Tarek R. Besold
Oliver Kutz
Carlos Leon (eds.)

Computational Creativity, Concept Invention, and General Intelligence

5th International Workshop, C3GI @ ESSLLI 2016,
Bozen-Bolzano, Italy, August 20/21, 2016

Pre-Proceedings

(Version 1.0 – August 17, 2016)

Volume Editors

Tarek R. Besold
The KRDB Research Centre
Faculty of Computer Science
Free University of Bozen-Bolzano

Oliver Kutz
The KRDB Research Centre
Faculty of Computer Science
Free University of Bozen-Bolzano

Carlos Leon
Departamento de Ingenieria del Software e Inteligencia Artificial
Facultad de Informatica
Universidad Complutense de Madrid

This volume contains the pre-proceedings of the workshop “Computational Creativity, Concept Invention, and General Intelligence (C3GI)” held in conjunction with ESSLLI 2016.

Program Committee

Committee Co-Chairs

- Tarek R. Besold, Free University of Bozen-Bolzano
- Oliver Kutz, Free University of Bozen-Bolzano
- Carlos Leon, Universidad Complutense de Madrid

Committee Members

- Mohammad Majid Al-Rifaie, Goldsmiths, University of London
- Liane Gabora, University of British Columbia – Okanagan
- Pablo Gervas, Complutense University of Madrid
- Hugo Goncalo Oliveira, University of Coimbra
- Kazjon Grace, University of North Carolina – Charlotte
- Bipin Indurkhya, AHG University of Science and Technology – Krakow
- Anna Jordanous, University of Kent
- Maximos Kaliakatsos-Papakostas, Aristotle University of Thessaloniki
- Maria Teresa Llano Rodriguez, Goldsmiths, University of London
- Ramon Lopez De Mantaras, IIIA-CSIC, Barcelona
- Stephen McGregor, Queen Mary, University of London
- Alison Pease, University of Dundee
- Enric Plaza, IIIA-CSIC, Barcelona
- Hannu Toivonen, University of Helsinki
- Mark Turner, Case Western Reserve University
- Lav Varshney, University of Illinois at Urbana-Champaign
- Tony Veale, University College Dublin
- Dan Ventura, Brigham Young University
- Pei Wang, Temple University Philadelphia
- Geraint Wiggins, Queen Mary University of London
- Frank van der Velde, University of Twente

Table of Contents

Modeling metaphor perception with distributional semantics vector space models

Kat R. Agres,[1] Stephen McGregor,[1] Karolina Rataj,[2,3]
Matthew Purver,[1] Geraint A. Wiggins[1]

[1] Queen Mary University of London, London, UK
{kathleen.agres,s.e.mcgregor,m.purver,geraint.wiggins}@qmul.ac.uk
[2] Department of Cognitive Psychology and Ergonomics,
University of Twente, Enschede, the Netherlands
[3] Faculty of English, Adam Mickiewicz University, Poznań, Poland
krataj@wa.amu.edu.pl

Abstract. In this paper, we present a novel application of a computational model of word meaning to capture human judgments of the linguistic properties of *metaphoricity*, *familiarity*, and *meaningfulness*. We present data gathered from human subjects regarding their ratings of these properties over a set of word pairs specifically designed to exhibit varying degrees of metaphoricity. We then investigate whether these properties can be measured in terms of geometric features of a model of distributional lexical semantics. We compare the performance of two models, our own Concept Discovery Model which dynamically constructs context-sensitive subspaces, and a state-of-the-art static distributional semantic model, and find that our dynamic model performs significantly better in its measurement of metaphoricity.

Keywords: metaphor, distributional semantics, vector space models, computational creativity

1 Introduction

In this study, we investigate whether computational models of lexical meaning might help explain human comprehension of metaphors. We examine several alternative models, based on language statistics in a large general collection of English, to see if they capture relations between words which correlate with human judgments of metaphoricity.

