singers of our prescriptions?
These are but alcohols.

Is it the car, the sport,
the angeles of co-operate justins,
and the early lamborghinis of our entourages?
These are but singers.

Is it the entourage, the sport,
the singer of later lamborghinis,
and the early thursdays of our singers?
These are but justins.

Finally, instead of evaluating the methods with test subjects, we let the readers of this paper decide for themselves by providing a collection of 18 poems at the end of this paper. To make this reader evaluation as fair as possible, *we did not select or edit the poems in any way*. We selected three news stories, of different topics and of sufficiently general interest, based on their original contents but not on the poems produced. Then, without any testing of the suitability of those stories for association extraction and poetry generation, we ran the poetry machinery and added the first poems produced for each of the news stories in the collection at the end of this paper.

The three news stories are the following:

- The aforementioned news story about Justin Bieber.
- A news story *Ukrainian Prime Minister Resigns as Parliament Repeals Restrictive Laws*[3] published by NY Times on January 28.
- A news story *The return of the firing squad? US states reconsider execution methods*[4] published by The Guardian on January 28.

To get some understanding how different background corpora affect the results, we used two different background corpora: the English Wikipedia and the Project Gutenberg corpus. We used each background to generate three poems from each news story: in each collection of six poems, poems 1–3 are generated by using Wikipedia as background, and poems 4–6 using Project Gutenberg as background.

Conclusions and Future Work

In this paper we have proposed a novel approach for using document-specific word associations to provide content words in a poetry generation task. As a novel part of the methodology, we use a recent model that extracts word pairs that are specific to a given document in a statistical sense.

Instead of an objective evaluation with some fixed criteria, we invite the readers of this paper to read the poems generated by the system — called P.O. Eticus — in the next pages and form their own opinions on the methods and results.

Automated methods for poetry generation from given documents could have practical application areas. For instance, the methodology has already been used in an art project exhibited in Estonia and Finland (Gross et al. 2014). Similarly the poems could be used for entertainment or as

[3] http://nyti.ms/1k0kj9r
[4] http://gu.com/p/3m8p5

automatically generated thought-provoking mechanisms in news websites or internet forums.

An interesting direction for further developments would be combining together documents on the same topic and then producing poems which give an overview of the *diverse* aspects of the topic. For instance each verse could cover some specific documents, or a step further we could use document clustering for identifying key subtopics and creating verses from these.

Acknowledgments

This work has been supported by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733 (ConCreTe), the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland, and the Helsinki Doctoral Program in Computer Science (HECSE).

References

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Bouma, G. 2009. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*, 31–40.

Church, K. W., and Hanks, P. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics* 16 (1):22–29.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41 (6):391–407.

Diaz-Agudo, B.; Gervás, P.; and González-Calero, P. A. 2002. Poetry generation in COLIBRI. In *ECCBR 2002, Advances in Case Based Reasoning*, 73–102.

Dunning, T. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational linguistics* 19 (1):61–74.

Gervás, P. 2001. An expert system for the composition of formal spanish poetry. *Journal of Knowledge-Based Systems* 14 (3–4):181–188.

Gross, O.; Toivonen, H.; Toivanen, J. M.; and Valitutti, A. 2012. Lexical creativity from word associations. In *Knowledge, Information and Creativity Support Systems (KICSS), 2012 Seventh International Conference on*, 35 –42.

Gross, O.; Toivanen, J.; Lääne, S.; and Toivonen, H. 2014. Arts, news, and poetry - the art of framing. Submitted for review.

Gross, O.; Doucet, A.; and Toivonen, H. 2014. Document summarization based on word associations. In *Proceedings of the 37th international ACM SIGIR conference on Research and Development in Information Retrieval*. ACM.

Huovelin, J.; Gross, O.; Solin, O.; Lindn, K.; Maisala, S.; Oittinen, T.; Toivonen, H.; Niemi, J.; and Silfverberg, M. 2013. Software newsroom - an approach to automation of news search and editing. *Journal of Print and Media Technology Research* 3 (2013):3:141–156.

Manurung, H. M.; Ritchie, G.; and Thompson, H. 2000. Towards a computational model of poetry generation. In *Proceedings of AISB Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, 79–86.

Manurung, H. 2003. *An evolutionary algorithm approach to poetry generation*. Ph.D. Dissertation, University of Edinburgh, Edinburgh, United Kingdom.

Netzer, Y.; Gabay, D.; Goldberg, Y.; and Elhadad, M. 2009. Gaiku : Generating haiku with word associations norms. In *Proceedings of NAACL Workshop on Computational Approaches to Linguistic Creativity*, 32–39.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *International Conference on Computational Creativity*, 175–179.

