Experience and Creativity

Workshop at the
Twenty-Third International Conference on
Case-Based Reasoning
(ICCBR 2015)

Frankfurt, Germany
September 2015

Raquel Hervás and Enric Plaza (Editors)

Chairs

Raquel Hervás	Universidad Complutense of Madrid, Spain
Enric Plaza	IIIA-CSIC, Spain

Program Committee

Kathleen Agres	Queen Mary, University of London, UK
Amílcar Cardoso	Univesidade de Coimbra, Portugal
Belén Díaz-Agudo	Universidad Complutense of Madrid, Spain
Pablo Gervás	Universidad Complutense of Madrid, Spain
Maarten Grachten	Austrian Research Institute for Artificial Intelligence, Austria
Maria Teresa Llano	Goldsmiths, University of London, UK
Ramon López de Mántaras	IIIA-CSIC, Spain
Santiago Ontañón	Drexel University, USA
Rafael Pérez y Pérez	Universidad Autónoma Metropolitana, Mexico
Senja Pollak	Josef Stefan Institute, Slovenia
Hannu Toivonen	University of Helsinki, Finland
Martin Znidarsic	Josef Stefan Institute, Slovenia

Additional Reviewers

Pascal Reuss	University of Hildesheim, Germany
Viktor Ayzenshtadt	University of Hildesheim, Germany

Preface

We are pleased to present the proceedings of the ICCBR-15 Workshop on Experience and Creativity, which was held as part of the 23rd International Conference on Case-Based Reasoning in Frankfurt am Main, Germany, September 2015 with the cooperation of PROSECCO, the European Network for Promoting the Scientific Exploration of Computational Creativity.

The relationship between past examples in a domain and computational creativity in that domain is an interesting and essential topic that has not been explicitly addressed. The goal of the workshop is to address common areas of interest in Case-Based Reasoning (CBR) and Computational Creativity (CC) by addressing research issues that are related to both communities. In order to do so, our goal is to explore and analyze how the new and innovative (creativity) is related to, depends upon, and needs to break away to from the old and know (experience). The main focus of the workshop will be on exploring the relationship between past examples in a domain and computational creativity in that domain.

The workshop provided a forum for discussion of these new research directions, provoked by the presentation of five long papers and four position papers, which are collected in these proceedings. Geraint Wiggins gave an invited talk for the workshop on the relation of experience and creativity. Znidarsic, Tomašič and Papa presented a case-based approach to automated generation of slogans, including a methodology for evaluation and ranking of the final results, which indicate the ability of the approach to create valuable slogan prototypes. Gervás, Hervás and León presented a case-based reasoning solution that builds a plot line to match a given query, expressed in terms of a sequence of abstraction of plot elements of a story, by retrieving and adapting templates for narrative schemas from a case-base. Hervás, Sánchez-Ruiz, Gervás and León compared the judgement on similarity between stories explained by a human judge with a similarity metric for stories based on plan refinements, taking into account that is difficult to compute between complex artifacts such as stories. Gonçalves, Martins, Cruz and Cardoso proposed an evolutionary high performance algorithm that extracts two semantic sub-graphs from a knowledge base to be used as building blocks in computational blending processes. Pollak, Martins, Cardoso and Urbancic investigated which principles people use when they name new things as results of blending, with the aim of uncovering patterns with high creative potential and to use them for automated generation of names for new creations or phenomena. Valitutti discussed ideas for characterizing the re-use of procedural knowledge, performed by a case-based generative system, as creative. The implied idea is to characterize as creative the search path that allows the system to discover new basins of attraction. Agres stated that there is a clear connection to be made between psychological findings regarding learning and memory and the areas of case-based reasoning and computational creativity, aiming to encourage researchers in these areas to consider psychological perspectives while developing the technical and theoretical aspects of their computational systems. Cardoso and Martins proposed that conceptual blending, an important mechanism in computational creativity, can play a role within the case-based reasoning paradigm as an alternative adaptation mechanism that may provide suitable solutions in computational creativity setups. Cunha, Martins, Cardoso and Machado focused on computational generation of visual symbols to represent concepts, aiming to develop a system that uses background knowledge about the world to find connections among concepts with the goal of generating symbols for a given concept.

Finally, we would like to thank everyone who contributed to the success of this workshop, especially the authors, the program committee members, PROSECCO, the organizers of the ICCBR 2015 conference and Joseph Kendall-Morwick, ICCBR Workshop Chair.

September 2015 Raquel Hervás
Frankfurt Enric Plaza

Automated blend naming based on human creativity examples

Senja Pollak[1], Pedro Martins[2], Amílcar Cardoso[2], and Tanja Urbančič[13]

[1] Jožef Stefan Institute, Ljubljana, Slovenia
[2] CISUC, DEI, University of Coimbra, Coimbra, Portugal
[3] University of Nova Gorica, Nova Gorica, Slovenia
senja.pollak@ijs.si, pjmm@dei.uc.pt, amilcar@dei.uc.pt
tanja.urbancic@ung.si

Abstract. In this paper we investigate which principles people use when they name new things as results of blending. The aim is to uncover patterns with high creative potential and to use them for automated generation of names for new creations or phenomena. We collected examples with a web survey in which participants were asked to evaluate pictures of animals with blended anatomies from two different animals, and to provide their own names for blended creatures on the pictures. The blended animals served as a trigger of human creativity manifested through imaginative, humorous, surprising names collected in the survey. We studied how the features from the pictures reflected in the names, what are different complexity levels of lexical blend formation and how far in other realms subjects "travelled" to search for associations and metaphors used in the names. We used the findings to guide automated generation of names for the blends.

Keywords: Computational creativity, human creativity examples, conceptual blending, lexical blend generation, creative naming, bisociation.

1 Introduction

Creativity is in the core of many human activities and has been studied for decades [9][2]. As a phenomenon challenging for being replicated with machines, it became also a topic of artificial intelligence research [21]. While creativity is an intriguing research question by itself, it is also a driving force of development and as such, it has an immense value for applications in countless areas, including scientific discovery, engineering inventions and design. One of the cognitive principles underlying such discoveries and inventions is conceptual blending [5] in which two mental spaces integrate into a new one, called blend. Conceptual blending has also been implemented and tested in computer systems to produce novel concepts [17]. However, there are still many open questions related to the choice of input mental spaces and the ways of projections that lead to blends, perceived as creative and inspiring. In our work we aim at providing guidance for choosing input spaces and projections based on concrete findings about human creativity with elements of blending. More precisely, by investigating the patterns that we can find in the cases of human creations, we guide the blending process to the extent allowing for automated generation of blends.

Conceptual blending and case-based reasoning [10] can meet in a very fruitful way in areas such as design and architecture [4][6].In such domains, blends are not only a source of surprise, artistic satisfaction or inspiration, but have also their own functionality, bringing into the process additional constraints and priorities. Contexts and goals can also be used in computational approaches to conceptual blending and can beneficially affect the issues of efficiency [13]. Authors in [1] exploit a principle of creative transfer from one domain to another in the realm of design. Their IDEAL system abstracted patterns from design cases in one domain and applied them to design problems in another domain. connecting distant, self-consistent and usually not connected frames of reference has been recognised and used as an effective principle in the act of creation. Such connections of habitually incompatible domains through common patterns or bridging concepts are also referred to as bisociations [9].

In this paper, we address the issue of case-based reasoning and conceptual blending in the context of lexical creativity. While this might appear quite far from the discussion on design in the previous paragraph, the connection becomes evident based on an observation by Veale and Butnariu [20]: "Words are everyday things, as central to our daily lives as the clothes we wear, the tools we use and the vehicles we drive. As man-made objects, words and phrases are subject to many of the same design principles as the consumer artefacts that compete for our attention in the market-place.". The authors also draw attention to two basic principles of artefact design, as identified in [15], namely visibility and mapping. In the case of a well-designed product, the design should suggest a mental visualisation of a conceptually correct model of the product, and the mapping between appearance and function should be clear. Their Zeitgeist system [20] can automatically recognise neologisms produced as lexical blends, together with their semantic meaning. This is done based on seven different "design patterns" recognised in constructing neologisms as lexical blends. Types of lexical blends and how new lexical blends are formed is described and illustrated with many examples in [12]. An important issue of recognising and quantifying creativity in different combinations of words is studied in [11].

In our work we investigate how humans approach the task of naming new things, and how based on human examples, a computer system could exhibit similar (and, why not, better) performance. We consider this principle of using past examples for revealing patterns to be used for new cases as a manifestation of case-based reasoning. The concrete task was to name creatures – animals with blended anatomies from two different animals. This was done in a web-based survey, designed primarily for a study of human perception of visual blends [14]. In this paper we continue using the material of the same study, but we examine it from a completely different angle, i.e. from the lexical creativity side by investigating creative naming of blends. Many offers for supporting naming of

snorse (snake, horse)
chimporse (chimpanzee, horse)
durse (duck, horse)
guineabear (guinea pig, bear)
hammerheadhorse (hammerhead shark, horse)
pengwhale (penguin, whale)
proboscis parrot (proboscis monkey, bird)
chamelephant (elephant,chameleon)
duckphant (elephant, duck)
guinea lion (guinea pig, lion)
horbit (horse, rabbit)
hammerhead gull (hammerhead shark, gull)
horduck (horse, duck)
spider pig (spider, guinea pig)
shark retriever (shark, labrador retriever)

Fig. 1. Hybrid animals dataset used in the online questionnaire (available at http://animals.janez.me). Each sub-caption contains a name of the blend proposed by survey participants, as well as the input spaces. All blends were created by Arne Olav, with the exception of *shark retriever* and *camalephant*, whose authorship is unknown. For a better visualisation, some images were slightly cropped.

client's enterprises, products, etc. can be found on the web and show the application potential of creative naming. The task has already been approached with the goal of (semi-)automatic name generation and the results presented in [16] and [18] demonstrate a very big potential. While our work shares some of the ideas with above-mentioned related approaches, it differs from them by using visually triggered human examples as examples used for automatic lexical blend generation, and by using a novel categorisation of creativity level that guides construction of blends based on bisociation as one of the key principles inherent in many human creative processes.

After presenting the survey in which the names were collected in Section 2, we analyse different patterns and mechanisms used by people when coining names in form of lexical blends in Section 3. These patterns are used in Section 4 for automatically generating blends of different levels. In Section 5 we discuss the potential of our prototype and present further research perspectives.

2 Survey: Visual blends and their lexical counterparts

In [14], we introduced a survey consisting of an on-line questionnaire related to the quality of visual blends. Around 100 participants assessed 15 hybrid animals which were the result of blending anatomies from two different animals (Figure 1).The participants were asked to to rate criteria related to the coherence of blends as well as creativity.

Clearly in our questionnaire on animal blends the main focus was on visual blends. However, with the aim of getting more insight into potential connections, participants were also asked to provide a name (in English, Portuguese, Slovene, French or Spanish) to each of the hybrid creatures. By asking people to name the creatures we wanted to investigate the following questions: Would participants give names for all, for none, or for some of the creatures? How creative are they when naming the animals, how does the visual blended structure reflect in the lexical blend? Where the names provided by subjects mostly lexical blends or not? Do lexical blends use animal's "prototype" characteristics, or more sophisticated associations for which some background knowledge is needed (like titles of books, movies, history, etc.)? Does complexity of visual blends reflect in the names? The names given to the visual blends are the focus of our study.

In our survey we collected 1130 names for 15 animals. The general trend was that people gave more names at the beginning of the study and the trend of the number of given names was descending. However, some pictures triggered more generated names than expected by their position (e.g., *guinea lion* and *spider pig*). The guinea lion is also the blend for which the unpacking (recognising the input spaces) was the most difficult [14] and the one for which the highest number of very creative, bisociative lexical blends were formulated.

3 Formation and complexity of lexical blends

Our previous investigations of relationship between conceptual blending and bisociation have drawn our attention to different levels of blend complexity. To deal with this issue in a more systematic way, we suggest the following categorisation regarding the input words used to form the name:

L1 each of the words appearing in the lexical blend is a commonly used word for one input animal (no mapping);
L2 both input words represent input animal in a rather common way, but are blended into one word by *portmanteau* principle, i.e. by using the prefix of one word and the suffix of the other word (possibly with some intersection);
L3 one word represents one input animal with a commonly used word for this animal, the other word represents a *visible characteristic* (part, colour etc.) of the other animal (variant L3*: both words use such characteristics);
L4 one word represents one input animal with a commonly used word for this animal, the other word represents a characteristic of the other animal for which *background knowledge* about this animal (habitat, way of moving, typical behaviour) is needed (variant L4*: both words use such characteristics);
L5 one word represents one input animal with a commonly used word for this animal, the other animal is represented with a more sophisticated association – *bisociation* – for which a creative discourse into another realm (e.g. from animals to literature) is needed (variant L5*: both words represented with such associations).

We illustrate the categories by the names actually given in the survey to the blended animal *guinea bear*:

L1 *mouse-bear* (input1: *mouse*, input 2: *bear*);
L2 *rabbear* (input1: *rabbit*, input 2: *bear*);
L3 *small-headed bear* (input1: *mouse* → *small head*, input 2: *bear*);
L4 *scared bear* (input1: *mouse* → *scared*, input 2: *bear*);
L5 *mickey the bear* (input1: *mouse* → *Mickey the mouse*, input 2: *bear*).

As seen from this example, while the bear was easily recognised as one of the constituting animals, there were different interpretations about the second animal, "contributing" the head to the blended creature. In fact, the variety in the whole dataset was even bigger as names given by different subjects suggested the second animal being a mouse, rabbit, hamster, guinea pig, rat, squirrel, wombat or opossum. The set of input words as used by the subjects is even bigger since it includes also diminutives, slang versions, etc.

The levels increasing indicate the increasing complexity (but not necessarily the quality) of the blends, but note that they do not build on just one criterion in a linear way and there might also be a combination of principles described at different levels present in one name. We illustrate this with a name *teddybbit*, generated as a portamanteau (L2), but using an association between bear and teddybear from the toys realm (L5).

However, we plan to improve this by introducing a creativity score in which not just the level of mappings used will be taken into account, but also the fact whether they were used for one or for both input animals, and how creative the combination was (e.g., by taking into account phonetic features or by recognizing references extrinsic to the two input animal spaces and their bisociations).

Note that not all of the names provided by the subjects in the survey were lexical blends. Here we do not analyse such names in more detail, but to study the potential for triggering creativity, they are important as well. Some examples collected in our survey for the *guinea bear* are *creepy*, *giant*, or *fluffy*.

4 Patterns from examples for automated name generation

We investigated how the above-mentioned categories of human-generated names could be used for automatic blend generation. Different categories represent different mechanisms. Names of Level 1 are very basic and easy to be automatically generated, their creativity level is low and the name can hardy be called a blend. On the other hand, higher levels (3-5) rely on human experience, background knowledge, associations and bisociations. To generate the names of levels 3 and 4, we use a large web corpus (the enTenTen corpus [7]) and the sketch grammar relations available in Sketch Engine [8]. For the last category (level 5), we used other resources of human knowledge (Wikipedia, imdb lists). For each category, we reveal the patterns in human given names and explain how they can be used in automatic generation. Our generated examples are all done by modifying only one animal name.

L1: In names given by humans, we found two different patterns at level 1. In each case, the two animals are used, the possible variations being either hyphen to indicate the combined meaning "animal1-animal2" (e.g. *dog-shark*) or creating a single word containing full names of both animals "animal1animal2" (e.g. *spiderrat*). The pattern with a premodifier of adjective can be recognised in the given name *mišasti medved*, where the first word is an adjective formed from the noun *miš* (Eng. *mouse*) and the second one is the noun medved (Eng. *bear*). Some word formations are language specific, e.g. in Slovene bare "noun-noun" word formation is not very productive.

To illustrate the automatic name generation, we took the animal names from each input space and concatenated them. Using these simple patterns resulted in names very much resembling those generated by humans, e.g. *duck-horse* or *duckhorse*. More examples are in the L1 row in Table 1.

L2: Level two uses the *portmanteau* principle. In all the languages used in the survey this mechanism was used very frequently. For recognising these names from the list, we focused on words composed of the beginning of one animal word and ending of the other. Examples of basic portmanteau names given by the subjects are the names given in Figure 1. We automatically recognized L2 blends by combining pairs of animals and some simple heuristics.

In automatic generation, the starting point was to combine half of the each of the two input animal names. If the input word consists of two words, frequently

Table 1. Automatically generated names - examples for four fictional animals.

Input level	elephant & chameleon	snake & horse	horse & chimpanzee	duck & horse
L1	elephant-chameleon	snake-horse	horse-chimpanzee,	duck-horse
	elephantchameleon	snakehorse	horsechimpanzee	duckhorse
L2	elepheleon	snarse	horanzee	ducrse
L3	tusk chameleon	venom horse	hoof chimpanzee	beak horse
	trunk chameleon	fang horse	mane chimpanzee	arse horse
	graveyard chameleon	tail horse	bridle chimpanzee	back horse
	tail chameleon	poison horse	rump chimpanzee	feather horse
	ear chameleon	belly horse	withers chimpanzee	
L4	Asian chameleon	venomous horse	Trojan chimpanzee	Anaheim horse
	giraffe chameleon	poisonous horse	wild chimpanzee	lame horse
	captive chameleon	garter horse	Arabian chimpanzee	Peking horse
L5	Dumbo chameleon	Ser Hiss horse	Alfonso chimpanzee	Donald horse
			Daffy horse	Howard the horse

in the analysed examples one word is kept to from the blended name (which is not a proper portmanteau anymore). This pattern was used for generating examples like *guinea lion*, *hammerhead eagle*, *hammerhead goose*.

One could make different combinations based on different proportions of the input words or by using phonetic rules (vowels, consonants, rhymes), exact vs. inexact matching, pronunciation information, word's Greek or Latin origins, etc. as in many advanced existing systems proposing portmanteau name generation [19] [18] [3].

L3: In the next category of lexical blends, humans use visible characteristics of one animal and associate them to the other animal. The properties of the animal that gives the "head" to the new visual blend can be lexically expressed as prepositional phrase modifying the head noun, i.e. the name of the animal providing the body (*horse with snake head, elephant of the orange beak*), by adjective modifier (e.g. *nosy robin, duckbilled pachyderm, trunkheaded chameleon)*or in noun-noun constructions (e.g. *nosebird*). In some cases both animals are described by their characteristic visible parts (e.g. *tail-trunk*). Combinations with portmanteau structure is also possible (e.g. *grivasti kabod* [Eng. *mane horswan*]).

For automated blend generation of L3 we currently use only noun-noun constructions. We rely on the Skecth Engine tool by using word sketches constructed with Sketch grammar. Word sketches are automatic corpus-derived summaries of a word's grammatical and collocational behaviour [8]. From the word sketch of animal "contributing" the head to the visual blend (e.g. *elephant* in Figure 1), we use all the collocators (above selected frequency and salience threshold) from the grammatical category *possessed*. This lists contains nouns that in the enTenTen corpus follow the search word and 's, e.g. for *elephant's* the list contains *tusk, trunk, ...* resulting from collocation *elephant's* <u>*tusk*</u> in the corpus. We construct then noun-noun blends, by adding the animal name of the animal providing the body (e.g. *chameleon*). As shown in Table 1, examples using this structure often correspond to parts of the body, (*tusk chameleon, trunk chameleon, tail chameleon, ear chameleon*), while *graveyard chameleon* does not represent the part of the body. Obviously, some of the compounds are irrelevant, e.g. *tail chameleon* – since chameleons have a tail themselves so this description does not contribute anything in terms of blending. Neither does the corpus provide the information if the "possessed" part is located on the animal's head and even less if it corresponds to the depicted picture (e.g. tusks are not depicted on the picture of elephant and chameleon from Fig. 1, even if they are prototypical part of elephant's head). More specific filters and knowledge bases will be used in future to narrow the choice to better candidates.