Psycholinguistic studies Research on metaphor in human participants has attempted to clarify the mechanisms underlying understanding metaphoric language. One of the earliest approaches, the standard pragmatic model, stipulates that the literal meaning of a metaphoric sentence needs to be rejected before the figurative meaning is generated [10]. Behavioral studies inspired by this model have shown that participants do not use more time to comprehend metaphoric than literal sentences, and that metaphoric meaning is generated in parallel

with the literal meaning of an utterance [8, 29]. These studies, however, did not make a distinction between conventional and novel metaphors. Later reports have shown that novel metaphors do require more processing time than literal sentences, while conventional metaphor and literal language comprehension take comparable time [2].

Recently, a number of electrophysiological (EEG) studies investigating metaphor comprehension have been reported in which the N400 component, a negative-going wave observed between 300 and 500ms after the presentation of the critical stimulus, has received considerable attention. Larger N400 amplitudes have been observed in the processing of metaphoric as compared to literal sentences, with no differences in component latency (i.e., the time window within which the effect is observed) or scalp distribution (i.e., the sites on the scalp over which the effect is present) (e.g., [2]). This increase in amplitude has been interpreted as reflecting more activity in memory needed to retrieve the semantic information necessary for comprehension of metaphoric as compared to literal sentences[16]. At the same time, comparable latency and scalp distribution of the component might be indicative of the involvement of similar mechanisms in literal and metaphoric language comprehension. Interestingly, differences have been found between conventional and novel metaphors, with the N400 amplitudes for conventional metaphors falling in-between those for novel metaphoric and literal utterances [2]. This graded effect has not been observed in reaction time studies, which demonstrates that ERP measures offer greater sensitivity to the time course of cognitive processes involved in metaphor comprehension. These findings raise important questions concerning the nature of the mechanisms involved in understanding metaphors.

One of the approaches that has attempted to elucidate metaphor comprehension mechanisms is the structure mapping model and its descendant, the career of metaphor model, which stipulate that the same mechanisms are involved in the comprehension of literal comparisons, similes, and metaphors [5, 30]. Within this view, comprehending metaphoric sentences, like *my mind is a warehouse*, requires a mapping that involves a symmetrical mechanism of alignment of relational commonalities in the source (*warehouse*) and target (*mind*), together with an asymmetrical mechanism of inference projection from the source to the target. Moreover, the career of metaphor model assumes that while novel metaphors are understood via comparison, categorization is involved in conventional metaphor understanding. These assumptions have received some support in ERP studies, which have also shown that a shared mapping process may be involved in categorization and comparison [9, 17]. Moreover, comparison seems to facilitate not only novel, but also conventional metaphor comprehension, although this facilitation is observed at later processing stages than in the case of novel metaphors.

Computational studies Computational approaches to metaphor have generally focused on a combination of pattern matching and hand coded information processing [25]. The KARMA model for metaphor understanding [7], for instance, attempts to encode environmentally grounded knowledge about action in the world into a framework of transferable domains. In a similar vein, ATT-Meta

[3] asks users to provide domain specific knowledge about entities and processes and then ports this knowledge between different contexts—this context sensitive aspect of the system in particular aligns with both the approach to theoretical work on metaphor and to the computational work presented here.

Moving towards data-driven approaches, a model for metaphor comprehension has been described that employs latent semantic analysis, a statistical technique for building spaces of word similarity, to the selection of the salient features trafficked between a metaphoric source and target [15]. This technique, along with a similar method involving the selection of transferred features based on proximity in a semantic space [27], bears comparison to our dynamically contextual method as described in Section 3.2: each of these models attempts to extract particular features of a semantic space in order to capture the semantic context of a metaphor. A more recent description of a Service-Oriented Architecture [28] discovers properties of source and target by matching patterns within large-scale web corpora and then looks for properties salient in the source which can be transferred to the target. Other contemporary approaches have tended towards the more overtly statistical, with for instance the application of linear algebraic operations to model the metaphoric composition of vector space type word representations [11].