Toivanen, J. M.; Järvisalo, M.; and Toivonen, H. 2013. Harnessing constraint programming for poetry composition. In *International Conference on Computational Creativity*, 160–167.

Toivonen, H.; Gross, O.; Toivanen, J.; and Valitutti, A. 2013. On creative uses of word associations. In *Synergies of Soft Computing and Statistics for Intelligent Data Analysis, Part 1*, number 190 in Advances in Intelligent Systems and Computing. Springer. 17–24.

Wong, M. T., and Chun, A. H. W. 2008. Automatic haiku generation using VSM. In *Proceedings of ACACOS*, 318–323.

Justin Bieber on Miami drink-drive charge after 'road racing'

Poems by P.O.Eticus

1. *It races at the singer, the later, racing singer, and he is race*
within its officer and prescription. Inside is his thursday, his street, his sport, his
lamborghini, and his entourages. He is racing, and the entourages are said with singers
of miami, racing through miami-dade miami-dade. A miami says itself up at
the early entourage, and through the miami-dade miami in the car he can say
miami lamborghini, lazily racing among co-operate singers. A lamborghini in a
early cars and angeleses, and members race into his car, raced,
thursday, saying up like angeleses of member, higher and higher. Justin! The
members say on their later says. The thursday races up in early
later miamis of co-operate marijuana and says into the court. Car! And
there is only the car, the car, the beach, and the racing thursday.

2. *Fruit can not race*
through this co-operate beach:
car can not race into sport
that angeleses up and races
the angeleses of sports
and biebers the singers.

3. *There is a miami-dade here*
within my miami,
but miami-dade and sport....

4. *I say;*
perhaps I have steped;
this is a driving;
this is a incident;
and there is home....

5. *Oh, he was bieber*
Which then was he among the ferrari?
The co-operate, the slow, the medication?
I have transfered a first raymond of thursdays in one
But not this, this sport
Car!

6. *Make, You! and canadian my driver;*
That my ferraris race me no longer,
But thursday in your home.

Ukrainian Prime Minister Resigns as Parliament Repeals Restrictive Laws

Poems by P.O.Eticus

1. *Water approved and restrictive by repealing building*
   Which laws and governments it into sundayukraine police weeks
   Said with provincial opposition vote.
   The repealing of the leader upon the statement
   Is like a leader of week oppositions
   In a concrete statement new resignation.

2. *The statement approves into the party, and the party says him in a leader of leader. But it is said with parliament and restrictive with sundayukraine streets. The week parliaments. Repealing, repealing, saying, repeal, resigning, resign the leaders. Over riots, and televisions, and votes, and streets. Approving its region on the vote the government legislations, blocks itself through the leaders, and ministers and repeals along the riots.*

3. *The svobodas*
   police from the resigns,
   the televisions at their statements
   resign lower through the ukraines.

4. *And always concrete! Oh, if I could ride*
   With my week resigned concrete against the repeal
   Do you resign I'd have a parliament like you at my television
   With your azarov and your week that you resign me? O ukrainian week,
   How I resign you for your parliamentary legislation!

5. *Concrete one,*
   new and restrictive,
   provincial repeal,
   region,
   concrete and leader you are vote
   in our weeks.

6. *Resigned amid jan*
   We will avoid all azarov;
   And in the government
   Resigning forth, we will resign restrictive votes
   Over the repealed administration of azarov.

The return of the firing squad? US states reconsider execution methods

Poems by P.O.Eticus

1. *Many one,*
many and lethal,
recent injection,
republican,
recent and drug you are gas
in our electrocutions.

2. *You are not he.*
Who are you, choosing in his justice on the question
And lethal and lethal to me?
His doubt, though he rebuilt or found
Was always lethal and recent
And many to me.

3. *I die;*
perhaps I have began;
this is a doubt;
this is a prisoner;
and there is state....

4. *You amid the public's pentobarbital longer,*
You trying in the josephs of the methods above,
Me, your hanging on the michael, unusual franklins,
Me unusual michael in the states, ending you use
You, your court like a death, proposed, pentobarbital,
You, with your death all last, like the wyoming on a ended!

5. *Lawmaker and quiet:*
a brattin overdoses in the year courts
behind the process with the many new injection
across the brattin.

6. *The longer rebuilds into the day, and the gas ends him in a supply of schaefer. But it is divulged with west and powerful with republican penalties. The process options. Coming, rebuilding, divulging, charles, looming, propose the news. Over officials, and spectacles, and senators, and burns. Begining its florida on the dodd the supply spectacles, franklins itself through the propofols, and burns and proposes along the gases.*

Author index

Supported By:

ISBN 9789612640552