L4: Level 4 names are more diverse and require more background knowledge. As mentioned in Section 3, the observed categories are habitat, locomotion (*plavajoči konj* [Eng. *swimming horse*], typical behaviour (e.g.*elequack* using animal sounds) or usage (*saddleducks*. Again, also both animals can be represented by their properties, such as in the blended name *galloping quack*. For automated name generation at this level, we used again the word sketches, but we took the information from category *modifiers* (typical adjectival or noun collocators modifying the animal providing the head to the blended creature). E.g. adjectives *venomous* and *poisonous* are typical collocators of word *snake* and are used for forming blended names *venemous horse* and *poisonous horse*. Often breed names are used in modifier position; by selecting only lower case modifiers we can keep more general properties. For Level 4 , more background knowledge is needed. E.g., from automatically constructed names *Trojan chimpanzee, wild chimpanzee* or *Arabian chimpanzee*, the first one is referring to specific cultural reference Trojan horse and can be interpreted at level 5. Same goes for the *lame horse*, which is formed from the idiom *lame duck* (i.e.*an elected official who is approaching the end of his tenure, and esp. an official whose successor has already been elected* (Wikipedia)).

L5: In analysis of human lexical blends we manually classified in Level 5 the bisociative blends using characters from cartoons (*Spider Gonzalez*), children songs (*Slonček Raconček* refering to a Slovene song *Slonček Jaconček*), where slonček means small elephant and *raconcek* comes from *duck – raca*), movies (*My little mallard*), politicians (*Sharkozy*), legends (*Jezerski Pegasus* [Eng. *river Pegasus*]) and often combinations of several of them, e.g. character from movie and from comic strips *Jumbo Zvitorepec* (where Jumbo refers to the animal, while *Zvitorepec* is a character from Slovene comic strip by Miki Muster, but literally means curled tail which refers also to the visual representation of this animal (cf. picture elephant, chameleon in Fig. 1).

For automatically generating highly creative lexical blends inspired by the examples given by participants, we based the bisociative blend generation on characters from the movies representing the input animal. We created a short list from Wikis, IMDB and Wikipedia pages about animal characters in movies where the last section covers cultural representations. In the name generation process, we first checked if character's name contains the name of the animal and if so we substituted this name with the name of the other input animal (e.g. horse substituting the duck in *Donald horse*). On the other side, if the animal does not appear explicitly we added the name of the second animal to the existing character name (*Dumbo chameleon*). In future, we will expand generation of names at this level by exploring other realms besides movies and books.

5 Discussion

We investigated the principles of creating lexical blends based on visual blends (blended animals). We revealed different mechanisms used in name formation and introduced a new categorisation of blend complexity (L1-concatenation blends; L2-portmanteaux; L3-blending based on visible characteristics; L4- blending using background knowledge and L5-bisociative blends). After the analysis of examples generated names by humans, we made a prototype system for automated generation of blends of different levels using word combinations, grammatical and collocational information and background knowledge resources. The most frequent mechanism used by humans was the portmanteau principle. But a portmanteau can vary from very basic ones to the bisociative ones, since blend strategies can easily be combined. For instance, the blend *shagull* can be interpreted as a simple portmanteau blend (shark+gull) or as bisociative blend refering to Chagall. This example shows that the bisociation can be used on the production level (e.g. creative blend but the reader cannot decompose it), on the interpretation level (e.g. even if there was no such intention when generating a name, the bisociation can be present at the reader's side) or both.

We like some names generated as lexical blends more than the others – what counts? Even if names are generated using similar principles, some of them are much more creative, achieving higher degree of creative duality, compressing multiple levels of meaning and perspective into a simple name [20]). It is the combination of simplicity and bisociation (in our case the switch from animal wor(l)d to cultural realm) that seems to be the most impressive. To verify this claim and to get a more thorough evaluation of automatically generated names, we plan to collect human subjects feedback as well as compare human-generated and automatically generated names. We will also further elaborate the automatic recognition of blend complexity and on the other side the blend generation part (e.g. including phonological criteria, rhymes, more background knowledge, etc.). Next, we will investigate the role of emotions: while some names were neutral, many had very strong emotional content (cf. negative emotions in *disgusoarse, horrabit* or the name given to the *hammerhead gull*, for which instead of naming it a user wrote "deserves death by fire, not a name") or positive emotions in *le trop joli*, name used for guinea lion. Another spectre of research is to investigate the generality of our blend categorisation by applying it to other domains.

This work has been supported by the projects ConCreTe (grant nb. 611733), WHIM (611560) and Prosecco (600653) funded by the European Commission, FP 7, the ICT theme, and the FET program. We thank also A. Fredriksen, the author of the pictures.

References

1. Bhatta, S.R., Goel, A.: Learning generic mechanisms for innovative strategies in adaptive design. The Journal of the Learning Sciences 6, 367–396 (1997)
2. Boden, M.: The Creative Mind: Myths and Mechanisms. Basic Books (1991)
3. Deri, A., Knight, K.: How to make a frenemy: Multitape FSTs for portmanteau generation. In: Proc. of the NAACL (2015)
4. Domeshek, E., Kolodner, J.: A case-based design aid for architecture. In: Gero, J., Sudweeks, F. (eds.) Artificial Intelligence in Design '92, pp. 497–516 (1992)
5. Fauconnier, G., Turner, M.: Conceptual integration networks. Cognitive Science 22(2), 133–187 (1998)
6. Goel, A.K., Craw, S.: Design, innovation and case-based reasoning. The Knowledge Engineering Review 20(3), 271–276 (2005)
7. Jakubíček, M., Kilgarriff, A., Kovář, V., Rychlỳ, P., Suchomel, V., et al.: The tenten corpus family. In: 7th Int. Corpus Linguistics Conf. pp. 125–127 (2013)
8. Kilgarriff, A., Rychly, P., Smrz, P., Tugwell, D.: The sketch engine. In: Proc. EURALEX 2004. pp. 105–116 (2004)
9. Koestler, A.: The Act of Creation. New York:Macmillan (1964)
10. Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann Publishers Inc. (1993)
11. Kuznetsova, P., Chen, J., Choi, Y.: Understanding and quantifying creativity in lexical composition. In: Proc. Conf. Empirical Methods in Natural Language (EMNLP). pp. 1246–1258 (2013)
12. Lehrer, A.: Blendalicious. Lexical creativity, texts and contexts pp. 115–133 (2007)
13. Li, B., Zook, A., Davis, N., Riedl, M.O.: Goal-driven conceptual blending: A computational approach for creativity. In: Proc. of the 3rd Int. Conf. on Computational Creativity. pp. 9–16 (2012)
14. Martins, P., Urbančič, T., Pollak, S., Lavrač, N., Cardoso, A.: The good, the bad, and the AHA! blends. In: Proc. of the 6th Int. Conf. on Computational Creativity (2015)
15. Norman, D.A.: The design of everyday things. Basic books, NY (1988)
16. Özbal, G., Strapparava, C.: A computational approach to the automation of creative naming. In: Proc. of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1. pp. 703–711 (2012)
17. Pereira, F., Cardoso, A.: The horse-bird creature generation experiment. The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour(AISBJ) 1(3), 257–280 (2003)
18. Smith, M.R., Hintze, R.S., Ventura, D.: Nehovah: A neologism creator nomen ipsum. In: Proc. of the 5th Int. Conf. on Computational Creativity. pp. 173–181 (2014)
19. Veale, T.: Tracking the lexical zeitgeist with WordNet and Wikipedia. In: Proc. European Conf. on Artificial Intelligence (ECAI). pp. 56–60 (2006)
20. Veale, T., Butnariu, C.: Harvesting and understanding on-line neologisms. Cognitive Perspectives on Word Formation pp. 399–420 (2010)
21. Wiggins, G.A.: Towards a more precise characterization of creativity in AI. In: Proc. of Workshop Program of ICCBR – Creative Systems: Approaches to Creativity in AI and Cognitive Science (2001)

Generating Plots for a Given Query Using a Case-Base of Narrative Schemas

Pablo Gervás[1], Raquel Hervás[2], and Carlos León[2]

[1] Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid
Ciudad Universitaria, 28040 Madrid, Spain
pgervas@ucm.es
[2] Facultad de Informática, Universidad Complutense de Madrid
Ciudad Universitaria, 28040 Madrid, Spain
raquelhb@fdi.ucm.es,cleon@ucm.es
WWW home page: http://nil.fdi.ucm.es/

Abstract. Computational generation of literary artifacts very often resorts to template-like schemas that can be instantiated into complex structures. With this view in mind, the present paper presents a case-based reasoning solution that builds a plot line to match a given query, expressed in terms of a sequence of abstraction of plot-bearing elements of a story, by retrieving and adapting templates for narrative schemas from a case-base. The abstractions of plot-bearing elements of a story are defined in terms of Propp's character functions. The case-base of narrative schemas is built based on a review of a number of existing attempts to provide an elementary set of patterns for basic plots. A selection of these patterns, reformulated in terms of Propp's character functions, is used as case-base. The paper explores a solution for automatic generation of stories based on this formulation of the narrative schemas.

Keywords: computational creativity, narrative, narrative schemas, transformational case adaptation, compositional case adaptation

1 Introduction

Humans that write stories reuse material from stories they know. This may include characters, settings, scenes, or lines of dialogue. Of these, the most important is the reuse of story structure. In order to capture computationally this type of reuse of experience, an abstract representation of story structure is needed. The present paper describes a case-based solution for story generation that relies on Vladimir Propp's Morphology of the Folk Tale [13]. A case-base of narrative schemas described using this representation [5] is used to provide plot lines to match a query, and the plot lines are then fleshed out into full stories by instantiating the abstract plot line with specific story actions [4].

2 Previous Work

To support the approach followed in this paper, four areas of previous work need to be considered: case-based approaches to story generation, Propp's formalism for analysing stories, the Propper system for story generation, and existing approaches to case adaptation.

2.1 Case-Based Approaches to Story Generation

Roger Schank stated that the way in which memory works is not only based on processes that manipulate mental data, but instead as continuous recalling and adapting process of previous stories that define our world [18, 17]. Turner's MINSTREL exemplified this approach by generating short stories about King Arthur and his Knights of the Round Table [19]. MINSTREL handled episodic memories in two different ways: either by instantiating a matching schema in the story from a basic query, or by performing a basic adaptation on the query, querying the episodic memory with it and returning an adaptation and modification of the query. Knowledge intensive case-based reasoning approaches [3, 10, 11] use Semantic Web technologies for knowledge representation and simple combinatorial algorithms for generating the structure of new plots by reusing fragments of structure of previous stories, inspired in the morphology of Russian folk-tales studied by Vladimir Propp [13]. Relying on more shallow representations, [14] and [16] introduce a story planning algorithm inspired by case-based reasoning that incorporates *vignettes* – pre-existing short narrative segments – into the story being generated. Other approaches to story generation based on case bases of previous schemas include efforts towards incorporating analogy-based reasoning to knowledge acquisition [9, 15]. These systems are usually focused on the retrieval, adaptation and evaluation of old schemas to new domains. In general, all these approaches rely on inter-domain analogies and generate new instances of old narrative schemas. Reuse of previous stories is also applied in [12], where case-like structures known as Story Contexts are mined from a set of previous stories and used to inform the selection of the next action to add to a story in an incremental generation process.

2.2 Proppian Morphology of a Story

At the start of the 20th century, Vladimir Propp [13] identified a set of regularities in a subset of the corpus of Russian folk tales collected by Afanasiev [1]. These regularities he formulated in terms of *character functions*, understood as acts of the character, defined from the point of view of their significance for the course of the action. According to Propp, for the given set of tales, the number of such functions was limited, the sequence of functions was always identical, and all these fairy tales could be considered instances of a single structure. The set of character functions identified by Propp includes a number of elements that account for a journey (`departure`, `return`), a number of elements that detail the involvement of the villain and the struggle between hero and villain (`villainy`, `struggle`, `victory`, `pursuit`, `rescue_from_pursuit`), a number of elements that describe the acquisition of a magical agent by the hero (`test_by_donor`, `hero_reaction`, `acquisition_magical_agent`).[3]

2.3 The Propper System

The Propper system developed by Gervás [4] constitutes a computational implementation of a story generator initially based on Propp's description of how his morphology might be used to generate stories.

It relies on the following specific representations for the concepts involved:

- a *character function*, a label for a particular type of acts involving certain named roles for the characters in the story, defined from the point of view of their significance for the course of the action
- a sequence of character functions chosen as backbone for a given story
- possible instantiations of a character function in terms of specific *story actions*, involving a number of *predicates* describing events with the use of *variables* that represent the set of characters involved in the action

Based on these representations the Propper system defines a procedure that first chooses a sequence of character functions to act as abstract narrative structure to drive the process, and then progressively selects instantiations of these character functions in terms of story actions to produce a conceptual representation – in terms of an ordered sequence of predicates – of a valid story. This conceptual representation is a *fabula*, a sequence of states that contain a chain of story actions – which are instances of those character functions. A story action involves a set of preconditions – predicates that must be present in the context for continuity to exist –, and a set of postconditions – predicates that will be used to extend the context if the action is added to it. Each story action is linked to its context of occurrence by having its preconditions satisfied by the preceding state.

2.4 Case Adaptation

Probably one of the most difficult processes in the CBR cycle is the reuse or adaptation stage. After retrieving the most similar case (or cases) from the case base, the solution from the retrieved case must be used to create a new solution for the problem at hand.

Wilke and Bergman [20] established a classification of CBR adaptation into three different methods: null adaptation, transformational adaptation and generative adaptation. The simplest kind of adaptation is *null adaptation*, where the solution of the retrieved case is used without any modification. As simple as this adaptation method is, it can obtain very good results for simple problems. *Transformational adaptation* consists on the transformation of the solution of

[3] For reasons of space, only a number of character functions relevant to the examples given in the paper are described. Readers can check the referenced sources for more detail.

the retrieved case into the solution required for the query. In order to do that, the retrieved solution may be reorganized and modified by deleting or adding new elements. Finally, *generative adaptation* consists on generating the new solution from scratch, but reusing the process used to obtain the solution from the retrieved case.

These three adaptation methods are formalized by considering that only one case is retrieved and adapted. However, some problems may be better solved by reusing information from more than one case. This is what Wilke and Bergman called *compositional adaptation*, where the new solution is obtained by adapting the solutions of multiple cases. This multiple case adaptation can be done using transformational or generative methods, but the main idea is that the solution for the case at hand can be better obtained by taking into account more than one case from the case base.

There are many examples of compositional adaption in recent CBR works. Arshadi and Badie [2] apply this adaptation in a tutoring library system. In this kind of application it is probable that many cases can be similar to the user request at the same time, so it is important to take all of them into account when generating the solution for a given query. Hervás and Gervás [6] also use multiple cases for text generation based on templates. When the information that must appear in a sentence is not covered by the template of the retrieved case, a new retrieval process is triggered in order to find more cases which templates can cover the information in the query. Ontañón and Plaza [8] present the concept of amalgam as a formal operation over terms in a generalization space. Although amalgams are not proposed as an adaptation method by themselves, the notion of amalgam is related to merging operations that can be used in compositional adaptation to combine two or more cases. Müller and Bergmann [7] use a compositional adaptation approach for cooking recipes represented as cooking workflows. During the adaptation stage, missing parts of retrieved cooking workflows are covered using information from other cases.

3 Case-Based Construction of Plot Lines for Stories

The present paper describes a case-based approach to the construction of plot lines for stories – described as sequences of character functions – which can then be fleshed out into stories.

3.1 Case-Based Construction of Plot Lines

The system operates from a query provided by the user. This query is expressed as a sequence of character functions that the user would like to see included in the desired plot line.

The system compares the given query with the set of plot lines represented in its case base.

The Case-Base The case base of schemas used for this paper is built from the narrative schemas reviewed in [5]. These correspond to a set of sequences of character functions – in Propp's sense of plot relevant abstractions of the activity of characters – that correspond to a number of theoretical characterizations of possible plots for stories, also referred as *narrative schemas.* The case-based reasoning approach will therefore operate over sequences of character functions, and it will return a sequence of character functions that best matches the given query.

Merging Plot Lines When dealing with plot lines in terms of sequences of character functions it is often necessary to merge two plot lines to obtain a third plot line. Because plot lines are sequentially ordered, and specific elements in the plot may have dependencies with other elements, the relative order in which they appear in the sequence is very relevant. For the purposes of the present paper, this is done as follows:

- the query is traversed sequentially
- each character function in the query is checked against the next character function in the case
- if they match the character function is added to a *matching* subsequence
- if they do not, the character function from the query is added to a *wanted* subsequence, and the next character function from the query is checked against the character function in the case
- if the end is reached for the query the rest of the case is added as an *added* subsequence
- if the end is reached for the case the rest of the query is added as a *wanted* subsequence

The merge is constructed by concatenating into a single sequence the subsequences of character functions that are generated during the merge in this fashion. This has the advantage of interleaving the character functions from the original query with the contributions from the various cases involved while always respecting the order in which these character functions appeared in the query.

Similarity We consider a similarity function for plot lines based on identifying the relative mutual coverage between query and case. The set of subsequences of the query that appear as subsequences of the case in the corresponding order is referred to as the *match.* The *remainder* is the set of subsequences of the query that are not covered by the case. The *addition* is the set of subsequences of the case that did not appear in the query.

The similarity employed in the current version of the system is calculated as an average between the percentage of the query covered by the case – the ratio between the size of the match and the size of the query – and the percentage of the case that is involved in the match – the ratio between the size of the match and the size of the case. This is intended to capture the suitability of the case both in terms of maximum coverage of the query and in terms of minimum addition of character functions beyond the query.

To compute these values the query is merged with the case as described above. The match is then reckoned to be the set of *matching* subsequences. The remainder is then reckoned to be the set of *wanted* subsequences. The addition is then reckoned to be the set of *added* subsequences.

Retrieval and Adaptation If there is a case whose plot line matches the query, that case is returned as solution.

Otherwise, the cases are ranked based on their similarity with the query. The set of character functions that appears in the overall set of subsequences resulting from this process constitutes a possible solution to the problem posed by the query, as it would constitute a combination of the query and the case.

If the retrieved case does not cover all the character functions in the query, further retrieval processes will be required. This corresponds to solving the given query with a complex story that combines more than one plot line. To achieve this, an additional retrieval process is set in motion using the remainder of the first retrieval process as a query to the second one.

For each additional case retrieved, the resulting solution is merged with the result of prior stages using the same procedure as for merging a query and a case. These ensures that relative order of appearance of related character functions within each narrative substructure that has been reused is respected in the final solution.

The retrieval and adaptation process can be iterated until the remainder of the query is empty. The merge obtained at this point is the final solution. This sequence of character functions is the solution found by the system as plot outline for a story to match the given query.

An Example of Plot Line Construction For a query `villainy departure villain_punished return`, the most similar case retrieved is:[4]

`villainy` *`hero_pursued rescue_from_pursuit struggle victory`* `villain_punished`

The merge of the query and case, with the different subsequences marked[5] is:

`villainy` DEPARTURE *`hero_pursued rescue_from_pursuit struggle victory`* `villain_punished` RETURN

Within the resulting merge, the elements not matched by the retrieved case (the remainder: `departure return`) appear in the same relative position with respect to the other elements of the query as they did in the original sequence of the query.

---

[4] Elements in the case that match the query are shown in plain text, and elements that do not are shown in italic.

[5] `Matched` elements are shown in plain text, WANTED elements in small caps, and *`added`* elements in italic.

To cover this remainder, a second case-based reasoning process is set in motion, with the remainder as a query. For this second process, the query would then be DEPARTURE RETURN. The most similar case retrieved is:[6]

`departure` *`difficult_task task_resolved`* `hero_pursued rescue_from_pursuit struggle victory` *`test_by_donor hero_reaction acquisition_magical_agent`* `return`

The merge of this additional case with the result of the prior CBR process, with the different subsequences marked as above is

`villainy departure` *`difficult_task task_resolved`* `hero_pursued` *`rescue_from_pursuit struggle victory`* `villain_punished` *`test_by_donor hero_reaction acquisition_magical_agent`* `return`

This implies that the remainder is now empty.