The computational component of the work presented here broadly falls within the paradigm of the *distributional hypothesis*, which holds that "words that occur in similar contexts have similar meanings," [26, p. 148]. The general methodology of distributional semantics involves the traversal of large scale textual corpora in order to build spaces of word-vectors where the proximity of two vectors reflects the tendency for those two words to be observed co-occurring with similar terms [6]. Initial approaches to distributional semantics typically involved building up word representations based on straight-up co-occurrence counts [24], while more recent methodologies have often incorporated matrix factorisation techniques to derive dense matrices from co-occurrence statistics [13] or employed neural network architectures to derive *word embeddings* from observations of co-occurrences across iterative traversals of a corpus [4].

In the study presented here, we will in particular be comparing Word2Vec [21], a neural network driven model for generating word embeddings that has achieved state-of-the-art results on tests of word similarity and analogy completion, with our own Concept Discovery Model [19], which deploys a word-counting approach to distributional semantics to dynamically construct contextualised subspaces in which conceptual relationships play out as geometric relationships [20]. We will be examining the ways in which spaces generated by each model compare with human assessments of the degree of *metaphoricity*, *familiarity*, and *meaningfulness* in noun-verb word pairings. With this in mind, recent work using distributional models enriched with information from lexical and associative knowledge bases to build spaces of word-vectors constructed for detecting similarity or relatedness should be taken into consideration [14].

2 Modeling human metaphor judgements

Our objective in this study is to explore the ways in which geometric models of word meaning can capture the perception of metaphor, and in particular can measure the degree to which two-word phrases are perceived as being metaphorical. To do so, we compare words' relations in the geometric model with human judgments of metaphoricity, via a set of empirically-derived normative data. Note that while data were collected for three types of norming measures (metaphoricity, meaningfulness, and familiarity), the principal aim of the computational work is to model the perception of metaphoricity, that is, to discover meaningful subspaces that reflect the extent to which a two-term expression is perceived as being metaphorical.

2.1 Materials

The materials were collected for an ERP study, which investigated metaphor comprehension in bilinguals [12]. Verb-noun word dyads in Polish (native language) and English (second language) were used in the ERP experiment. In each case, the verb was considered the metaphoric source and the noun the target: so, for example, in the instance of the conventional metaphor "cut pollution", some salient property of the action CUTTING is being transferred to the entity POLLUTION.

Prior to the ERP experiment, five normative studies were carried out to ensure the word pairs fell within the following three categories: novel metaphors (e.g., *to harvest courage*), conventional metaphors (e.g., *to gather courage*), and literal expressions (e.g., *to experience courage*). Based on the results of the normative studies, the final set of 228 English verb-noun word dyads (76 in each category) was selected for the purpose of the current study. Out of the five normative studies, four will be reported here. The statistical analyses consisted of mixed-design analyses of variance (ANOVAs), with utterance type as a within-subject factor and survey block as a between-subject factor. No main effect of survey block was observed. Significance values for the pairwise comparisons were corrected for multiple comparisons using the Bonferroni correction. When Mauchlys tests showed that the assumption of sphericity was violated, the Greenhouse-Geisser correction was applied. In such cases, the original degrees of freedom are reported with the corrected p value. The demographic data for the participants of the four normative studies are presented in Table 1.

Table 1. Demographic characteristics of participants of the four normative studies, including the number of participants (number of female participants) and mean age.

Normative study type	**Number of participants(female)**	**Mean age**
Cloze probability	140 (65)	23
Meaningfulness ratings	133 (61)	22
Familiarity ratings	101 (55)	23
Metaphoricity ratings	102 (59)	22

Cloze probability Because reduced N400 amplitudes have been observed in relation to expected as compared to unexpected words, a cloze probability test was performed prior to the ERP study to ensure the second word in a given word dyad was not highly anticipated by the participants of the ERP experiment. Each participant of the cloze probability test received the first word of a given word pair, and was asked to provide the second word, so that the two words would make a meanigful expression. Due to the length of the test, all word pairs were divided into four blocks, so that each word was completed by 35 participants. If a given word pair was observed in the cloze probability test more than 3 times, the word pair was excluded from the final set and replaced with a new one. This procedure was repeated until the cloze probability for word pairs in all categories did not exceed 8%.