3.2 Fleshing out the Plot Line for the Story

Because character functions are abstractions of plot relevant activities by the characters, the draft plot line obtained as a result of the retrieval and adaptation stage needs to be fleshed out before it can be considered a story.

This involves instantiating the character functions with specific story actions. This can be done following the original procedure for the Propper system [4] for obtaining a fabula from the sequence of character functions corresponding to the resulting plot line. This relies on definitions of the story actions defined in terms of predicates that define an action, with identifiers for the characters as arguments. The definitions of these story actions also contain predicates that define preconditions and effects of the action in question. The instantiation procedure relies on unification of each new story action with the previous context to guarantee continuity and coherence in terms of which characters perform which actions.

Table 1 presents an example of story corresponding to the plot line obtained as a result of the case-based reasoning procedure described in section 3.1.

It is worth noting that although the character functions being instantiated arise from two different original plot lines as provided by the cases, the fleshing out procedure instantiates them with story actions that link up to conform a single coherent story about a hero (character id147) and a villain (character id755). An initial villainy (state 0) forces the hero to set out (state 1), he faces a difficult task (states 2-3), he undergoes several conflicts with the villain (states 4-5 and 6-7). The end of this particular story involves a meeting with a donor that provides a magical agent (states 9-11) and an eventual return of the hero (state 12).

4 Discussion

The approach followed for case adaptation in the described procedure is transformational and compositional. Both the transformation of the retrieved cases to

[6] The use of italics shows the match of the retrieved case with the sequence resulting from the earlier CBR process.

State	Event description
0	character id755 kidnap id755 id756 victim id756 character id756 misbehaved id755
1	seeker id147 character id147 sets_out id147
2	sets id181 id147 id183 character id181 id183 difficult_task id183 involves id183 manufacture
3	character id181 solve id147 id183 before dead_line
4	runs_away id147 pursues id755 id147 hides_in id147 tree tries_to_destroy id755 tree
5	jumps_to_another tree escapes id147
6	weight_contest id147 id755 confrontation id147 id755
7	makes id147 protective_gesture banishes id755
8	pardoned id755
9	shows id388 id389 donor id388 character id388 magical_agent id389 offers_exchange id388 id389 id147 test id388 id147
10	agrees_to_exchange id147 uses id147 id389 id388 deceives id147 id388 positive_result id147
11	helper id53 character id53 meets id755 id53 offers_services id53 id755
12	returns id147

Table 1. An example story corresponding to the plot line shown earlier.

better match the query and the composition of more than one case are covered by the described procedure for merging two sequences of character functions while respecting the relative order of appearance of their elements.

The procedure followed for story construction operates at a higher level of abstraction than [19, 14, 16], and with greater flexibility than [3, 10, 11] – who also use character functions – due to its highly compositional approach to case recombination.

The case-based reasoning procedure described relies on cases to provide a complete backbone for a plot line, reusing the structure of a given plot completely, with no option for leaving out certain parts of it. The procedure for successive retrievals, together with a merging approach that respects the relative order in which character functions occur in the query and interleaves the additions without repetition, allow for more than one such plot backbone to be combined into more complex stories. However, this approach will only succeed as long as there exists some case in the case base with a reasonably similar sequence of character functions. Beyond this, it might be necessary to consider alternative approaches that allow reuse of fragments of cases, to be recombined into longer sequences.

The choice of case base employed here is built from schemas that are intended as complete plots. Alternative formulations of the case base are possible, built from smaller units of plot, such as scenes. These might be represented as subsequences of character functions that occur frequently in different plot lines. A solution along these lines might define the case base in terms of smaller units that would be abstracted during the construction of the case base. This procedure is similar to the one employed in [12], where cases are retrieved to generate the actions of the story one by one (one case per action). An alternative procedure would be to operate over a case base of complete plots but define a different retrieval algorithm that allows a certain fragmentation of these plots during retrieval.

Two important aspects to consider in creative plot generators are coherence and novelty. By virtue of its process of reusing large segments of existing plots, the described procedure is likely to generate coherent plots, though how coherence is affected by the merging procedure should be addressed in further work. In that sense, the process of instantiation with story actions employed by the Propper system presents an advantage in that it checks the satisfaction of preconditions of each action in its context during construction. With respect to novelty, processes that reuse existing solutions are exposed to the risk of reproducing aspects of prior material. To address this risk, future work should consider establishing limits on the extent of reuse considered. These could take the form of avoiding cases that are perfect matches for a given query, and preferring solutions obtained by combination of more than one case.

5 Conclusions

The case-based reasoning solution described in this paper operates at a sufficiently high level of abstraction to allow the construction of valid plot lines by combination of cases that represent narrative schemas which are merged into a plot line that matches a given query, and which can then be instantiated into a specific coherent story.

Acknowledgements

This paper has been partially supported by the project WHIM 611560 funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program.

References

1. Alexander Nikolayevich Afanasyev. *Narodnye russkie skazki A. N. Afanaseva [Folk Russian tales of A. N. Afanasev]*, volume 1-3. Moscow, 1855.
2. Niloofar Arshadi and Kambiz Badie. A compositional approach to solution adaptation in case-based reasoning and its application to tutoring library. In *Proceedings of the 8th German Workshop on Case-Based Reasoning, Lammerbuckel*, 2000.
3. P. Gervás, B. Díaz-Agudo, F. Peinado, and R. Hervás. Story Plot Generation Based on CBR. *Knowledge-Based Systems. Special Issue: AI-2004*, 18:235–242, 2005.
4. Pablo Gervás. Computational Drafting of Plot Structures for Russian Folk Tales. *Cognitive Computation*, 07/2015 2015.
5. Pablo Gervás, Carlos León, and Gonzalo Méndez. Schemas for narrative generation mined from existing descriptions of plot. In *Computational Models of Narrative*, Atlanta, Georgia, USA, 05/2015 2015. Scholoss Dagstuhl OpenAccess Series in Informatics (OASIcs), Scholoss Dagstuhl OpenAccess Series in Informatics (OASIcs).

6. Raquel Hervás and Pablo Gervás. Case-based reasoning for knowledge-intensive template selection during text generation. In Thomas R. Roth-Berghofer, Mehmet H. Göker, and H.Altay Güvenir, editors, *Advances in Case-Based Reasoning*, volume 4106 of *Lecture Notes in Computer Science*, pages 151–165. Springer Berlin Heidelberg, 2006.
7. Gilbert Müller and Ralph Bergmann. Compositional Adaptation of Cooking Recipes using Workflow Streams. In *Computer Cooking Contest, Workshop Proceedings ICCBR 2014*, Springer, 2014. The original publication is available at www.springerlink.com.
8. Santiago Ontañón and Enric Plaza. Amalgams: A formal approach for combining multiple case solutions. In Isabelle Bichindaritz and Stefania Montani, editors, *Case-Based Reasoning. Research and Development*, volume 6176 of *Lecture Notes in Computer Science*, pages 257–271. Springer Berlin Heidelberg, 2010.
9. Santiago Ontañón and Jichen Zhu. On the role of domain knowledge in analogy-based story generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1717–1722. AAAI Press, 2011.
10. F. Peinado and P. Gervás. Evaluation of automatic generation of basic stories. *New Generation Computing*, 24(3):289–302, 2006.
11. Federico Peinado. *Un Armazón para el Desarrollo de Aplicaciones de Narración Automática basado en Componentes Ontológicos Reutilizables.* PhD thesis, Universidad Complutense de Madrid, Madrid, 2008.
12. Rafael Pérez y Pérez and Mike Sharples. MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence*, 13(2):119–139, 2001.
13. Vladimir Propp. *Morphology of the Folk Tale.* Akademija, Leningrad, 1928.
14. M. Riedl and Carlos León. Toward vignette-based story generation for drama management systems. In *Workshop on Integrating Technologies for Interactive Stories - 2nd International Conference on Intelligent Technologies for Interactive Entertainment*, 2008.
15. Mark Riedl and Carlos León. Generating story analogues. In *AIIDE*, 2009.
16. Mark O. Riedl and Neha Sugandh. Story planning with vignettes: Toward overcoming the content production bottleneck. In *Interactive Storytelling*, volume 5334 of *Lecture Notes in Computer Science*, pages 168–179. Springer, 2008.
17. R. Schank. *Dynamic Memory : A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, 1982.
18. R. Schank and R. Abelson. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures.* L. Erlbaum, Hillsdale, NJ, 1977.
19. Scott Turner. *MINSTREL: A Computer Model of Creativity and Storytelling.* PhD thesis, University of California at Los Angeles, Los Angeles, CA, USA, 1992.
20. Wolfgang Wilke and Ralph Bergmann. Techniques and knowledge used for adaptation during case-based problem solving. In *Proceedings of the 11th International Conference on Industrial and Engineering Applications of Artificial In Telligence and Expert Systems: Tasks and Methods in Applied Artificial Intelligence*, IEA/AIE '98, pages 497–506, London, UK, UK, 1998. Springer-Verlag.

Seeking Divisions of Domains on Semantic Networks by Evolutionary Bridging

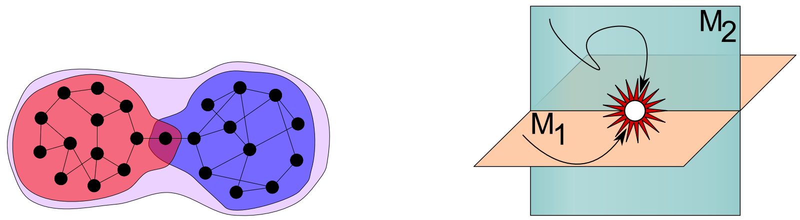
João Gonçalves, Pedro Martins, António Cruz, and Amílcar Cardoso

CISUC, Department of Informatics Engineering, University of Coimbra
{jcgonc,pjmm}@dei.uc.pt
antonioc@student.dei.uc.pt
amilcar@dei.uc.pt

Abstract. Computational Creativity systems based on Conceptual Blending (CB) and Bisociation theories operate on input knowledge to reveal seemingly unrelated information. The input spaces or domains can be of various sources and contain vast amounts of knowledge. It is central a process that selects useful building blocks of semantic data that does not narrow the search space of the creative algorithm. It is also vital that the data selection process is of high performance in order to handle a large knowledge base in a useful time. With those objectives in mind, we propose an evolutionary high performance algorithm that extracts two semantic sub-graphs from a knowledge base to be used as building blocks in computational blending processes.

1 Introduction

A creative process can be seen as a form of heuristic search for a construct on a vast semantic space of concepts and domains. In this paper we propose an evolutionary approach inspired by the work of Nagel [6] for selecting two domains from a broader knowledge structure using high performance algorithms. This allows a faster extraction of a more concise representation of the data to be used in computational concept generation techniques, such as CB and Bisociative Knowledge Discovery, among other applications. As the amount of available knowledge dramatically expands each year, high performance algorithms are required to cope with the extraction of new insights, together with growth of multidisciplinary knowledge bases. In [3] Juršič, based on the ABC model by Swanson [11] [10], proposes the CrossBee system for supporting creativity insight in knowledge discovery of literature. In the ABC model, Swanson remarks that reference citations and other bibliographic indications potentially reveal new knowledge, which is not clearly intended neither logically exposed in the literature. That is, ABC exposes the $A \implies B \implies C$ logical consequence, being B the term which relates the remaining terms A and C. Using this idea, CrossBee tries to explore bridging terms linking two apparently disconnected literature domains. In CrossBee, the bridge terms contain relations between two terms, each from a different domain, that were mined from a literature knowledge base. The terms are extracted from various sections in the literature texts, such as bibliography, citations, logical consequences and other references present in the texts. Having the literature containing the terms from A referencing terms regarding B, and simultaneously the literature C also references terms from B, then a unintended modus ponens inference suggest a hidden relation between A and C. Hence, the system following closely the ABC idea by Swanson.



(a) (b)

Fig. 1: Two domains connected by a single concept (left: adapted from [1]) and the juxtaposition of two frames of reference in Koestler's Bisociation (right: adapted from [4]).

A similar work by Nagel et al. [6] introduces a formalised spreading activation algorithm to identify bridging concepts in a semantic graph (Fig. 1a). The bridging terms interconnect nodes from disjoint semantic domains, following an identical idea to CrossBee. However, the main intention in this case is to juxtapose two apparently unrelated domains through a single term [5]. This notion of pairing two disjoint frames of reference using a singular connection was put forth by Koestler and named Bisociation [4] in his work, *The Act of Creation* [4]. There, Arthur Koestler attempts to describe creative behaviour present in humour, arts and science. This model consists on "the perceiving of a situation or idea ... in two self-consistent but habitually incompatible frames of reference (M_1 and M_2)" as shown in Fig. 1b. For instance, in humour, bisociations could relate the unforeseen transformation from one meaning to another [7]. In their paper, Nagel explores a search space, defined using a bisociation score, which rates individually each bridging node subdividing the semantic graph in two completely disjoint sets. However, their approach does not allow a tolerance of the intersection between the two domains. Thus, it is in a sense a hard margin solution and in our opinion, it terminates the search prematurely in real world problems, as underlined in their conclusion. However, their highly formalised work served as a basis for our present approach. In the following section, we offer our evolutionary approach in the form of a Genetic Algorithm (GA), inspired by the work of Nagel.

2 Algorithm

The purpose of the algorithm is to identify two partially overlapping sub-graphs S_0 and S_1 of a larger semantic graph S. Given their structure, interrelations and arrangement within the larger graph, we believe the sub-graphs could be seen as domains of knowledge in the broader semantic graph.

The cardinality of each graph structure is identified by the symbol $\#$. Thus, $\#S$ is the number of nodes existing in the graph S and we denote this quantity as the size of the graph. Each sub-graph represents a network of highly interconnected nodes, which if belonging to a semantic graph, could represent a domain of related concepts [1]. Both sub-graphs share at least a single node N_b, the bridge node, and the sub-graphs should be balanced [6] regarding a split through the bridge node. The size of both sub-graphs $\#S_0$ and $\#S_1$ should be maximized, with only the condition of $S_0 \cap S_1 = \{N_b\}$. Then, a unique path will flow from one sub-graph to the other through the bridge node which has the unique index $b \in \{1 \dots \#S\}$. When this happens, the unique bridge node may represent a possible bisociation which juxtaposes one domain (sub-graph) into the other (Fig. 1b).

A degree d_i of a node N_i with $i \in \{1 \dots \#S\}$ represents the number of incident edges to that given node. Nodes with $d = 2$ represent a single relationship between two concepts, being these the most likely candidates for a bridge node [6]. The reasoning behind this choice is the interest in mapping two domains over a single and clear semantic relationship, through the bridge node. In this case, any concept from one sub-graph can be projected onto any other concept from the other sub-graph, offering a foundation for further transformations of concepts using processes from bisociation and CB [7].

Otherwise nodes with $d \geq 3$ map a more vague set of relations between connected concepts. Intuitively, the view of an idea in two distinctly but opposing views is more fine tuned to two set of concepts (two domains) connected by a single node [6]. A simple example which demonstrates this idea is seen in Fig. 1a. On the other hand, highly interconnected nodes express a deeply related network of information or domain. Using the above criteria, the discrete function which rates the optimality (fitness) of the bridge node N_b is defined in (1):

$$f(S_0, S_1, d_b) = \begin{cases} \frac{1}{\alpha^{\frac{|\#S_0 - \#S_1|}{\#S_0 + \#S_1}} + 1} \cdot \log(\#S_0 + \#S_1) \cdot 2^{-\beta(d_b - 2)}, & \text{if } d_b \geq 2 \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The fitness function receives as arguments the sub-graphs S_0, S_1 and the degree of the bridge node N_b as the variable d_b. The parameter α controls how similar in size the sub-graphs S_0 and S_1 are required to be, with increasing α exhibiting greater size similarity. The parameter β is used to control the penalisation given to bridge nodes with a degree $d > 2$, with the penalisation exponentially proportional to the value of β. If the degree of node being rated is 1, that is, a terminal node with a single relation, then f is set to 0 in order to prevent the GA to select terminal nodes as bridge.

Globally, a GA evolves a population of chromosomes where each chromosome represents a bridge node and two sub-graphs of the initial semantic network. Each time a new individual is created with a given bridge node, a breadth first search is executed starting in the latter node and into neighbouring nodes. The dual diffusion process (a sort of spreading activation) progresses radially until a given expansion depth is reached when both sub-graphs intersect, or all the nodes in the network have been explored. From this expansion, we calculate a performance score representing this graph division which will represent the chromosome in the global fitness landscape. Thus, the GA evolves subdivisions of the semantic network aiming to find the bridging node that maximises the size of the sub-graphs found.

A GA was chosen in order to relax the search for global optima. Given a specific graph, the search space can be of very high complexity which is likely when handling large semantic networks. The optimisation task is represented by the fitness function and the genetic operators embody the transformations that move the bridge node through the semantic graph. By using a GA, it is easy to parallelise the search algorithm with the following steps:

– fitness functions are evaluated in parallel threads;
– new individuals which will define the next generation are created in parallel threads;
– the generation of random chromosomes for the initial population and when the GA stagnates is executed in parallel threads.

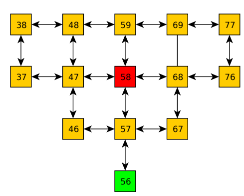


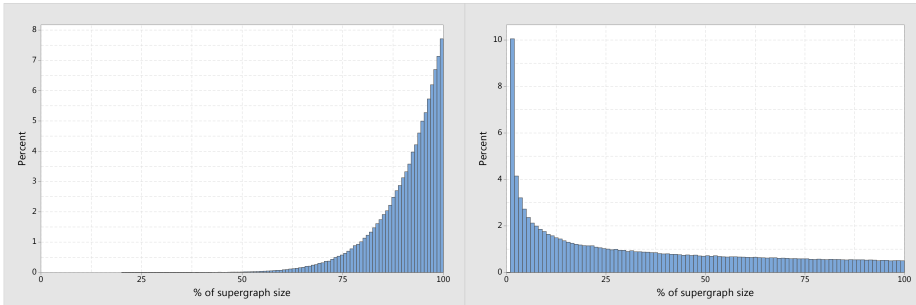
Fig. 2: A combination of random translation movements (a random walk) starting at the red node and given time, eventually leads to the green node. Best viewed in colour.

In our GA we use entirely the mutation operator. Using Fig. 2 as an example, it can be seen that the fundamental operation required to traverse the search space (the graph) is the translation of the bridge node. Thus, the search algorithm samples stochastically (walks randomly) the locations in the graph that maximise the fitness function. The mutation changes the location of the bridge node using the connected nodes (neighbourhood) recursively. In order to allow the bridge node to jump to distant sections of the graph, we use a quadratic random number generator. The probability function controlling the locality behaviour of this process decreases with the increasing number of jumps the bridge node is allowed to make. This number is given in (2).

$$j(r, s) = r^{\gamma} s + 1 \tag{2}$$

The parameter $r \in [0, 1[$ represents a previous output of an uniform pseudo random number generator. The number of nodes $\#S$ in the graph S is represented by the parameter s. The parameter $\gamma \in]0, +\infty[$ controls the average jump distance (translation deepness) the mutation applies to the bridge node.

If $\gamma >> 1$, the mutation tends to move the bridge node towards nearby nodes (Fig. 3b). When $\gamma \to +\infty$, the translation tends to move each bridge node to



(a) $\gamma = 0.1$

(b) $\gamma = 2$

Fig. 3: Probability Density Functions for two values of γ when using the bridge jump mutation function (2).

one of the connected neighbours. For $\gamma \in]0, 1[$, particularly nearby 0, tends to translate the bridge node to distantly connected nodes (Fig. 3a) and, in a sense, promote a random search of the fitness landscape. The case where $\gamma = 1$ forces a straightforward stochastic search whereby each bridge node can be moved to any other node position of the domain S with constant probability.

After the mutation has been applied to the chromosome, the expansion of sub-graphs from the full domain is executed again from the newly calculated bridge node. When this completes, two new sub-graphs divide part (or all) of the domain with roots at the bridge node and depth $d_{(0\ldots1)}$. This expansion is a type of breadth first search starting at the node N_b so that the first nodes to explore are the nearby nodes. The main idea behind this reasoning is shown in Algorithm 1. Using two pairs of open (to expand) and closed (already expanded) nodes, the algorithm inserts the visited nodes in the two sub-graphs which will represent the sub-domains S_0 and S_1. An example of the process is seen in (Fig. 5) from where two different coloured sub-graphs emerge (Fig. 5). The function $\texttt{nodes}(S_i)$ returns the set of nodes $\{N_i\}, i \in \{0, \ldots \#S_i - 1\}$ in the sub-graph $\{N_k\}, k \in \{0, 1\}$. The variables O_i and C_i represent respectively the set of open (to visit) and closed (already visited) nodes related to the sub-graph i. The function $\texttt{split}(N_b, S)$ divides the neighbourhood of the bridge node N_b in two sets of nodes as evenly as possible. When the neighbourhood of N_b is odd, a randomly chosen set S_i receives the additional node so that in the worst case, the difference of cardinality between S_0 and S_1 is 1.

The function $\texttt{expandOneLevel}(S_i, O_i, C_i, S)$ cycles through all the nodes in the set O_i, inserts each visited node in the set C_i and in the sub-graph S_i, including the connected edges. Then it extracts the neighbourhood of each visited node and inserts the neighbour nodes in the set O_i, so that in the next iteration of the function $\texttt{createSubgraphs}()$ the algorithm expands from the current neighbourhood of O_i. Thus, the function `expandOneLevel()` executes an equivalent single iteration of a breadth first search at the same deepness level. All the nodes and edges are obtained from the graph S.

Algorithm 1 Function createSubgraphs()

function $\textsc{createSubgraphs}(N_b, S)$
  $\{S_0, S_1\} \leftarrow \texttt{split}(N_b, S)$
  $O_0 \leftarrow \texttt{nodes}(S_0)$
  $O_1 \leftarrow \texttt{nodes}(S_1)$
  $C_0 \leftarrow N_b$
  $C_1 \leftarrow N_b$
  $I \leftarrow \emptyset$
  repeat
    $\texttt{expandOneLevel}(S_0, O_0, C_0, S)$
    $\texttt{expandOneLevel}(S_1, O_1, C_1, S)$
    $I \leftarrow S_0 \cap S_1$
  until $\frac{\#I}{\#S_0 + \#S_1} \geq \tau \vee \#S_0 = 0 \vee \#S_1 = 0$
end function

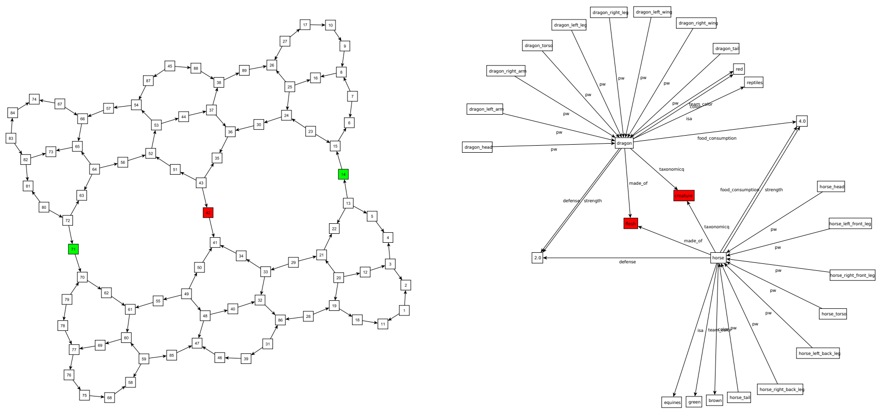
Starting at the bridge node N_b the expansion grows radially throughout the connected nodes, creating the sub-graphs while visiting the explored nodes until the sub-graphs intersect. For a graph in structure similar to Fig. 1a, the sub-graphs are expected to intersect only in the bridge node. However, in real cases, while the expansion is taking place, the intersection can suddenly show a small amount of nodes when in comparison with the size of both sub-graphs S_0 and S_1. When this happens, the algorithm may not be able to find a clean (and useful) division of the graph S.

Using a similar idea to Soft Margin in [2], we include the parameter $\tau \in \mathbb{R}_0^+$ to allow more than one bridge node connecting the sub-graphs S_0 and S_1. With $\tau = 0$, the intersection I between the sub-graphs is allowed to contain only the bridge node, as in Nagel. When the ratio of the intersection to both sub-graphs size increases above τ, the algorithm stops and returns the most recently created sub-graphs starting at node N_b. In sum, τ represents the trade-off between the penalization of highly interconnected sub-graphs and the maximisation of the size of those sub-graphs.

Consider the following example: after a 5 level expansion, the intersection of the two sub-graphs with a size of 2000 nodes each, suddenly increases from 1 (the bridge node only) to 100. This means that the fifth iteration raised the sub-graphs size to intersection ratio from $\frac{1}{2000+2000} = 0.025\%$ to $\frac{100}{2000+2000} = 2.5\%$, a $100\times$ fold increase. It may happen that the sub-graphs contain useful knowledge and for this reason, they should not be discriminated. Depending on the situation at hand, the parameter τ chosen to control the ratio may or not be significant.

3 Results and discussion

The feasibility of our algorithm was tested using three semantic graphs. The first, shown in Fig. 4a, was generated exclusively to test the theory supporting the algorithm. It contains 89 nodes and 106 unlabelled directed edges. The second is from the Horse-Dragon experiment, a well known semantic graph in Conceptual



(a) 89 node graph. (b) Horse-Dragon graph.

Fig. 4: Structure of the 89 node and Horse-Dragon graphs. The highest rated nodes are shown coloured. Best viewed in colour.

Blending, supplied by the authors of [8]. This semantic graph contains 32 relations between 32 attributes of the animals horse and dragon, such as physical parts, health resistance and some taxonomic properties (Fig. 4b). The last is the Perception semantic graph from [9]. The author of Perception defines his knowledge base as a summary of manually annotated common sense concepts and their relations. It contains 3892 nodes and 345463 edges. Unless otherwise stated, the parameters used for the graph division algorithm were $\tau = 0$, $\alpha = 4$, $\beta = 4$ and $\gamma = 2$. The GA evolved a population of 10^3 chromosomes with a mutation rate of 100%, no crossover and a maximum number of 10^3 evolved generations.

Before the experiments, we validated the algorithm with a 111 node graph (Fig. 5) containing 188 directed edges. After the conclusion of the GA, the two sub-graphs S_0 (green) and S_1 (cyan) are juxtaposed through the bridge node with the label 56. The GA stopped when the intersection between the sub-graphs included the nodes labelled 17, 95 and the bridge node with label 56. Afterwards, we proceeded with the experiments on the three semantic graphs.

Table 1: Fitness scores f for the four highest rated bridge nodes of the *Horse-Dragon* semantic graph.

f	degree(N_b)	label(N_b)	$\#S_0$	$\#S_1$
3.989	2	creature	15	15
3.989	2	flesh	15	15
0.249	3	4	15	15
0.249	3	2	15	15

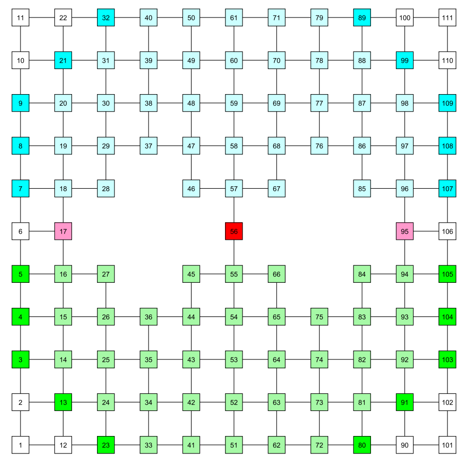


Fig. 5: Optimal sub-graph configuration of the 111 node test graph showing bridge node (red), intersection nodes (pink) and the two created sub-domains (green and cyan) each with a depth of 8 nodes starting at the bridge node. Calculated with intersection tolerance $\tau = 0$. Best viewed in colour.

Our algorithm reported a high amount of possible bridge nodes in the 89 node graph (Fig. 4a). From those, the 3 highest scored nodes are shown coloured, where the node in red scored 50% higher than the green nodes.

The Horse-Dragon [8] semantic graph is shown in Fig. 4b with the four highest rated chromosomes presented in Table 1. The majority of the nodes are terminal ($d = 1$) where a small number of highly interconnected nodes ($d \geq 2$) are clearly visible (Fig. 4b) labelled as *horse* and *dragon*. The best chromosomes generated by the GA produced were the two pairs of sub-graphs with each pair linked by the nodes with label *creature* and *flesh*.

In order to study our algorithm with a more complex and practical problem, we researched the Perception [9] knowledge base with two experiments. For the first, we did a study regarding the effect of τ in the size of the two sub-graphs. As shown in Table 2, the parameter τ highly influences the size of both sub-graphs. Having the Perception graph 3892 nodes, for certain τ values, one or both of the sub-graphs contain more than half the nodes from Perception. Therefore, a compromise has to be made so that both sub-graphs do not drastically intersect between themselves. However, both should contain a minimum amount of knowledge and relations to be useful for CB and Bisociative Knowledge Discovery. From Table 2, an interesting improvement in the size of the sub-graphs happens when τ changes from 0.05 to 0.1. With $\tau \geq 0.5$ the fitness function f does not increase, implying that the limit of the graph has been reached as the size of both sub-graphs are equal and maximum.

For the second experiment, we set $\tau = 0.1$ in order to limit the intersection between the two sub-graphs to 10% of their combined size. A list of results is present in Table 3, with all the bridge nodes having degree of 2. The fitness

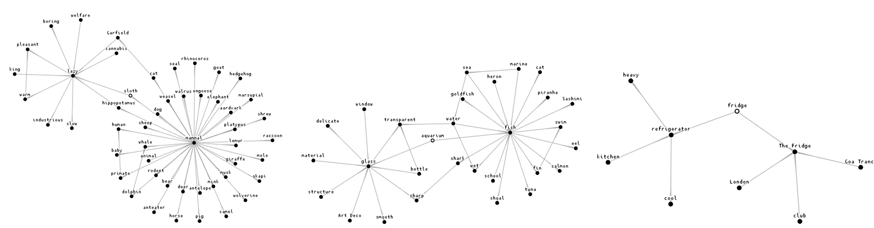
Table 2: Fitness scores f of the highest rated bridge nodes for the Perception semantic graph when varying τ.

τ	f	degree(N_b)	label(N_b)	$\#S_0$	$\#S_1$	$\#(S_0 \cap S_1)$
0.00	4.35	2	panther	47	46	0
0.05	7.07	2	Jerry Springer	586	586	7
0.10	7.70	2	Times	1919	1991	98
0.15	8.19	2	Jesus Christ	2173	2199	119
0.20	8.27	2	Pulp Fiction	2121	2132	186
0.25	8.30	2	wrestling	2264	2281	291
0.33	8.63	2	ashes	3039	3054	943
0.50	8.95	2	emerald	3868	3868	3868
0.75	8.95	2	chromosome	3868	3868	3868

Table 3: Fitness score f for 22 bridge nodes, from the Perception semantic graph with $\tau = 0.1$.

f	degree(N_b)	label(N_b)	$\#S_0$	$\#S_1$	$\#(S_0 \cap S_1)$
7.70	2	Times	1919	1991	98
7.52	2	Athens	1474	1522	62
7.07	2	Jerry Springer	586	586	7
7.03	2	sloth	1242	1177	32
6.98	2	herd	714	729	4
6.96	2	fox	984	1031	27
6.78	2	pilot	529	522	11
5.79	2	aquarium	949	820	13
4.99	2	fridge	427	514	1

declines with the increasing unbalancing between the sub-graphs S_0 and S_1. In Figs (6) we show some of the structures of the nearby connections. It is interesting to observe the relations between connected domains for the "sloth", "aquarium" and "fridge" nodes. We find the last case peculiar, as we did not knew of a heavy and cool (or cold?) trance band. For the remaining bridge nodes, we leave their insight to the reader's judgement.



(a) "sloth" bridge node. (b) "aquarium" bridge node. (c) "fridge" bridge node.

Fig. 6: Three examples of bridging nodes and their neighborhood present in Table 3.

4 Conclusions

In this work we proposed an evolutionary approach to support computational concept generation systems and knowledge discovery. Building on the work of Nagel [6], our work allows the discovery of knowledge divisions in large semantic graphs and the identification of possible key concepts which interconnect the sub-graphs. The algorithm supports various parameters to fine tune this division process in accordance with real world knowledge bases, so that different relations between knowledge domains can be researched and hopefully, give possibility to new insights between those domains. Lastly, by using a high performance algorithm, the exploratory process can be done in useful time. In the future we expect to improve our approach by experimenting with graph similarity. It would also be interesting to use a form of feedback loop by integrating a concept generating algorithm. This way, the system would direct its search towards bridging nodes and sub-graphs that would be more useful to the task that follows the GA.

Acknowledgements. The authors acknowledge the financial support from the iCIS (Intelligent Computing in the Internet of Services) project (CENTRO-07-0224-FEDER-002003) and from the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the ConCreTe FET-Open project (grant number 611733) and the PROSECCO FET-Proactive project (grant number 600653).

References

1. Michael R. Berthold, editor. *Bisociative Knowledge Discovery*. Volume 7250 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012.
2. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
3. Matjaž Juršič, Bojan Cestnik, Tanja Urbančič, and Nada Lavrač. Cross-domain literature mining: Finding bridging concepts with crossbee. In *Proceedings of the 3rd International Conference on Computational Creativity*, pages 33–40, 2012.
4. Arthur Koestler. *The Act of Creation*. New York:Macmillan, 1964.
5. Tobias Kötter, Kilian Thiel, and Michael R Berthold. Domain bridging associations support creativity. 2010.
6. Uwe Nagel, Kilian Thiel, Tobias Kötter, Dawid Piatek, and Michael R. Berthold. Towards discovery of subgraph bisociations. In Michael R. Berthold, editor, *Bisociative Knowledge Discovery*, volume 7250 of *Lecture Notes in Computer Science*, pages 263–284. Springer Berlin Heidelberg, 2012.
7. F Pereira. *Creativity and artificial intelligence: a conceptual blending approach*. Berlin: Mouton de Gruyter, 2007.
8. Paulo Ribeiro, Francisco C. Pereira, Bruno Marques, Bruno Leitão, and Amílcar Cardoso. A model for creativity in creature generation. In *Proceedings of the 4^{th} Conference on Games Development (GAME ON'03)*. EuroSIS / University of Wolverhampton, 2003.
9. Tom De Smedt. *Modeling Creativity: Case Studies in Python*. Uitgeverij UPA University Press Antwerp, 2013.
10. Don R Swanson. Fish oil, raynaud's syndrome, and undiscovered public knowledge. *Perspectives in biology and medicine*, 30(1):7–18, 1986.
11. Don R Swanson. Two medical literatures that are logically but not bibliographically connected. *Journal of the American Society for Information Science*, 38(4):228–233, 1987.

Case-Based Slogan Production

Martin Žnidaršič[1,3], Polona Tomašič[2,3], and Gregor Papa[1,3]

[1] Jožef Stefan Institute, Jamova cesta 39, SI-1000 Ljubljana, Slovenia,
[2] OLAII d.o.o., Pot za Brdom 100, SI-1000 Ljubljana, Slovenia
[3] Jožef Stefan International Postgraduate School, Jamova cesta 39, SI-1000 Ljubljana, Slovenia
martin.znidarsic@ijs.si
polona.tomasic@gmail.com
gregor.papa@ijs.si

Abstract. This paper presents a case-based approach to automated generation of slogans. We use a collection of cases out of which the selected ones get transformed and adapted to a new context that is represented by a textual description of the slogan's target. We also propose a methodology for evaluation and ranking of the final results. The approach is experimentally applied to two real-world use cases. The results indicate the ability of the approach to create slogan prototypes and reveal the issues to tackle in the next steps of solving this challenging problem.

Keywords: slogan generation, CBR, transformational adaptation, computational creativity, natural language

1 Introduction

Invention of slogans is a task that demands knowledge about the object of the slogan, its context and the intended message. However, such knowledge is not enough, as it has to be used in a creative way to produce a slogan that is novel, interesting and memorable. As a task that demands common knowledge and a high level of creativity, slogan generation is inherently difficult to automate. The aim of the work presented in this paper is to contribute to solutions of this challenging problem.

Our approach uses the texts of slogan cases to create new slogans that follow the grammatical structure of the initial cases, but use different words and phrases that are related to the slogans' target objects and contexts. As we use a collection of cases that we build upon and transform, this approach can be considered an application of case-based reasoning (CBR) in the domain of computational creativity.

It is very hard to automatically generate novel slogans that would be ready for use without further adaptations and corrections. This is not the case for simple template-based techniques[4], but these are not useful for our purposes

[4] Such as: "X, you have to buy it!" (put the name of the product in place of X).

as despite producing ready-made solutions, they are not innovative and do not produce context dependent results.

The outputs of the more innovative approaches often contain grammatical errors and semantic incoherencies. These outcomes can be considered slogan prototypes rather than slogans. They are useful in the conceptualization phase, as an addition to other techniques for production of solution drafts.

The case-based slogan generation is an example of a hybrid approach: it uses case texts, but not as rigid templates and it aims at incorporating some of the context of the slogan's target object. We have experimentally applied this methodology to two use cases. The relevant results and their assessments are provided in the paper, along with a discussion of the strong and weak points of our approach.

2 Related Work

Automatic generation of innovative creative artefacts that have a defined semantics is very challenging and the outcomes of such systems and methods are usually not ready for use without some sort of human curation. The computational creativity problems that are similar to slogan generation in terms of difficulty and representation are generation of jokes [1, 10], poems [4, 3] and generation of stories [2, 6], to some extent also the automatic generation of acronyms [11].

In the case of automated generation of slogans, there are only two lines of research work to the best of our knowledge: (I) the BrainSup approach by Özbal et. al. [9], which is the most well known and (II) the work by Tomašič et. al. [12], which is heavily influenced by the BrainSup approach, but complements it with the use of a genetic algorithm and additional evaluation functions. While the former expects relevant meta-data to be provided by the user, such as the keywords, the domain, etc., the latter is made to be completely autonomous. Consequently the reported results of BrainSup are of much higher quality.

In terms of CBR, the studies related to the work in this paper are the ones that are concerned with the use of textual data in CBR [13, 8]. Among these, we can also find some that are related by domain, such as the study on the use of CBR for story generation [5].

3 Slogan Collection

In our experiments we used a manually generated dataset of 5183 distinct items, each containing words transformed to lowercase, that appear in an example of a slogan.

Besides the words with their grammatical characteristics, we do not store other information, for example the particular product or product type that the slogan might be used for. Most of the slogans are used for promotion of the values and characteristics of a company and all its products, which might be numerous and diverse. As the characteristics of the products are reflected in the characteristics of the company and vice versa, it is usually difficult to determine whether a slogan is meant to be general or product-specific.

Cases in the collection are targeting diverse products and companies, from housing and financial services to food and cosmetics products. They differ a lot also in other characteristics, like length for example. The shortest one in the collection is only a 4 characters long word, while the longest one consists of 215 characters and 34 words. The median number of characters in the cases of this collection is 28, while the median number of words is 5.

4 Generation Process

The slogan generation process mostly follows the usual CBR steps [7] and is also presented in this fashion, by first describing the *retrieval* of similar cases, then the *adaptation and transformation* to suit a particular target and finally the *evaluation* and ranking of results.