Meaningfulness In order to assess the meaningfulness of the stimuli, participants were asked to rate how meaningful a given word pair was on a scale from 1 (totally meaningless) to 7 (totally meaningful). The set of 228 word dyads was divided into four survey blocks in order to avoid the repetition of the target word within the same survey. Additionally, 76 meaningless word pairs were included in this normative study. The results revealed a main effect of utterance type, $[F(3, 387) = 1611.54, p < .001, \epsilon = .799, \eta_p^2 = .93]$. Pairwise comparisons revealed that literal word pairs were assessed as more meaningful ($M = 5.99, SE = .05$) than conventional metaphors ($M = 5.17, SE = .06$) ($p < .001$), and conventional metaphors were assessed as more meaningful than novel metaphors ($M = 4.09, SE = .08$)($p < .001$).

Familiarity Familiarity of each word pair was assessed in another normative study. Participants were asked to decide how often they had encountered the presented word pairs on a scale from 1 (very rarely) to 7 (very frequently). The set of 228 word dyads was divided into three survey blocks in order to avoid the repetition of the target word within the same survey. Again, a main effect of utterance type was found, $[F(2, 296) = 470.97, p < .001, \epsilon = .801, \eta_p^2 = .83]$. Pairwise comparisons showed that novel metaphors ($M = 2.15, SE = .07$) were rated as less familiar than conventional metaphors ($M = 2.97, SE = .08$), ($p < .001$), with literal expressions being most familiar ($M = 3.85, SE = .09$), ($p < .001$). Furthermore, conventional metaphors were less familiar than literal word dyads, ($p < .001$). It is crucial to note that although differences were observed between categories, all word pairs were relatively unfamiliar. This is visible in the mean score for literal word pairs, which are most familiar of all three categories, but at the same time relatively low in familiarity (below 4 on a scale where 6 and 7 represent very familiar items). The reason why familiarity was low in all three categories is the same as for the cloze probability test, i.e., that we intentionally excluded highly probable combinations.

Metaphoricity In order to assess the metaphoricity of the word pairs, participants were asked to decide how metaphoric a given word dyad was on a scale

from 1 (very literal) to 7 (very metaphoric). The set of 228 word dyads was again divided into three survey blocks in order to avoid the repetition of the target word within the same survey. The results revealed a main effect of utterance type, $[F(2, 198) = 588.82, p < .001, \epsilon = .738, \eta_p^2 = .86]$. Pairwise comparisons confirmed that novel metaphors (M = 5.00, SE = .06) were rated as more metaphoric than conventional metaphors (M = 3.98, SE = .06), ($p < .001$), and conventional metaphors were rated as more metaphoric than literal utterances $(M = 2.74, SE = .07), (p < .001)$.

3 Computational Modeling Method

In order to computationally model human judgment of the conceptual features of word dyads, we construct distributional semantic spaces where the proximity of word-vectors relates to their semantic similarity, and then explore the geometry of these spaces for ways of mapping relationships between words that are productive with regard to such conceptual, cognitive phenomena as metaphor. Specifically, we compare two different distributional semantic models to assess the difference in performance between a model that might be described as *static*, such as the one outlined in Section 3.1, versus one that is contextually *dynamic*, as is the intent with our own model as explained in Section 3.2.

For both our static and dynamic models, we train vectors on the English language version of Wikipedia. For the purpose of capturing word co-occurrences, we focus only on the descriptive content of Wikipedia pages, ignoring headers, lists, captions, and the like. Considering only sentences at least five words in length, we strip the corpus of punctuation, remove articles (*the*, *a*, and *and*), and remove parenthetical phrases, resulting in an overall corpus of approximately 7.5 million word types and 1.1 billion word tokens. For the construction of both models, we consider context windows of five words on either side of a target word, treating sentence endings as contextual boundaries as well. We take the 200,000 most frequently occurring words in the corpus as the vocabulary for both models, constructing one word-vector for each word in the vocabulary.