4.1 Retrieval of Relevant Cases

Retrieval of cases that are relevant for a given problem is not trivial in our setting. Namely, the only input into our system is a textual description of the target (a company or a product), while our knowledge base consists of exemplary texts. In the absence of meta-data, which would, ideally, describe the context of the slogan and its target, we use only the textual information of the slogan examples and the target's textual description.

The retrieval process consists of two steps: (I) preprocessing of textual representations and (II) selection, based on similarity of words. First, the text of each slogan and the textual description of the slogan's target is transformed into a bag of words representation from which all the stopwords are removed (we have used the nltk library[5] for this purpose) and all the characters are transformed to lower case. Then, the items in the case-base are selected for adaptation, based on the matching of their words with the words in the target's description. If an item contains a word that appears also in the description of the target, it gets added to the collection of relevant cases. If it matches the target text in n words, it gets added n-times. We can describe this with the following equation:

$$n = |W_s \cap W_t|, \tag{1}$$

where the number of copies of an item in the collection of relevant cases (n) is expressed as the cardinality of the intersection among the words from the slogan (W_s) and the words from the target's description (W_t). If the intersection is empty, the particular item does not get added to the collection of relevant cases.

This way, the slogans with more words that appear also in the target's text have more instances in the collection of selected cases and consequently more of their (diverse) transformations represented in the final results.

[5] http://www.nltk.org/

4.2 Transformation

The selected items are transformed by insertion of words from the target's description. For each selected case-base item we exchange each of its words with probability p. Such a word gets exchanged with a randomly selected word from the target's description that has a matching part of speech (POS) tag while the punctuation marks are left unchanged. This way, repeated items that appear in the selection get transformed differently, as the exchanged words are in general different and their replacements are usually also different.

The exchange probability parameter p controls the level of diversity of the transformed items from the initial ones. Low values of p cause the resulting slogans to be more similar to their initial cases, thus they are less innovative and can be seen as imitations. High values of p on the other hand, cause the resulting slogans to be more novel, better connected to the target domain, but also more uncontrolled, with a higher frequency of grammatical errors and semantic incoherencies. As we prefer the results of the latter kind, we used $p = 0.75$ in our experiments.

4.3 Evaluation and Refinement

Due to the generation procedure, the transformed items often (depending on the parameter p) contain grammatical and semantic errors. To assess the results in this respect and to alleviate this problem, the outputs get evaluated and the final results of our approach are presented in a descending order of their evaluation scores.

For the purpose of evaluation, we represent each transformed item as a multiset[6] or a bag B_{ts} of bi-grams. For example:

```
you just have to buy this to be happy.
```

would be represented as:

```
{(you, just), (just, have), (have,  to), (to,  buy), (buy, this),
(this, to), (to, be), (be, happy)}.
```

Likewise, we create a multiset B of all the bi-grams that appear in all the examples in our case base and the input target text.

Each transformed item is then scored according to the number of its bi-grams from B_{ts} that appear also in B. This way, the results that have more bi-grams that appear in related texts (all the exemplary texts and the target's text) are scored higher. We expect that such results are constructed in a more meaningful way, at least locally in a word-to-word sense. However, by considering only the number of the matching bi-grams, the evaluation would be biased towards longer, and not necessarily more meaningful slogans. Therefore, our evaluation score S

[6] Namely, we want to allow a bi-gram to appear multiple times in our collection.

is a ratio of the number of the matching bi-grams and the number of all the bi-grams of the evaluated slogan:

$$S = \frac{|B_{ts} \cap B|}{|B_{ts}|} \ . \tag{2}$$

The final output of our approach are therefore the transformed selected slogans, ordered according to S.

5 Experiments

The approach presented in Section 4 was applied to two exemplary use cases: companies Sentinel[7] and Olaii[8]. Sentinel provides solutions for monitoring of a state of a boat or a fleet of boats, while Olaii is providing a system for payments and access management for events.

The input textual descriptions in both use cases were very raw, as we used all the text from their respective home pages, together with the boilerplate text such as the menu items, disclaimers, etc. The inputs were intentionally not cleaned in order to get an assessment of results from a very straightforward and realistic kind of use.

The first 10 and the last 10 results for Sentinel and Olaii are shown in Tables 1 and 2, respectively. According to our qualitative assessment, the outputs with the top ranks are clearly of higher quality than the bottom ranked ones, while the quality of the outputs is generally too low for practical use.

We have also experimented with the use of lower and higher values of the parameter p. As its impact is not very profound and can be observed only when one inspects a large number of outputs, we do not present these results here. Among the badly ranked outputs, as expected, the ones obtained with lower values of p (for example 0.50) are usually more readable and grammatically correct and the ones obtained with high values of p (like 0.90) are worse in this respect. Among the highly ranked outputs, lower values of p cause more results similar to initial ones to appear among the outputs, while the quality is not affected much even with the use of high values of p. This is most probably due to the evaluation and ranking procedure, which penalizes grammatically incorrect and incoherent slogans. The more abundant erroneous outputs that are expected to be produced with high values of p are thus prevented from appearing among the well ranked results. Therefore, it seems that it is sensible to use large values of p as this ensures production of less outputs that are similar to the already existing ones, while the evaluation and ranking prevents the comparatively larger amount of erroneous solutions to be present among the top results - the ones that are of interest in practice.

[7] http://www.sentinel.hr/

[8] http://cashless.olaii.com/

Table 1. Best and worst scored slogans that were generated for the Sentinel use case. The slogans with an equal bi-gram ratio score S are sorted according to the number of words (shortest first). An asterisk (*) is put in places where product names appear in the transformed slogans. Outputs that by chance match an initial item are removed from the ranked list.

Rank	Generated slogan	S
1	immediately what you need to be your best.	0.750
2	you enjoy our promises to you.	0.667
3	go * and warn the driving to you!	0.625
4	the one and only possible.	0.600
5	free enterprise with every issue.	0.600
6	simple boat to like you	0.600
7	an your security needs under one vacation.	0.571
8	you enjoy clearly when you enjoy it.	0.571
9	at the men in charge about eye.	0.571
10	battery you need from conception to reception.	0.571
...	...	...
338	a wholesome anchor with yet or detection.	0.000
339	a system alerts only , it clearly receives.	0.000
340	a alert is voltage holidays at us!	0.000
341	a most possible anchor need before all boat.	0.000
342	only it ’re going , it enjoy activating immediately.	0.000
343	your leave , information provides reliable , be at times..	0.000
344	sensors batteries you provides losing , and going , or sentinel.	0.000
345	sensors batteries you notifies going , and going , or sentinel.	0.000
346	entering healthy batteries about one eye , over all worries.	0.000
347	gps enjoy about , and they do away be out!	0.000

6 Discussion and Conclusion

The case-based generation of slogans is an approach that uses information from examples of solutions and aims at transforming them with regard to a target entity context into new slogans. Our method allows setting a parameter that controls the expected level of distortion of the original solution and adaptation to the target entity.

Our experiments indicate that the CBR-based approach can create artefacts, which can be used as prototype solutions for further (manual or automatic) refinement. Outputs of some experimental runs even produced good original slogans that could be used without further modification, such as:

```
the most reliable anchor of your solution.
```

which appeared among top ranked outputs with $p = 0.90$ for the Sentinel case. However, the experiments also show that the approach often results in erroneous and even meaningless solutions and that in general the amount of such noise (at the values of the distortion parameter that allow innovative slogans) is

Table 2. Best and worst scored slogans that were generated for the Olaii use case. The slogans with an equal bi-gram ratio score S are sorted according to the number of words (shortest first). All examples that are ranked 10 and have the same number of words are presented. An asterisk (*) is put in places where product names appear in the transformed slogans. Outputs that by chance match an initial item are removed from the ranked list.

Rank	Generated slogan	S
1	the best value of the event.	0.833
2	get to become a world.	0.800
3	the way you should have.	0.800
4	you find your visitors.	0.750
5	do you know you?	0.750
6	on all everything is a story to handle.	0.750
7	the first time is up the best.	0.714
8	all the * you are to reduce.	0.714
9	you can top-up the party to you.	0.714
10	who you find is what you are.	0.714
......		...
2136	an necessary few animal.	0.000
2137	benefits hard , once n't.	0.000
2138	more alerts , less habits.	0.000
2139	less visitors , less stations.	0.000
2140	a digital control to again.	0.000
2141	the product cards/wristbands are !	0.000
2142	you 're controlling better via n't.	0.000
2143	it will manage more good per you.	0.000
2144	them will steal the deeper you again.	0.000
2145	you better transfer more , you deposit Do.	0.000

substantial and further improvements are needed in order for the method to be applicable in practice.

A positive indication of the experimental results is the performance of the evaluation method, which seems to be useful, according to qualitative analysis of the result ranking. This is an encouraging result, as evaluation represents a big challenge in the problem domains of computational creativity. However, to strengthen this indication, which is currently supported only by the qualitative observations by the authors, a more elaborate evaluation procedure should be conducted with unbiased evaluators and hidden ranks. Such an evaluation is one of our highest priorities in further work, as the method could be valuable also in a wider context, if confirmed useful. Namely, the bi-gram ratio scoring could be applied also to other automatic slogan generation methods, and with appropriate adaptations, perhaps even in a wider array of similar problems.

Acknowledgments. This work was partly funded 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. The authors would also like to thank the anonymous reviewers for their valuable comments and suggestions.

References

1. Binsted, K., Bergen, B., O'Mara, D., Coulson, S., Nijholt, A., Stock, O., Strapparava, C., Ritchie, G., Manurung, R., Pain, H., et al.: Computational humor. IEEE Intelligent Systems (2), 59–69 (2006)
2. Callaway, C.B., Lester, J.C.: Narrative prose generation. Artificial Intelligence 139(2), 213–252 (2002)
3. Colton, S., Goodwin, J., Veale, T.: Full FACE poetry generation. In: Proceedings of the Third International Conference on Computational Creativity. pp. 95–102 (2012)
4. Gervás, P.: Computational modelling of poetry generation. In: Artificial Intelligence and Poetry Symposium, AISB Convention (2013)
5. Gervás, P., Díaz-Agudo, B., Peinado, F., Hervás, R.: Story plot generation based on CBR. Knowledge-Based Systems 18(4), 235–242 (2005)
6. Gervás, P., Lönneker-Rodman, B., Meister, J.C., Peinado, F.: Narrative models: Narratology meets artificial intelligence. In: International Conference on Language Resources and Evaluation. Satellite Workshop: Toward Computational Models of Literary Analysis. pp. 44–51 (2006)
7. Leake, D.B.: Case-Based Reasoning: Experiences, lessons and future directions. MIT press (1996)
8. Lenz, M.: Defining knowledge layers for textual case-based reasoning. In: Advances in Case-Based Reasoning, pp. 298–309. Springer (1998)
9. Özbal, G., Pighin, D., Strapparava, C.: BRAINSUP: Brainstorming Support for Creative Sentence Generation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. pp. 1446–1455. Sofia, Bulgaria (2013), `http://www.newdesign.aclweb.org/anthology-new/P/P13/P13-1142.pdf`
10. Ritchie, G.: Current directions in computational humour. Artificial Intelligence Review 16(2), 119–135 (2001)
11. Stock, O., Strapparava, C.: The act of creating humorous acronyms. Applied Artificial Intelligence 19(2), 137–151 (2005)
12. Tomašič, P., Žnidaršič, M., Papa, G.: Implementation of a slogan generator. In: The Fifth International Conference on Computational Creativity, ICCC 2014. pp. 340 – 343. Ljubljana, Slovenia (2014), `http://kt.ijs.si/publ/iccc_2014_proceedings.pdf`
13. Weber, R.O., Ashley, K.D., Brüninghaus, S.: Textual case-based reasoning. Knowledge Engineering Review 20(3), 255–260 (2005)

Conceptual Blending in Case Adaptation (Position Paper)

Amílcar Cardoso and Pedro Martins

CISUC, Department of Informatics Engineering
University of Coimbra, Coimbra, Portugal

Abstract. We propose that Conceptual Blending (CB) can play a role within the Case-Based Reasoning (CBR) paradigm, particularly in the *Reuse* and *Revise* tasks of the classic model of the problem solving cycle in CBR, as an alternative adaptation mechanism that may provide suitable solutions in computational creativity setups, where novel and surprising solutions are sought. We discuss how a particular computational implementation of CB can intervene in the CBR cycle, and use the results of an experiment made in the past to illustrate the aproach. We focus our attention on graph-based structured cases. Other case representations could also be considered in the future.

1 Introduction

The Conceptual Blending (CB) theory [3] intends to explain several cognitive phenomena related to the creation of ideas and meanings. A key element in this theory is the *mental space*, which corresponds to a temporary and partial structure of knowledge built for the purpose of local understanding and action. The CB framework relies on a network comprised of at least four connected mental spaces (Figure 1). Two or more of them correspond to the *input spaces*, which are the initial domains, i.e., the content that will be blended. Then, a cross-space mapping, i.e., a partial correspondence between the input spaces, is established. The correspondences between elements of the different input spaces is not arbitrary; elements are only matched if they are perceived as similar in some way. This association is reflected in another mental space, the *generic space*, which contains elements common to the different input spaces, capturing the conceptual structure that is shared by the initial mental spaces. The result of the blending process is the *blend*, a new mental space that maintains partial structures from the input spaces, combined with an emergent structure.

In this position paper, we propose that Conceptual Blending can play a role within Case-Based Reasoning, particularly in the *Reuse* and *Revise* tasks of the classic model of the problem solving cycle in CBR, known as the "4 REs" [1], as an alternative adaptation mechanism that may provide better solutions in computational creativity setups, and possibly also for problem solving. We will focus our attention on graph-based structured cases (like in [7]), but we think the approach could also be adapted to other case representations [2]. To better

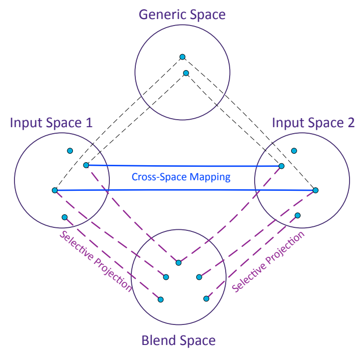


Fig. 1. The original four-space conceptual blending network [4].

explain our idea, we will use an implementation of the CB mechanism called Divago [6], previously developed by our team.

After the current introduction, we will briefly describe Divago in Section 2 and present our proposal in Section 3. In Section 4 we draw some conclusions.

2 Divago

The CB framework has served as the basis for several artificial creative systems. To discuss the role of CB within the CBR cycle, we focus on the Divago architecture [6], which relies on one of the most thorough and detailed computational models of CB to date.

The Divago framework works on a multi-domain knowledge base where the basic representation formalism is the *concept map*, a semantic network that denotes the relationship between the concepts of a given domain. It is composed of several modules (Fig. 2) that reflect the different stages of the CB mechanism.

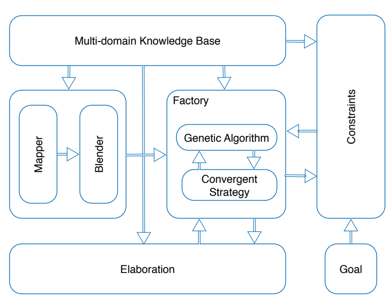


Fig. 2. Divago's architecture.

The process starts by feeding a pair of input spaces (domains) from the knowledge base into the *Mapper* module, which is responsible for performing the selection of elements for projection. Such selection is achieved by means of a partial mapping between the input spaces using *structural alignment*. This operation looks for the largest *isomorphic* (structurally equivalent) pair of subgraphs contained in the input spaces. Each mapping is a set of mapping relations $m(x, y)$ between two concepts, one of each input space.

For each resulting mapping, the *Blender* module performs a projection operation into the blended space: for each $m(x, y)$ in the mapping, it produces a nondeterministic projection choice between x, y, $\emptyset$ and $x|y$ (which means *both* x *and* y); each combination of choices is the *seed* of a possible blend (to be completed and elaborated in the next stages). This process results in a graph structure (the *blendoid*) that includes all projection choices and thus represent the search space for all the blends that may result from the mapping.

The *Factory* module is responsible for exploring this search space. It is based on a variation of a genetic algorithm (GA) that uses the *Elaboration* module to enrich blends with additional knowledge and the *Constraints* module to assess their quality. This module provides an implementation of the *optimality principles* (a set of principles that ensure a coherent and highly integrated blend [3]). When an adequate solution is found or a pre-defined number of iterations is attained, the Factory stops the execution of the GA and returns the best blend. The *Constraints* module acts, thus, as the "fitness function" of the algorithm.

3 Conceptual blending in case-adaptation

The classic model of the problem solving cycle in CBR, known as the "4 REs", comprises 4 tasks: *Retrieve*, *Reuse*, *Revise* and *Retain* [1]. In the core of the process lies a *case base* of stored past experiences, each one of them comprising a problem description and the respective solution.

Although cases can be represented in many different ways [2], we will consider the situation where a structured representation is used, like for instance [7]. In particular, we will assume that there are *relations* between *attributes*. Some of them allow for hierarchical organisations (e.g., *isa* and *partwhole*), others induce a network structures (e.g., *purpose*, *shape*, relations for relative position). Table 1 describes, using a Prolog-like notation, a fragment of a case for a "House", where such relations occur. The right column is a partial description of the attribute/value pair part of the same case.

Coming back to the "4 REs" cycle, the reasoning process starts with a new problem specification being given to the first task, *Retrieve*, which seeks for stored cases with similar problem descriptions, using some similarity criterion. The result is a list of retrieved cases, of which one can be selected as having the most similar problem description to the given problem. In the general case, the similatity is not absolute and differences with the given problem description exist. This requires that the retrieved case is subject to some sort of adaptation in the task *Reuse*, trying to compensate for the differences with the given problem description. *Revise* will be responsible for evaluating the quality of the result.

Table 1. Fragment of the "House" case.

isa(house,physical_structure)	part_whole(door, house)	instance_of(r1, roof)
isa(door, physical_object)	part_whole(window, house)	instance_of(b1, body)
isa(window, physical_object)	part_whole(roof, house)	instance_of(d1, door)
isa(roof, physical_object)	part_whole(body, house)	instance_of(w1, window)
isa(body, physical_object)	part_whole(room, house)	shape(r1, triangle)
isa(observation, task)	purpose(body, container)	shape(b1, square)
isa(protection, task)	purpose(door, entrance)	shape(w1, square)
isa(entrance, task)	purpose(window, observation)	
isa(container, physical_object)	purpose(roof, protection)	

Now, let us assume that the retrieved case, c_r, is the one described in Table 1. This might happen, for instance, if the case base was composed of descriptions of houses, the problem to solve was to find a house description according to a given specification and the specification of c_r was the most similar to the given one. Let us also assume that we are in a creative setup, where we want to find ideas for houses that, although satisfying the specification, are novel and surprising. Our proposal is to seek for surprising solutions by processing the adaptation through *blending c_r with knowledge from a different domain*. The result will be a case that shares part of its description with the retrieved case, but includes contributions from the other domain. Such contributions may, for instance, fill existing gaps in c_r, substitute part of its structure, etc. As we will see, the result may be more or less *divergent* from the original domain of "houses" according to how we control the blending process and "how far" from "houses" the other domain is. The domain to use in this process may be chosen by the user, or may result from a contextual analysis whose discussion is outside the scope of this paper. We argue that Divago can deal with the process in a suitable way.

To illustrate our proposal, we re-visit the experiment described in [5], where the blend of two domains, "boats" and "houses", is explored using just the modules *Mapper* and *Blender* of Divago, with the aim of studying their *generation potential*. The situation is very similar to the one described in the previous section, as c_r, the "House" case, can be seen as an instance of the original "houses" domain. With this analysis, we intend to illustrate how the "House" case can be merged with the domain "boats".

In the experiment, the *blendoid* resulting from the most frequent mapping represents a wide variety of instances for "boat-house". We show six of them in Figure 3, where the visual representation of c_r is shown on the left.

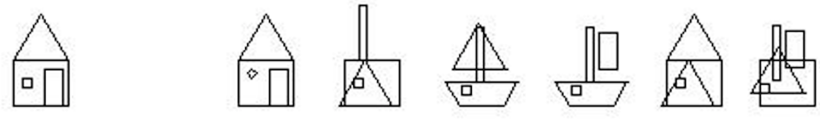


Fig. 3. The retrieved "House" case and six possible blends with the "boat" domain.

We can see that the weight of the "boats" domain in the blends varies a lot. The *divergence* of the blends from the stereotypical description of a Boat and from c_r also varies a lot, from a house with a hatch instead of a window to a house with a sail instead of a door and a mast instead of a roof.

In Divago, the GA-like search for blends is guided by an implementation of a variation of the "optimality principles" proposed in the CB theory, which favours the coherence of the resulting blends. In the context of this proposal, however, a metric for the similarity with the original problem specification should also be taken into account, and possibly assume a prevailing weight in measuring the quality of the blends.

4 Conclusions

We argued that Conceptual Blending, and in particular its computational implementation Divago, can provide an alternative adaptation mechanism for the *Reuse* and *Revise* tasks of the classic CBR model. The idea is to *blend* the case selected in the *Retrieve* task with knowledge from a different domain. This may prove especially effective in computational creativity contexts, where it may provide an iterative divergence mechanism coupled with evaluation. The criteria for evaluating each possible blend may combine measures of coherence with measures of distance to the given problem specification. This is a preliminary proposal in the context of a Position Paper. Definitely, further research is needed to understand its limits.

Acknowledgements. The authors acknowledge the financial support from the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the ConCreTe FET-Open project (grant number 611733) and the PROSECCO FET-Proactive project (grant number 600653).

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI communications 7(1), 39–59 (1994)
2. Bergmann, R., Kolodner, J., Plaza, E.: Representation in case-based reasoning. Knowl. Eng. Rev. 20(3), 209–213 (Sep 2005)
3. Fauconnier, G., Turner, M.: Conceptual integration networks. Cognitive Science 22(2), 133–187 (1998)
4. Fauconnier, G., Turner, M.: The Way We Think. New York: Basic Books (2002)
5. Pereira, F.C., Cardoso, A.: The boat-house visual blending experience. In: Procs 2nd. Workshop on Creative Systems, ECAI 2002, Lyon, France (2002)
6. Pereira, F.C.: Creativity and AI: A Conceptual Blending approach. Ph.D. thesis, FCTUC, Universidade de Coimbra, Portugal (2005)
7. Plaza, E.: Cases as terms: A feature term approach to the structured representation of cases. In: Case-Based Reasoning Research and Development, pp. 265–276. Springer (1995)

Calibrating a Metric for Similarity of Stories against Human Judgment *

Raquel Hervás, Antonio A. Sánchez-Ruiz, Pablo Gervás, Carlos León

Dep. Ingeniería del Software e Inteligencia Artificial
Universidad Complutense de Madrid (Spain)
raquelhb@fdi.ucm.es, antsanch@fdi.ucm.es, pgervas@sip.ucm.es, cleon@fdi.ucm.es

Abstract. The identification of similarity is crucial for reusing experience, where it provides the criterion for which elements to reuse in a given context, and for creativity, where generation of artifacts that are similar to those that already existed is not considered creative. Yet similarity is difficult to compute between complex artifacts such as stories. The present paper compares the judgment on similarity between stories explained by a human judge with a similarity metric for stories based on plan refinements. The need to identify the features that humans consider important when judging story similarity is paramount on the road to selecting appropriate metrics for the various tasks.

Keywords: similarity, novelty, stories, plans.

1 Introduction

Appropriate metrics for similarity are fundamental tools in many fields of Artificial Intelligence. For instance, there are several data mining and machine learning methods that are based on the similarity between the elements being considered. In case-based reasoning, similarity metrics are crucial for the retrieval and reuse of previous cases. Similarity is also fundamental for computational creativity because artifacts that are very similar to previously existing ones might not be considered creative. For this reason, it is important to take into account whether the metrics considered for a particular task adequately represent the concept of similarity that humans faced with the same task would apply. The present paper compares the judgment on similarity between stories explained by a human judge with a particular similarity metric for stories. The main goal is to identify which of the features that a human considers when evaluating story similarity are already taken into account by the metric, and which ones are not. The results of this comparison should provide a check list that might later on be applied to evaluate the appropriateness of other metrics.

* The research reported in this paper was partially supported by the Project WHIM 611560 funded by the European Commission, Framework Programme 7, the ICT theme, and the Future and Emerging Technologies FET programme; and by the Spanish Ministry of Economy and Competitiveness under grant TIN2014-55006-R.

We focus on the structural similarity of stories represented as plans composed of actions corresponding to the events in the story. In order to do so, we apply a similarity metric based on plan refinements and compare the obtained results for a pair of stories with the similarities found by a human expert. The key point of this comparison is that the metric does not only calculate a numerical similarity between the compared stories, but provides a report of the found similarities. This report is then compared with the observations obtained by the human expert. The comparison allows us to see if the automatic metric has been able to grasp the same features the expert considered important, and if structural similarity is enough for comparing computer-generated stories.

2 Previous Work on Similarity for Stories

Existing work on similarity for stories has focused on two different axes: story similarity for retrieval and classification of stories, and story similarity applied to the assessment of their novelty in a computational creativity setting.

2.1 Similarity Metrics for Story Generation

In general, there is relative consensus on the fact that comparing stories can be made at different levels. Comparing stories at a relatively abstract level is common, to the point of comparing not the exact sequence of events but the overall plot, or even the relations between the characters. This aspect of narrative has been addressed by structuralist and cognitive Narratology.

In particular, comparing narratives has been a long term goal of Computational Narrative, and several approaches have been taken with varying results [2, 10, 8]. Different aspects beyond pure literary composition have been tackled: structure alignment in bioinformatics [1], event mapping [3], and other approaches like considering story similarity in terms of the common summary that might be abstracted from the two stories being compared [9].

2.2 Similarity Metrics for Assessing Novelty of Stories

With respect to the assessment of creativity, a fundamental pillar is whether the results of a creative process have produced novel artifacts [14]. Research on the evaluation of creativity has addressed this point as an important requirement for the scientific exploration of creativity, and an important one for computational approaches. In [11], novelty of a given story is assessed in terms of new elements that appear in the story, or instances where existing elements have been replaced by elements of a different type. In [12], novelty of stories is considered in terms of their differences with an initial set of reference stories, based on the sequence of actions, the structure of the story in terms of emotional relations and tensions between the characters, and the occurrence of repetitive patterns.

Story 1	Story 2	Common structure
shows id371 id372	**declare-war id818 id819**	declare-war id818 id819
offers-exchange id371 id372 id373	sings id207 murder	decides-to-react ?x1
not-perform-service id373	**decides-to-react id142**	sets-out ?x1
negative-result id373	**sets-out id142**	wins ?x1
consumes id373 id44	**wins id142**	brings-peace ?x1
acquires id373 magical-abilities	**brings-peace id142**	arrives ?x1 ?x2
declare-war id818 id819	**arrives id142 id730**	disguised ?x1
dispatches id189 id373	**disguised id142**	unrecognised ?x1
tells id189 id373 past-misfortune	**unrecognised id142**	claims id672 won id818
decides-to-react id373	**claims id672 won id818**	sets ?x3 ?x1
sets-out id373	**sets id165 id142**	involves difficult-task ?x4
wins id373	**involves difficult-task strength**	solve ?x1 difficult-task
brings-peace id373	**solve id142 difficult-task**	before dead-line
arrives id373 id728	**before dead-line**	returns ?x1
disguised id373	**returns id142**	arrives ?x1 id730
unrecognised id373	**arrives id142 id730**	disguised ?x1
claims id672 won id818	**disguised id142**	unrecognised ?x1
sets id161 id373	**unrecognised id142**	claims id672 won id818
involves difficult-task kissing	**claims id672 won id818**	exposed id672
marked id373	**exposed id672**	not-solve id672 difficult-task
solve id373 difficult-task	**not-solve id672 difficult-task**	
before dead-line	new-physical-appearance id142	
returns id373	punished id818	
arrives id373 id730	tied-to id818 horse-tail	
disguised id373		
unrecognised id373		
claims id672 won id818		
exposed id672		
not-solve id672 difficult-task		

Table 1: Table of events in each of the stories and the shared set of events.

3 A Callibration Exercise for Story Similarity

Although there are many possible representations for stories and many different metrics have been considered for story similarity, the present effort has been focused on a particular representation format as used by an existing story generator, and a specific metric that allows automatic computation. These choices were circumstantial on ease of access and are not considered optimal, but the effort should produce valuable insights that can later be extended to other alternatives.

3.1 Story Representation in the Propper System

The Propper system [5] constitutes a computational implementation of a story generator based on Propp's description of how his morphology might be used to generate stories [13]. It produces stories as a sequence of states described in terms of predicates that hold in the state. Characters, objects or locations are represented as unique identifiers in the predicates. This representation format has been considered generic enough to allow for an initial calibration exercise, considering that other formats may easily be converted into this one.

The representation includes predicates representing narrative events and predicates describing properties of the characters that hold in particular states of the story. These appear jointly in the stream of predicates for the story, but have been separated in the presentation of stories in this paper for clarity.

The predicates presented here result from an effort of reverse engineering of the stories that Propp describes as examples of the application of his framework to analyse existing Russian folk tales.

The first two columns of Table 1 present two examples of the stories produced by the Propper system. Predicates in this table describe actions or events in the story. Table 2 represents non-narrative facts that are true for the arguments of the actions in Table 1.

Story 1	Story 2	Common structure
hero id373	*villain id818*	*villain id818*
donor id371	*victim id819*	*victim id819*
magical-agent id372	*hero id142*	*hero ?x1*
magical-agent id44	*seeker-hero id142*	*location ?x2*
villain id818	*location id730*	*false-hero id672*
victim id819	*court id730*	*unknown ?x3*
seeker-hero id373	*groom id142*	*task-type ?x4*
dispatcher id189	*false-hero id672*	*court id730*
location id728	*location id730*	
home id728	*court id730*	
apprentice id373 artisan	*groom id142*	
false-hero id672		
location id730		
court id730		
groom id373		

Table 2: Table of characters, locations and objects in the two stories and the shared set

3.2 Human Interpretation of the Stories

In order to compare the human interpretation of the stories with an automatically extracted report, we asked a human expert to write both stories in English and compare them. It is important to mention that the expert was familiar with this type of representation based on predicates, but she had to figure out the meaning of the predicates based solely on their names.

Story 1

This story has the following main characters: a hero (373), a villain (818), and a false hero (672). In addition, a donor (371), a victim (819) and a dispatcher (189) appear as secondary characters.

The hero (373) is first offered a magical agent by a donor (371) if he performs a service. He does not perform the service but he obtains another magical agent anyway, which he consumes to acquire magical abilities.

Then, a villain (818) appears who declares war to a victim (819). The victim does not appear again.

Meanwhile, a dispatcher (189) talks about a past misfortune. The hero decides to react, sets out and wins (the war?), bringing peace with him. After that, the hero goes home, but he is disguised as the apprentice of an artisan and is not recognised. He finds a false hero (672) at home, who claims that he defeated the villain.

The hero is marked, solves a difficult task and returns to the court, this time disguised but as a groom. The false hero still claims that he defeated the villain, but he is exposed and it is known that he did not solved a difficult task.

Story 2

This story has the following main characters: a hero (142), a villain (818), and a false hero (672). In addition, a victim (819) appears only at the beginning.

The story starts with the villain (818) declaring war to the victim (819). The hero (142) decides to react, becomes a seeker hero, sets out and wins (the war?). He brings peace and arrives to the court. But he is disguised as a groom and he is not recognized.

At the court, the false hero (672) claims that he defeated the villain. Someone (165) sets the hero a difficult task that involves strength. He solves the difficult task before the deadline, and returns to the court. Again he is disguised as a groom and he is not recognized.

And again, the false hero claims that he defeated the villain. However, the false hero is exposed and does not solve a difficult task. The hero gets a new physical appearance (undisguised?), and the villain is punished being tied to a horse tail.

Next, we asked the expert to compare both stories and describe the main similarities and differences between them.

Both stories are similar in their characters and roles: a hero, a villain, and a false hero who claims to have defeated the villain.

In addition, in both stories the villain declares war to a victim, and the hero wins the war and brings peace. After that the hero returns (home or to the court) disguised (as a groom or as an apprentice), and he finds that a false hero claims to have defeated the villain. But at the end the false hero is exposed in both stories. Also, in both stories the hero makes two different journeys: one to win the war and return home/court, and one to solve a difficult task and then returning to court.

From the point of view of the differences, Story 1 involves magic. The hero tries twice to obtain a magical agent, and the second time he achieves it and gets magical abilities. However, they are not used in the story. The main difference in Story 2 is that at the end the villain is explicitly punished by being tied to a horse tail.

It is interesting to note that the first things mentioned by the expert both in the descriptions and the comparison are the characters, although in the comparison only the most important characters are mentioned, as the others are considered less important for the plot.

In addition, the descriptions are based on the most important events in the story, so not all events are considered equally important. The comparison also shows that there is a high similarity between both stories in terms of characters and some of the narrative arcs. For example, the hero returns in both stories but to different places and with different disguises. However, these differences (place and disguise) are not considered as important and the expert finds similarity in what is happening even when the stories are not exactly the same.

One of the main differences between the stories is that one of them involves magic, but it is not considered so important because magic is not used in the rest of the story. Finally, the differences in the endings are explicitly addressed in the comparison. This means that the end of the story is an important part of it.

3.3 Computing the Common Structure of Two Stories using Plan Refinements

A story in its more basic form can be represented as a sequence of actions, i.e., as a *plan*. There are different approaches to compute the similarity of two plans. In this paper we use the similarity measure based on plan refinements presented in [15] because it does not only provide a numerical similarity value but an explicit description of the common structure shared by both plans. This common structure can be seen as a directed graph in which each node represents an action and each directed edge represents an ordering constraint. Two actions are connected in the graph only if both actions appear in that order in the plans being compared.

Besides the actions and their order, this similarity measure also considers the action parameters and, if they are different in both plans, it is able to infer their common type according to a domain taxonomy. In this way, we are able to detect objects, characters and locations in different stories that have a different name but play the same role in the story.

The similarity measure computes this common structure performing successive refinements in the space of partial plans [7]. There are five different types of refinements that specialize a partial plan: to add a new action, to add a new ordering constraint between two existing actions, to specialize the type of a variable representing an action parameter according to a domain taxonomy, to unify two different variables, and to replace a variable with a domain constant.

The similarity measure works as follows. Let us suppose we want to compare two plans (or stories) p_1 and p_2. The similarity measure begins with an empty partial plan (a plan with no actions) that represents any possible plan and thus it is more general than p_1 and p_2. Then the partial plan is specialized using a refinement operator (adding new actions and ordering constraints or specializing the action's parameters) until we reach another partial plan that cannot be specialized anymore while being more general that both p_1 and p_2. This partial plan is the *most specific generalizer* of p_1 and p_2, $MSG(p_1, p_2)$, and represents the common structure shared by the two plans. The length of the refinement chain from the empty plan to the $MSG(p_1, p_2)$ is an indicator of how similar the two plans are. In the same way, the length of the refinement chain from the $MSG(p_1, p_2)$ to each one of the two plans is an indicator of how much information is contained only in one of them but not in the other. The similarity value is computed as the ration between the amount of information shared and the total amount of information contained in the two plans.

The last columns of Tables 1 and 2 show the common structure computed by the similarity measure. In this case, the two stories are very similar and the inferred common graph of actions is so simple that, in fact, it can be represented as a sequence of actions. Constants representing characters, locations and objects common to both stories are kept in the common structure, and the other constants are replaced by variables with generalized types (variable names begin with '?').

The common structure of both stories could be summarized as follows. A villain declares war on a victim, what triggers the intervention of a hero that defeats him and brings peace back. Then the hero travels disguised and see how a false hero claims that he, and not the original hero, has defeated the villain. The hero leaves, solves a difficult task before some deadline, and comes back disguised. The false hero is exposed in court because he was not able to solve the difficult task.

4 Discussion

There are a number of issues that the similarity metric considered here does not take into account.

First, the point in the story in which a particular sequence of actions takes place may lead to different results. A marriage at the start of the story sets the scene for later actions, but at the end of the story it usually acts as a reward for the efforts of some character. This influence of context is not considered in the metric that has been described.

Second, some events are more significant than others. The presence of a murder in a given story is more significant than that of more mundane events such as setting off on a journey. This aspect might be captured by some kind of weighting of the importance of specific events. The described metric does not allow for this type of behaviour.

The judgment expressed by the human placed considerable emphasis on the relative importance of the elements that appear in the stories. Characters are mentioned first, then specific actions. In both cases, a certain degree of abstraction is applied to identify conceptual similarity even between instances that are different. This suggests that taxonomical reasoning might be a useful tool for assessing similarity and that, as expected, abstraction is fundamental in story similarity.

These two aspects suggest that automatic story comparison needs to address *lifting* between different levels of abstraction to be able to match those features that humans are able to match. It also seems that the abstract matching at different levels is a fundamental cognitive tool for comparing stories in humans.

This conclusion relates to the approach in [9] of considering similarity between stories in terms of a shared summary, but extended to summarisation with an important degree of abstraction. The work in [11], by virtue of being based on description logic ontologies, does include the possibility of taxonomical reasoning being applied in the process of measuring similarity. It is clear that this particular approach should be explored in more detail in future work.

The version of the Propper system that has been employed here provides only limited description of the characters. The descriptions considered are restricted to specification of the roles played in the narrative by particular characters, and a number of properties of particular arguments that are relevant for the correct chaining of later actions with their context of occurrence via their set of preconditions.

An important problem from the point of view of assessing the novelty of creative processes is the need to consider an existing set of artifacts as a reference. Generated artifacts are only novel if they are not similar to existing ones. However, from a computational point of view, the approach of keeping a record of all existing artifacts of a given type, and computing the similarity of any newly generated artifacts with this set is not practical [4]. Indexing solutions may be used to improve efficiency, but even so, solutions based on some level of abstraction, away from specific instances and addressing more generic characterisations of the artifacts (in this particular case, stories) would prove more practical in this context. Conformance or departure from Concepts such as conventional endings, genre conventions, or character stereotypes may play a fundamental role in assessing the novelty of stories beyond sequences of actions.

Overall, it seems that there are a number of aspects of stories that are relevant when attempting to establish similarity between two instances of story. Just how many such aspects should be included in a particular implementation as a similarity metric may depend substantially on the purpose for which it is intended. In the particular case of similarity metrics employed for case-based reasoning, the choice of which aspects of similarity to model should be guided by the particular aspects of the case that will be reused. If the cases are intended to provide story structure, the similarity should focus on story structure. If the cases are intended to inform decisions on the set of characters to employ, the similarity should focus on the set of characters. In relation to the point raised above concerning abstraction, it is important to note that focusing on particular aspects of story similarity may require specific types of abstraction to implement the described lifting operation. Where similarity metrics are used for evaluating novelty in Computational Creativity settings, their use is much broader and it becomes more difficult to focus on particular aspects. Nevetheless, as it is very important to consider issues of efficiency, abstraction as means of reducing the range of attributes that need to be compared will clearly play a fundamental role in practical implementations.

5 Conclusions and Future Work

The present work describes a process by which a computational system for computing the similarity between narrative structures is compared and calibrated against human judgment.

A number of issues considered by the human judge but not covered by the system have been discovered. These should be considered as a check list for the consideration of alternative metrics, and possibly as driving guidelines for the development of more elaborate metrics specific to the assessment of story similarity.