As our measure of semantic relatedness between two words, we take the cosine similarity between their corresponding word-vectors, in line with a number of other contemporary distributional semantic models [18, 23]. It should be noted that in the case of a normalised distributional semantic space, such as that described in Section 3.1, relations based on cosine similarity are equivalent to those based on Euclidean distance.

3.1 Word2Vec

As our primary point of comparison in this study, we use the Word2Vec distributional semantic model [22]. This model has achieved state of the art results on analogy completion tasks in particular, and has generally received widespread attention within the field of computational linguistics. A critical feature of the model is its deployment of a neural network to build a space of word-vectors. One result of this process is that the model's dimensionality cannot be interpreted:

Word2Vec treats a dimension as an arbitrary handle for pulling word-vectors into the desired relationship based on observations of co-occurrences in training data. Therefore, in comparison to our model described in Section 3.2, it is not possible to project dimensionally contextualized subspaces from a Word2Vec type model in a direct manner (while perhaps a separate neural network could be designed and trained specifically to perform this projection, this is beyond the scope of this paper).

Two different network architectures have been reported in the literature; here, we employ the *Skip-gram* architecture, consisting of a two-layer neural network which learns to predict context terms based on an input word, as this approach has been reported as performing particularly well on semantically oriented tasks [21]. The model takes the form of a set of word-vectors arrayed across the surface of a hypersphere. Here, we build a 300-dimensional space based on 10 passes over the corpus described above, with a negative sampling rate of 10. To assess the model's ability to capture the human metaphoricity judgment data, we then measure the cosine similarity between the word-vectors for each word in each word pair from the study described in Section 2.

3.2 Conceptual Discovery Model

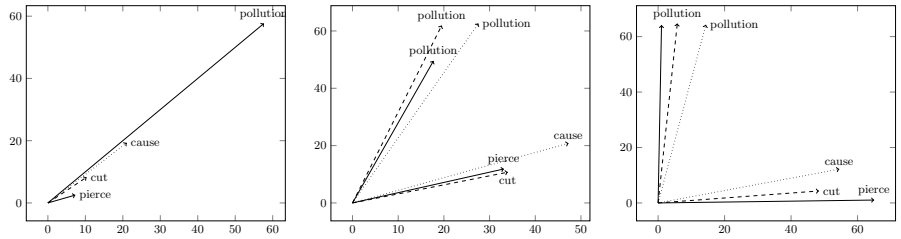
We compare the performance of the established Word2Vec semantic model with the output of a distributional semantic model which dynamically interprets input text to project context-sensitive subspaces from a sparse, high-dimensional base space. As described in detail elsewhere [1, 19], the Conceptual Discovery Model builds a base space populated by what might be described as literal statistical data about word co-occurrences as observed in a large-scale corpus: each dimension in the space corresponds directly to a co-occurrence term, and no matrix factorisation or other dimensional reduction technique is applied to this base space. Rather, each dimension c of a word-vector $\overrightarrow{w}$ is populated with a pointwise mutual information (PMI) score based on this equation, where $n_{w,c}$ represents the frequency at which word w is observed occurring within 5 words of word c, n_w is the independent frequency of w, n_c is the independent frequency of c, W is the total word count, and a is a smoothing constant:

$$\overrightarrow{w_c} = \log_2 \left(\frac{n_{w,c} \times W}{n_w \times (n_c + a)} + 1 \right) \tag{1}$$

We build a base space of roughly 7.5 million dimensions, corresponding to the number of word types in Wikipedia. From this base space, we dynamically pick 200-dimensional subspaces, specific to each word-pair in the study. Three different methodologies for projecting subspaces will be discussed below, but in each case, the input used to determine the projection is simply the pair of words involved in a potentially metaphoric dyad, and the projection is based on an analysis of the respective values of these inputs along any given dimension. The intuition behind this methodology is that subspaces consisting of dimensions which are mutually salient for both components of a dyad will capture something of the semantic context in which the candidate metaphor might be meaningful.

Within these subspaces, as with Word2Vec, we assess the relationship between the words in a word pair in terms of the cosine similarity between their two corresponding word-vectors. One of the primary considerations in the application of this model is therefore the method for selecting these subspaces.