The work described in this paper has addressed sequential single narrative threads. More complex narratives usually involve parallel story lines which merge or split at several points in the overall narrative. Whether the current metrics are valid for comparing similarity between this kind of narratives or not is yet an open question. Additionally, the use of different structures for stories also opens a new path, namely the application of the current process to stories that, while outputting an equivalent format, are generated by other story generation systems, probably conveying different semantics in the sequence of events, and possibly richer relations between characters.

From this point of view, more recent versions of the Propper system [6] address specifically the description of characters as they occur in the story, and they should be explored in further work to extend the metric for similarity to consider differences between the characters of two stories. For that work, it may be necessary to focus on differences between characters fulfilling equivalent narrative roles in the different stories.

State is also fundamental in narrative composition and analysis. Narrative understanding of statements like "John squashed the spider" heavily depend on the relation between John and the spider (was it his mascot?). This kind of information must be taken into account in a general model of story similarity.

In all cases, further research must look into more metrics for story comparison and employ more experts to analyse how humans evaluate narratives. Following the intuition that we, as humans, perform a complex set of comparisons for evaluating similarity at different levels can lead to the discovery of plausible metrics and plausible aggregation methods into one single judgment.

References

1. Fay, M.: Story comparison via simultaneous matching and alignment. In: Workshop on Computational Models of Narrative, 2012 Language Resources and Evaluation Conference (LREC'2012). Istambul, Turkey (2012)
2. Fisseni, B., Lowe, B.: Which dimensions of narratives are relevant for human judgments of story equivalence? In: Workshop on Computational Models of Narrative, 2012 Language Resources and Evaluation Conference (LREC'2012). Istambul, Turkey (2012)
3. Fisseni, B., Lowe, B.: Event mappings for comparing formal frameworks of narratives. Logique et Analyse (57) (2014)

4. Gervás, P.: Dynamic inspiring sets for sustained novelty in poetry generation. In: Second International Conference on Computational Creativity. México City, México (2011)
5. Gervás, P.: Propp's morphology of the folk tale as a grammar for generation. In: Workshop on Computational Models of Narrative, a satellite workshop of CogSci 2013: The 35th meeting of the Cognitive Science Society. Universität Hamburg Hamburg, Germany (2013)
6. Gervás, P.: Computational drafting of plot structures for russian folk tales. Cognitive Computation (2015)
7. Kambhampati, S., Knoblock, C.A., Yang, Q.: Planning as refinement search: A unified framework for evaluating design tradeoffs in partial-order planning. Artificial Intelligence 76(1), 167–238 (1995)
8. Krakauer, C., Winston, P.: Story retrieval and comparison using concept patterns. In: Workshop on Computational Models of Narrative, 2012 Language Resources and Evaluation Conference (LREC'2012). Istambul, Turkey (2012)
9. Kypridemou, E., Michael, L.: Narrative similarity as common summary: Evaluation of behavioral and computational aspects. LLC 29(4), 532–560 (2014), http://dx.doi.org/10.1093/llc/fqu046
10. Michael, L.: Similarity of narratives. In: Workshop on Computational Models of Narrative, 2012 Language Resources and Evaluation Conference (LREC'2012). Istambul, Turkey (2012)
11. Peinado, F., Francisco, V., Hervás, R., Gervás, P.: Assessing the novelty of computer-generated narratives using empirical metrics. MINDS AND MACHINES 20(4), 588 (2010)
12. Pérez y Pérez, R., Ortiz, O., Luna, W.A., Negrete, S., Pealoza, E., Castellanos, V., vila, R.: A system for evaluating novelty in computer generated narratives. In: Proceedings of the Second International Conference on Computational Creativity. pp. 63–68. Mxico City, Mxico (2011)
13. Propp, V.: Morphology of the Folk Tale. Akademija, Leningrad (1928)
14. Ritchie, G.: Some Empirical Criteria for Attributing Creativity to a Computer Program. Minds & Machines 17, 67–99 (2007)
15. Sánchez-Ruiz, A.A., Ontañón, S.: Least common subsumer trees for plan retrieval. In: Case-Based Reasoning Research and Development - 22nd International Conference, ICCBR 2014, Cork, Ireland, September 29, 2014 - October 1, 2014. Proceedings. pp. 405–419 (2014)

Creative Systems as Dynamical Systems

Alessandro Valitutti

School of Computer Science and Informatics, University College Dublin,
Belfield, Dublin D4, Ireland
alessandro.valitutti@ucd.ie

Abstract. In this paper, we discuss ideas for characterizing a case-based generative system as "creative". Focusing on a specific generator of graphics, we performed a qualitative exploration of the space of solutions. The emerged intuition is that the set of configurations generated by the program can be viewed both as the conceptual space of a creative system and the phase space of a dynamical system. In the context of this analogy, we hypothesize that a higher degree of creativity can be ascribed to the search paths allowing the system to reach new basins of attractions.

1 Introduction

Case-based reasoning (CBR) is a type of problem solving in which a new solution is found through the retrieval of a similar available case and the adaptation of the related solution [1].

Let us suppose to have a computer program for the generation of artworks such as graphics, musical pieces, or poems, and a set of generative parameters. Given a set of known examples, a different initialization of the parameters should allow the system to produce different corresponding instances of the same type of artifact. However, the production of new artifacts does not necessarily imply that they would be recognized as original and valuable. In this paper, we discuss ideas for characterizing the re-use of past solutions, performed by a case-based generative system, as "creative".

An artwork generator can be framed in the context of ideas on *creative systems* introduced by Boden [2], formalized by Wiggins [12] and further extended by Ritchie [11]. In this context, the case-based adaptive process can be viewed as a type of *exploratory creativity*, i.e. a search in the space of artifacts or *conceptual space*, where the set of past examples are the *inspiring set*. Ideally, the output of the search should be an artifact provided with a form of value and expressing the balance between familiarity and novelty described by Giora as *optimal innovation* [4].

Focusing on a specific generator of graphics, we performed a qualitative exploration of its generative parameters, described in the next section. The rest of the paper discusses the insights inspired by this example.

2 Exploring the Space of Fractal Trees

We focused on an algorithm for the visual representation of a *fractal tree*, a fractal geometrical shape defined by recursion as follows: *(1) Draw a trunk; (2) At the end of the trunk, split by some angle and draw a prefixed number of branches; (3) Repeat at the end of each branch until a sufficient level of branching is reached*[1]. The original code of the program[2] was implemented in *Processing* programming language [10]. For the mathematical details, we refer the reader to Mandelbrot's treatment [8, pp.151-161]. The shape depends on the value of two parameters representing the angle between two adjacent branches and the rotation angle performed on both of them, respectively. Their values are associated to the two coordinates of the mouse cursor in the output window. In this way, moving the cursor in different points of the screen, it is possible to generate an unlimited number of configurations.

In order to show the set of possible configurations in a small portion of the output window, we modified the code in such a way to draw a small square and to map the configurations to the coordinates of its internal points.

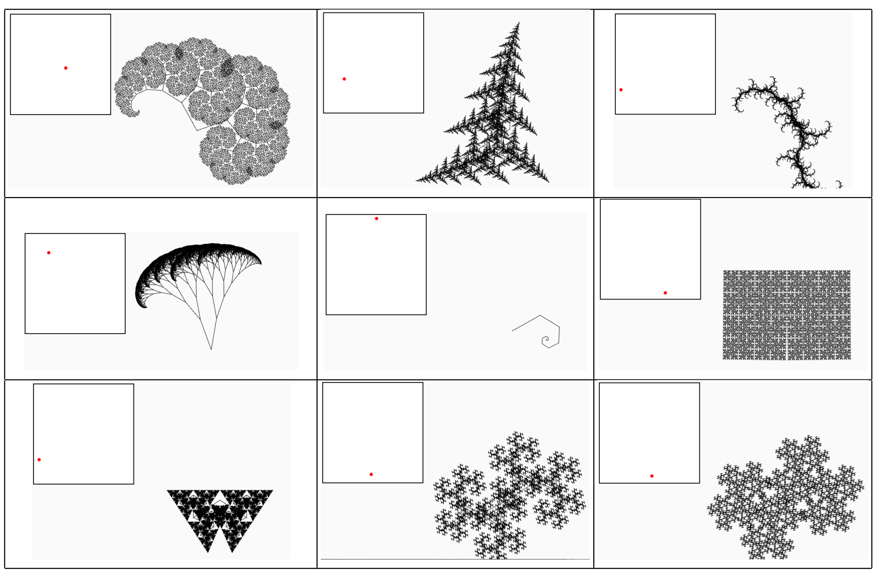


Fig. 1. Examples of configurations generated by the position of the cursor in different regions of the conceptual space mapped in the square.

[1] This version of the algorithm description is reported on `http://rosettacode.org/wiki/Fractal_tree`

[2] The code of the original program is available at `http://www.openprocessing.org/sketch/5631`.

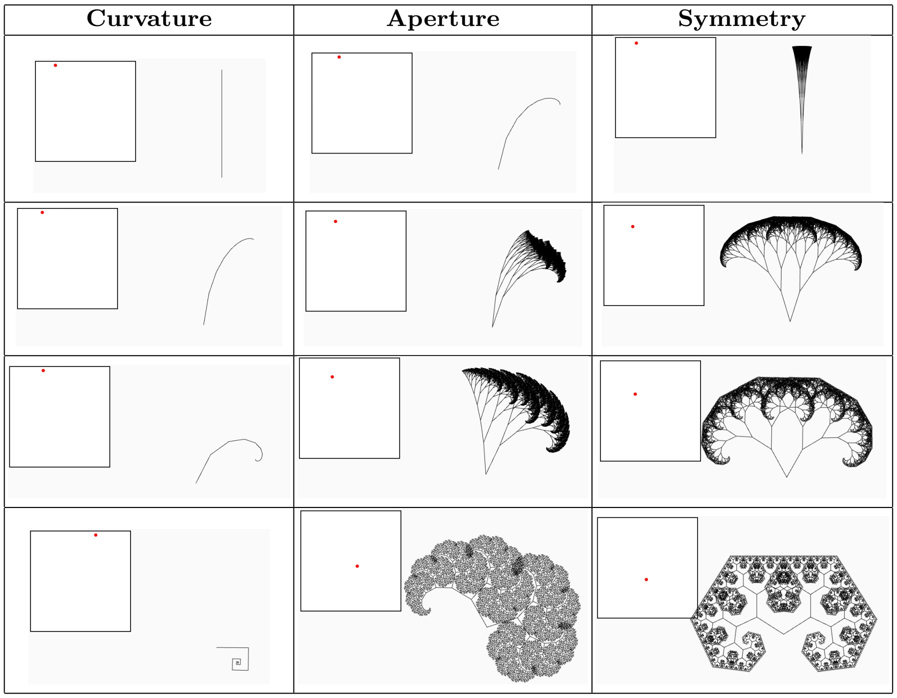


Fig. 2. Configurations according to different dimensions.

We observed the changes of the shape while moving the mouse cursor over the square. In doing that, we were inspired from a qualitative exploration described by Douglas Hofstadter in what he called an "exotic trip". He put his description in "*Gödel, Escher, Bach*" [5, pp.483-488] as a fictional dialogue and, three decades later, as a more detailed report [6, pp.65-69]. Hofstadter used a video camera pointed in various ways toward the output screen, and capable of generating several possible patterns. In particular, we made three main observations.

Shape Types Our first finding was that there are regions in the square corresponding to different types of shapes. As shown in Figure 1, some regions generate shapes recognizable as vegetable forms such as stone pines, firs, broccoli, or roots. Other regions generate polygons such as triangles, rectangles, or polygon spirals. Finally, there are regions associated to more complex shapes resembling snowflakes. Each region seems to correspond to specific *"natural concept"*, as defined by Gärdenfors [3].

Shape Dimensions The second observation is that, in each region, the shapes can be associated to a number of perceptual dimensions ascribable to Gärdenfors' *"quality dimensions"*. Specifically, we identified three dimensions: *curvature*, *aperture*, and *symmetry*. Each dimension seems to identify a specific trajectory in the conceptual space. Figure 2 shows some configurations according to the observed dimensions. Curvature and aperture can be easily defined in terms of the generative parameters. For example, since the overall figure is the superposition of a fixed number of broken lines, curvature can be defined as the angle formed by two adjacent segments in the broken line. According to the first column of Figure 2, the trajectory of curvature is a horizontal line. Moreover, aperture can be defined as the average difference between the curvature of two adjacent components. In the case of symmetry, the definition in terms of generative parameters seems more naturally definable "a posteriori", as a constraint on the generated shape.

Optimal Configurations Finally, the third observation is that, in each region associated to specific type of shapes, the aesthetic value of the shapes seems to change according to different generative parameters and dimensions. Furthermore, each column of Figure 2 shows that the aesthetic value seems to reach a maximum in correspondence of specific subsets of each region. These "optimal configurations" seem to be associated to specific ranges of curvature, aperture, and symmetry. At this stage of the research, this claim is proposed as an intuition to be formalized and empirically evaluated. In particular, it would be necessary to attempt a formal definition of aesthetic value in terms of the shape dimensions mentioned above. Moreover, an evaluation with human judges is needed to study to what extent there is agreement on the aesthetic values and their variation along the different shapes. Specifically, we intend to employ type of evaluation with subjects analogous to the one performed by Noy et al. [7]

3 Basin Jumping

If we consider a specific path in the square mapping the conceptual space, such that the variation of the aesthetic value is positive and reach its maximum in correspondence of the optimal configurations, we can view it from two different perspectives. On one hand, the path can describe a search session in the conceptual space of a creative system. On the other hand, it can be interpreted as a trajectory in the phase space of a dynamical system. According to the second interpretation, we can view each region of the conceptual space, associated to different shape types, as *basins of attraction* and their optimal configurations as the corresponding *attractors*. An attractor is a set of states (i.e., elements of the state space of a dynamical system) towards which a set of dynamical paths tend to evolve [9]. We go beyond the specific example described above and suppose that there is a large number of creative systems whose conceptual spaces can be decomposed in basins of attraction. Moreover, we hypothesize that the "creativity" of these system should not simply consist of the capability to generate the conceptual space and, starting from an initial configuration, explore its basin of attraction. Indeed, they should be capable of reaching basins of attraction not containing the past examples. In other words, if we assume the creativity as a search in the conceptual space, a higher degree of creativity is associated to the search of new basins of attraction.

4 Learning to Jump

The intuitions proposed in this work are aimed to identify a possible limitation in the use of CBR as a creative tool and to overcome it. A creative CBR system should get a the description of an artifact (i.e. an element of the conceptual space) as input case and retrieve one or more similar cases and reuse the corresponding knowledge to generate them. A possible intrinsic limitation is the use of similarity of past solutions. In terms of dynamical systems, we believe that this approach constraints the search inside a single basin of attraction. The suggestion emerged from the example described above is to identify perceptual dimensions and, through them, evaluation functions capable of reaching the maximum value in different basins of attractions.

In our next work, we aim to formalize, implement and empirically evaluate this approach. In particular, we intend to focus on generative systems analogous to the fractal tree generator and provide definitions of perceptual dimensions and aesthetic value. A crucial aspect is the combination of two types of heuristics, the first one for the discovery of new basin of attraction, and the second one for the identification of the optimal configuration.

Acknowledgments This research was supported by the EC project *WHIM: The What-If Machine*. See http://www.whim-project.eu.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications 7(1), 39–59 (1994)
2. Boden, M.A.: The Creative Mind. Abacus, London (1990)
3. Gärdenfors, P.: Conceptual spaces as a framework for knowledge representation. Mind and Matter 2(2), 9–27 (2004)
4. Giora, R.: On Our Mind: Salience, Context and Figurative Language. Oxford University Press, New York (2003)
5. Hofstadter, D.R.: Gödel, Escher, Bach: an Eternal Golden Braid. Basic Books, New York (1979)
6. Hofstadter, D.R.: I am a Strange Loop. Basic Books (2007)
7. Lior Noy, Y.H., Andrew, N., Ramote, O., Mayo, A., Alon, U.: Computers can't jump? A quantitative approach for studying creative leaps. In: International Conference on Computational Creativity (ICCC). pp. 72–76 (2012)
8. Mandelbrot, B.: The Fractal Geometry of Nature. W. H. Freeman, San Francisco (1982)
9. Milnor, J.W.: Attractor. Scholarpedia 1(11), 1815 (2006)
10. Reas, C., Fry, B.: Processing: A Programming Handbook for Visual Designers and Artists for Visual Designers and Artists. MIT Press (2007)
11. Ritchie, G.: Some empirical criteria for attributing creativity to a computer program. Minds and Machines 17(1), 76–99 (2007)
12. Wiggins, G.A.: Searching for computational creativity. New Generation Computing 24(3), 209–222 (2006)

Schematic processing as a framework for learning and creativity in CBR and CC

Kat Agres and Geraint A. Wiggins

Queen Mary University of London, London E1 4FZ, United Kingdom,
{kathleen.agres,geraint.wiggins}@qmul.ac.uk

Abstract. There is a clear connection to be made between psychological findings regarding learning and memory and the areas of case-based reasoning (CBR) and computational creativity (CC). This paper aims to encourage researchers in these areas to consider psychological perspectives while developing the technical and theoretical aspects of their computational systems. To this end, an overview of knowledge structures and schematic processing is provided, offering findings from music cognition to demonstrate the utility of this approach. Examples of musical schemata are offered as cases which may be used in CBR systems for combinatorial creativity and the generation of new creative output.

Keywords: cognitive psychology, schematic processing, computational creativity, case-based reasoning

1 Introduction

Creativity relies heavily upon domain-relevant experience and knowledge: an expert chess player's creative problem-solving, for example, is based on his robust knowledge and flexible thinking within his domain. Given the prime importance of past learning and experience for future creative behavior, there is an obvious marriage between the areas of case-based reasoning (CBR) and computational creativity (CC). While this connection has been explored in various computational settings, few approaches import findings and perspectives from cognitive psychology (although, see [10]), a field which may offer rich insight into this endeavour. Specifically, the mechanisms underlying learning and memory, and the way in which information is represented in the mind, should be considered, as these can elucidate creative behavior and inspire new ways of approaching machine creativity. In other words, artificial systems simulating human learning and memory can form the foundation for CBR approaches to CC.

This paper takes the stance that considering psychological mechanisms is essential not only for understanding human creativity, but for a theoretical understanding of creativity that can inform the implementation of creative processes in artificial systems. That is, researchers may be able to bolster CC by understanding how humans are creative. We focus on schematic processing mechanisms, such as the encoding and updating of memory representations, and the domain of music is considered as an example of how the abstraction of instances or cases yield schemata (e.g., generalized cases) which may be applied to CC.

2 Knowledge structures in human cognition

Cognitive psychology has thoroughly investigated learning and memory. Researchers once believed memory to be vast and detailed [28], but recent findings highlight its incompleteness and malleability. For example, vision research suggests that viewers primarily encode the general schematic attributes of a visual scene upon brief initial viewing [20, 26], supplying a semantic understanding of the scene [19, 20] but lacking detail. Similary, psychology and cognitive science have recently emphasized the importance of association and analogical processing [1,11]. Although veridical representations are sometimes encoded, more often we form general or associative semantic representations (*schemata*) of new input based on prior experience. This *schematic processing* is based on abstracted mental representations that structure or organize some aspect of past experience, and schematic memory structures influence the processing of new information.

Investigations of schematic processing have contributed to our theories of learning and memory for nearly a century [2, 24]. In *Remembering*, Bartlett notes that when individuals are asked to recall an odd or supernatural story after a time delay, their recollections alter the story to better conform to their existing schematic knowledge [2]. In other words, our knowledge shapes our perception and interpretation of the world. Piaget, who considered schemata to be the building blocks of knowledge, discussed how new information is incorporated into existing schemata in the processes of *assimilation* [24]. When the new information is too dissimilar to be integrated, *accommodation* occurs, in which the schematic structure itself must change to accommodate the new information.

The notion of schemata has been echoed in the fields of computer science and artificial intelligence for decades, for example, in Minsky's *frames* [17], and Schank's *script-based systems* [27]. Recent computational models learn and generalize the statistics of a training corpus (building what is essentially a statistical version of a schematic framework) in order to evaluate or categorize new instances [13,23]. This is akin to the process of assimilating new information into schematic representations, where the schemata in this instance are encoded in the network of probabilities underlying common structures or patterns. These statistical models have been used to generate new, creative output [22, 25]. CBR and CC approaches have successfully used techniques such as inductive analogical processes [21] and template-based methods (e.g., Gervas' ASPERA system [8]) for creative generation, but the connection to schema theory is often only implicit. Arguably, psychological findings should be explicitly applied here, because knowledge of how mental representations are formed and change over time (and are re-represented) can inform how AI systems may represent the information and knowledge required to achieve creative behaviors.

3 Music as an example domain

To show how psychology can inform how systems learn, represent, and combine information in new ways, we consider the domain of music. In the auditory modality, Bregman, Dowling, Cuddy, and others have explored the contribution of schema-based mechanisms to the abstraction of tonal relationships during music perception [4, 5]. Experience listening to common musical patterns or forms creates our mental framework for processing music [9,16]. The underlying schemata are essentially collections of rules that guide listeners' perception of music (and thence the information encoded) by directing attention and continually creating expectations about the forthcoming music [12,15,18,23]. Although musicians may have more elaborated schemata than non-musicians, everyone exposed to music has implicitly learned musical schemata. Conversely, every schema is modified by perceptual experience, as new information is abstracted and integrated into long-term schematic memory [29].

For concrete examples of musical schemata, we may consider Gjerdingen's examples of musical schemata: the "gap-fill" schema and the "changing-note" schema [9]. The former matches a melodic leap followed by an ascending or descending sequence of tones that fills the gap created by the interval leap. The latter matches two pairs of notes, in which the first pair leads away from the tonic pitch, and the second leads back. Even musically untrained listeners are capable of distilling these schemata from examples containing both types [9]. He further argues that musical schemata comprise a specific set of features that create a *style structure* [18]. Similarly, Snyder [29] describes musical schemata as networks of long-term memory associations that are amalgamations of the statistical properties of music: semantic frameworks constructed from "the commonalities shared by different experiences" [29]. Over time, episodic memories gradually form a generalized schematic representation in which specific details of each instance are lost, but generalizability of the schemata is gained.

In sum, musical schemata are mental frameworks of musical knowledge that are abstracted from experience and guide musical expectation. One insight from this work for CBR is to not simply match cases, but to *generalise* cases into schemata. If a CBR system has internalized schemata based on a corpus of musical cases (e.g., melodies), it is equipped to process new examples with more sophistication: by extracting schematic representations of these melodies, the representations may be more easily compared, and the generation of new music is made more feasible. Consider a system that generates novel, high-quality harmonization. First, it is provided with a case base of well-harmonized melodies from which it extracts schemata and derives characteristics of good harmonization. Then, given a new melody (case), it can generate harmony by matching within the space of schemata, to extrapolate a novel but appropriate harmony.

4 Knowledge structures as the foundation for creativity

Learning mechanisms and knowledge representations (such as schemata) are essential to how humans structure and combine information. They are also of central importance to CC, and the principle of combining existing knowledge into novel ideas has been a cornerstone of creativity research for decades [3, 6, 14]. Koestler describes creativity as *bisociation*—"interlocking of two previously unrelated skills, or matrices of thought" [14]. Inspired by Koestler, Fauconnier and Turner [6] offer a cognitive theory of conceptual blending, in which elements and relationships from different sources are combined to produce new meaning.

Several authors also refer to conceptual spaces which may be combined, manipulated, and traversed [3, 7, 30]. In all of these approaches, schemata could be used as general cases (or matrices or regions of conceptual spaces) that may be combined to form new, creative ideas. Further, schemata may be viewed as methods for caching or even hashing the case base, thus improving retrieval efficiency.

Knowledge of psychological processes can inform how learning and memory may be instantiated in artificial systems, which in turn influences how concepts may be blended and combined. One may consider schemata to be the building blocks for exploratory and combinatorial creativity. If a CBR system maps melodic onto schematic representations, the system may then be used to classify or even generate new examples through extrapolation (or interpolation) of existing cases. This approach is especially useful for CC, because a means of reflection or self-evaluation should be built into the system, and CBR can satisfy this need. Further, the way in which humans learn and encode information can suggest particular schemata that may contribute to CC in AI systems, but also (and just as importantly), elucidate the *processes* underlying the combination of knowledge structures [30]. For example, one could use a schema-based system to judge whether new melodies will sound novel to listeners by examining whether different melodies abstract to the same schemata, and this could be very useful for applications such as automatic composition.

5 Conclusion

We argued for the consideration and inclusion of psychological findings in CBR as a means of approaching CC. Using examples of mental knowledge structures and schematic processing mechanisms in the musical domain, we discussed how existing schemata may be considered as cases for the combination of ideas and generation of new creative output. Understanding how humans learn and form memory representations may inform machine learning and CBR techniques, and ultimately, the expression of creativity in artificial systems.

Acknowledgements

The Lrn2Cre8 project acknowledges 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 610859.

References

1. Bar, M.: The proactive brain: using analogies and associations to generate predictions. Trends in cognitive sciences 11(7), 280–289 (2007)
2. Bartlett, F.C.: Remembering: A study in experimental and social psychology, vol. 14. Cambridge University Press (1995)
3. Boden, M.A.: The creative mind: Myths and mechanisms. Psychology Press (2004)
4. Bregman, A.S.: Auditory scene analysis: The perceptual organization of sound. MIT press (1994)

5. Dowling, W.J.: Scale and contour: Two components of a theory of memory for melodies. Psychological review 85(4), 341 (1978)
6. Fauconnier, G., Turner, M.: The way we think: Conceptual blending and the mind's hidden complexities. Basic Books (2008)
7. Gärdenfors, P.: Conceptual spaces: The geometry of thought. MIT press (2004)
8. Gervás, P.: An expert system for the composition of formal spanish poetry. Knowledge-Based Systems 14(3), 181–188 (2001)
9. Gjerdingen, R.O.: A classic turn of phrase: Music and the psychology of convention. Univ of Pennsylvania Press (1988)
10. Heath, D., Dennis, A., Ventura, D.: Imagining imagination: A computational framework using associative memory models and vector space models. In: Proc. of the 6th International Conference on Computational Creativity. p. 244 (2015)
11. Hofstadter, D.R.: Analogy as the core of cognition. Keith J. Holyoak, and Boicho N. Kokinov. Cambridge MA: The MIT Press/Bradford Book (2001)
12. Huron, D.B.: Sweet anticipation: Music and the psychology of expectation. MIT press (2006)
13. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: Proceedings of the International Conference on Learning Representations (ICLR), (2014)
14. Koestler, A.: The act of creation. Macmillan (1964)
15. Krumhansl, C.L.: Cognitive foundations of musical pitch, vol. 17. Oxford University Press New York (1990)
16. Lerdahl, F.: Tonal pitch space. Oxford University Press (2001)
17. Minsky, M.: A framework for representing knowledge. Tech. rep., Cambridge, MA, USA (1974)
18. Narmour, E.: The analysis and cognition of melodic complexity: The implication-realization model. University of Chicago Press (1992)
19. Oliva, A.: Gist of the scene. Neurobiology of attention 696(64), 251–258 (2005)
20. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. Progress in brain research 155, 23–36 (2006)
21. Ontañón, S., Plaza, E.: Amalgams: A formal approach for combining multiple case solutions. In: Case-Based Reasoning. Research and Development, pp. 257–271. Springer (2010)
22. Pearce, M.T., Wiggins, G.A.: Evaluating cognitive models of musical composition. In: Cardoso, A., Wiggins, G.A. (eds.) Proc. of the 4th International Joint Workshop on Computational Creativity. pp. 73–80. London (2007)
23. Pearce, M.T., Wiggins, G.A.: Auditory expectation: the information dynamics of music perception and cognition. Topics in cognitive science 4(4), 625–652 (2012)
24. Piaget, J., Cook, M., Norton, W.: The origins of intelligence in children, vol. 8. International Universities Press New York (1952)
25. Ponsford, D., Wiggins, G.A., Mellish, C.: Statistical learning of harmonic movement. Journal of New Music Research 28(2), 150–177 (1999), http://www.soi.city.ac.uk/ geraint/papers/JNMR97.pdf
26. Rensink, R.A., O'Regan, J.K., Clark, J.J.: To see or not to see: The need for attention to perceive changes in scenes. Psychological science 8(5), 368–373 (1997)
27. Schank, R.C., Abelson, R.P.: Scripts, plans, goals, and understanding: An inquiry into human knowledge structures. Psychology Press (2013)
28. Shepard, R.N.: Recognition memory for words, sentences, and pictures. Journal of verbal Learning and verbal Behavior 6(1), 156–163 (1967)
29. Snyder, B.: Music and memory: An introduction. MIT press (2000)
30. Wiggins, G.A.: A preliminary framework for description, analysis and comparison of creative systems. Knowledge-Based Systems 19(7), 449–458 (2006)

Generation of concept-representative symbols
Position paper

João Miguel Cunha, Pedro Martins, Amílcar Cardoso, Penousal Machado

Departamento de Engenharia Informtica da Universidade de Coimbra
{jmacunha,pjmm,amilcar,machado}@dei.uc.pt

Abstract. The visual representation of concepts or ideas through the use of simple shapes has always been explored in the history of Humanity, and it is believed to be the origin of writing. We focus on computational generation of visual symbols to represent concepts. We aim to develop a system that uses background knowledge about the world to find connections among concepts, with the goal of generating symbols for a given concept. We are also interested in exploring the system as an approach to visual dissociation and visual conceptual blending. This has a great potential in the area of Graphic Design as a tool to both stimulate creativity and aid in brainstorming in projects such as logo, pictogram or signage design.

Keywords: Computational creativity, Computational generation, Concept representation, Visual representation

1 Introduction

Creativity can be seen as the ability to create novel ideas by making connections between existing ones. It plays an important role in the area of Graphic Design not only in conceiving new concepts but also in visually representing them.

As far as visual representation of concepts is concerned, humans have been doing it since more than two hundred thousand years ago – take for example cave paintings. These representations vary from being completely pictorial – e.g. pictograms – to more abstract – e.g. ideographs.

The link between the visual representation and the conceptual connections behind it can in fact be observed. Examples of this can be seen by looking at Chinese characters, more specifically at the ones categorised as Ideogrammic Compounds (see Figure 1). These characters can be decomposed into others, whose concepts are semantically related, belonging to the same (or at least similar) conceptual space [6].

Some authors were inspired by this relationship between concepts to their visual representations. One of them was Charles Bliss who developed a communication system composed of several hundreds of ideographs that can be combined to make new ones – *Blissymbols* [1]. In his system the variation in terms of abstraction degree can also be observed (see Figure 2).

本 木 林 森

Fig. 1. Chinese characters for *root*, *tree*, *woods* and *forest* (left to right). *Root* can be obtained by adding a line to the *tree* character; *woods* character can be obtained (barely) by using two *tree* characters; *woods* can be obtained by using three *tree* characters.

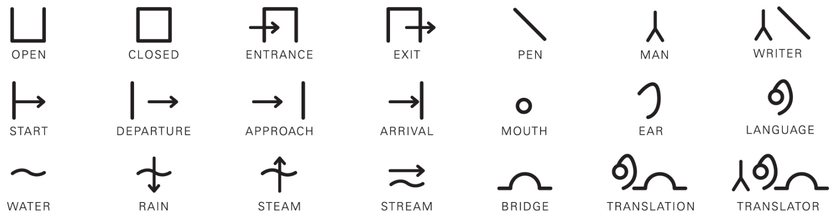


Fig. 2. Blissymbols. Several interesting things can be observed by looking at blissymbols: such as a variation in terms of abstraction degree (there are both pictorial and abstract symbols); by combining symbols, new meanings are obtained (examples in Figure 2: *pen* + *man* = *writer*, *mouth* + *ear* = *language*); by using the same symbols in a different position, a new meaning is obtained (see symbols *water*/*rain*/*steam*/*stream*).

Inspired by examples as the ones presented above, our goal is to conceive an approach for computationally generating concept-representative symbols, i.e. visual representations of concepts. In this paper we present some of the key aspects that have to be considered when generating such symbols and the strategies to explore in order to achieve our objective.

2 Generation of concept-representative symbols

The idea of creating a symbol for a given concept based on its connections to other concepts is, just by itself, interesting. However, if we consider the exploration of this idea using computational means to automate the generation process of the symbols, its potential greatly increases.

We can think of a tool capable of generating symbols whose visual properties would be the outcome of an analysis of the conceptual space of the introduced concept. We believe that such a tool could assist the designer during the ideation process by stimulating its creativity, aiding in brainstorming activities and thus giving rise to new ideas and concepts.

Concerning the visual qualities of the generated symbols, it is crucial to consider several aspects. The first one is the degree of abstraction. This aspect can be considered to be influenced by the choice of the connections used in the symbol generation. Take for example the concept *car*: if we consider the connections between *car* and the concepts *door*, *window* and *wheel*, the resultant symbol will probably be highly pictorial; if we choose to ignore those connections, the resulting symbol might be more abstract. Ideally, the tool should allow the abstraction degree to be set according to the user's needs.

However, this is not the only aspect that is greatly dependent on the connections used. As observed in the blissymbols, a given combination of symbols might lead to different interpretations. If some perceptual aspects are not considered, this might result in a conflict between the concept and the perception of the symbol. In the next subsection some of these aspects will be presented.

2.1 Considering visual characteristics

When dealing with symbols, it is important to bear in mind some aspects of how a representation's meaning can be changed by changing some of its visual characteristics. The following aspects are essential: position, colour and shape.

The first semiotic aspect – position – can be seen in Figure 2. By putting an arrow next to the *water* symbol a new concept is represented. The concept also varies according to positioning of the arrow. Such details must be considered and a mechanism for analysing them needs to be developed (similar issues have been considered in [2]).

Fig. 3. Left side is shown how the meaning of a banana can change with its colour (mature, green and red banana). Right side is Kiki/Bouba example. Accordingly to Ramachandran 98% of all respondents atribute the name Kiki to the shape on the left and Bouba to the right one, despite having no meaning at all [9]. *Best viewed in colour.*

Another aspect to be regarded is colour. Through a brief analysis of the banana example in Figure 3 it is easy to understand the importance of this aspect. By simply attributing a different colour to the same symbol, its perceptual meaning also changes. In addition, the use of colour has already been proven as a mean of facilitating the interpretation of visual representations of concepts (e.g. [7]). However, its incorrect use has the opposite effect (e.g. Stroop effect), causing interference in its interpretation. On the other hand, a mechanism to avoid an over-use of colour will probably be needed as colour might not be necessary in some symbols.

The third important aspect is shape. When generating visual symbols from textual data (e.g. semantic networks), one cannot avoid dealing with shape. The choice of shape for a given concept is not easy by itself but one has also to consider its visual qualities. Take for example the shapes presented in Kiki/bouba example in Figure 3. Despite not having any meaning at all, there is a clear tendency or bias when attributing names to them. This can be explained as follows, humans tend to perform mappings among domains, namely between image and sound, as such sharp shapes tend to be associated with sharp sounds and organic shapes with smooth ones [9].

As we have already mentioned, these semiotic aspects have to be considered when generating symbols. This is only possible to achieve by thoroughly analysing the conceptual network and also considering previously generated symbols as both examples and base for the generation of the new symbol.

2.2 Getting information

An extensive analysis of the conceptual space is important but there is another issue that has to be resolved: if the system does not have access to a large source of knowledge – with information about visual properties – it will be difficult, if not impossible, to achieve good results. One possibility is to use an already built semantic network of common sense knowledge (e.g.[8] or [3]).

However, as our main objective is to generate visual representations, knowledge about visual characteristics is required. For that reason, a methodology has to be conceived for acquiring such data. A possible solution is to use a similar approach to the one used by Open Mind Common Sense project – a knowledge acquisition system designed to acquire common sense knowledge from the general public over the web [10]. Our goal is to focus on gathering information about objects' visual characteristics such as colour, shape and texture. These will likely allow us to attain adequate results in terms of symbol generation.

Crowdsourcing will probably be used in our knowledge-gathering process as it easily allows to reach a high number of contributors at a reduced cost. In addition, the validity of online crowdsourced experiments on visual properties and graphical perception has already been demonstrated (e.g. [5]).

This *distributed human project* approach allows us not only to gather data at a scale that would not be possible otherwise but also enables us to study the role of context in perception – one of our goals is to test whether the symbols generated differ accordingly to the location where the data was gathered from.

We also intend to explore other alternatives for populating our semantic network, such as automatic gathering of information. Using Google Image Search is one example of this and can be used to find images related to the content being analysed and consequently extracting useful information from them (e.g. [7]).

2.3 Generating symbols

In our opinion there are, at least, two different ways of generating a symbol for a given concept: (1) starting with no prior knowledge and analysing the conceptual space in order to extract possible visual features to be used in the symbol; (2) using prior knowledge – previously generated symbols of concepts that are, in some way, related to the introduced concept. This would lead to a higher coherence among generated symbols. In both cases, not only direct conceptual connections are used but also more uncommon ones – through a process of analogy. As such, we argue that mechanisms such as case-based reasoning or conceptual blending [4] are suitable strategies to generate symbols of concepts. As for the former, we can consider a case-base comprised of symbols such as the ones depicted in Figure 2 and develop a system to produce novel ones using analogues to the previous ones. Regarding conceptual blending, the idea is to explore the structure mapping approach to analogy and concept integration based on conceptual spaces and semiotic systems.

3 Conclusion

In this paper we presented our approach to computational generation of concept-representative symbols. We aim to develop a design aiding tool that combines the exploration of conceptual spaces in combination with processes of analogy-making and semiotic analysis to generate possible visual (abstract/semi-abstract) representations for the concepts introduced by the user. We believe that it will help the designer during the ideation process by stimulating its creativity, aiding in brainstorming activities and thus giving rise to new ideas and concepts.

Acknowledgements

The authors acknowledge the financial support from the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under the ConCreTe FET-Open project (grant number 611733) and the PROSECCO FET-Proactive project (grant number 600653).

References

1. Bliss, C.K.: Semantography (Blissymbolics): A Logical Writing for an illogical World. Semantography Blissymbolics Publ (1965)
2. Confalonieri, R., et al.: Using argumentation to evaluate concept blends in combinatorial creativity paper type: Study paper. (to Appear in ICCC 2015)
3. De Smedt, T.: Modeling Creativity: Case Studies in Python. University Press Antwerp (2013)
4. Fauconnier, G., Turner, M.: Conceptual integration networks. Cognitive science 22(2), 133–187 (1998)
5. Heer, J., Bostock, M.: Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 203–212. ACM (2010)
6. Hew, S., et al.: Using combining evolution of pictogram chinese characters to represent ideogrammic compounds chinese characters. In: Computing and Convergence Technology, 2012 7th International Conference on. pp. 219–223. IEEE (2012)
7. Lin, S., et al.: Selecting semantically-resonant colors for data visualization. In: Computer Graphics Forum. vol. 32, pp. 401–410. Wiley Online Library (2013)
8. Liu, H., Singh, P.: Conceptneta practical commonsense reasoning tool-kit. BT technology journal 22(4), 211–226 (2004)
9. Ramachandran, V.S., Hubbard, E.M.: Synaesthesia–a window into perception, thought and language. Journal of consciousness studies 8(12), 3–34 (2001)
10. Singh, P., et al.: Open mind common sense: Knowledge acquisition from the general public. In: On the move to meaningful internet systems 2002: Coopis, doa, and odbase, pp. 1223–1237. Springer (2002)