Discovering conceptually relevant spaces We experimented with three different techniques for choosing subspaces from our base space, in each case focusing on the relationship between the target and source word in each dyad.



(a) Noun-Only Subspaces (b) Joint Subspaces (c) Independent Subspaces

Fig. 1. Here three different types of subspaces are presented, with three expressions involving the word "pollution" superimposed on each two-dimensional projection: the literal phrase "cause pollution", the conventional metaphor "cut pollution", and the novel metaphor "pierce pollution". Angles between the vectors for both words in each pair are measured for correlation with human judgments of the metaphoricity of each expression. Angles and vector lengths from the 200 dimensional subspaces we analysed are preserved in these projections. The word-vector for pollution is the same for all three versions of the noun-only space, since the other terms have no influence on the selection of dimensions here.

- **Noun-Only Subspaces**: the subspace is selected based only on associations with the target term: we take the 200 dimensions with the highest PMI value (as expressed in Equation 1) for the target (i.e., noun) in a given dyad.
- **Joint Subspaces**: selection is based on associations shared by the source and target terms: we select the 200 dimensions with the highest average PMI for both target and source terms in each dyad.
- **Independent Subspaces**: selection is based on independent associations with the source term and the target term, such that we select the 100 terms with the highest PMI values for the source term and the target term independently, and then merge these two sets of dimensions into a single 200 dimensional space.

An example of how a target and source dyads manifest in these three subspaces is shown in Fig. 1.

4 Results and Discussion

For both Word2Vec and CDM, cosine similarity values were computed for each word pair used in the behavioral study described above. Because the human ratings are the ground truth in this instance, Cosine Similarity is the dependent variable in each of the multiple regressions reported below. The three measures provided by human raters – Metaphoricity, Meaningfulness, and Familiarity – are the independent variables used in the analyses. The general aim is to identify which aspects of the word pairs (in terms of perceived metaphoricity, etc) are captured by cosine similarity in a given space. We also explore which type of subspace is best able to capture metaphor alone (that is, which space accounts for the most variability in human responses for metaphoricity).

We first report the results for Word2Vec, which are then used as a baseline against which to compare the results of our CDM model. The results for CDM are broken down by the type of underlying subspace.

4.1 Word2Vec results.

The results of the multiple regression analysis for Word2Vec indicated that the predictors accounted for a significant proportion of the variance in Cosine Similarity scores [$R^2 = .249$, $F(3, 224) = 24.81$, $p < .001$]. Metaphoricity significantly predicted Cosine Similarity scores, [$\beta = -0.25$, $t(224) = -3.09$, $p < .01$], as did Familiarity [$\beta = 0.22$, $t(224) = 2.65$, $p < .01$]. Low values of Metaphoricity tend to yield high values of Cosine Similarity, and low values of Familiarity tend to yield low values of Cosine Similarity. Meaningfulness was not a significant predictor in the regression.

First, these results confirm that Cosine Similarity does, in fact, capture more information than simply similarity about a given pair of terms: both Metaphoricity and Familiarity help to account for the variance in Cosine Similarity values for Word2Vec. Second, these results provide a standard by which we are able to compare the Conceptual Discovery Model's performance.

4.2 CDM results.

Unlike Word2Vec, the CDM model affords the discovery of different kinds of geometrically-defined subspaces. The crucial advantage of the CDM model is its ability to project a context-specific subspace geared towards capturing the semantics of situations in which a metaphor can be meaningfully applied. As such, our objective is to compare the performance of Cosine Similarity scores for detecting properties of metaphors using different techniques for constructing context-specific subspaces, in particular, the Noun-only, Joint, and Independent methods described in Section 3.2. The same multiple regression analysis as above was performed for these three CDM model configurations. Finally, the relationship between Cosine Similarity and Metaphoricity ratings is explored in more depth for the best performing model.

Noun-only subspaces and Joint subspaces The regression analysis for the Noun-only subspaces indicates that the predictors account for a limited proportion of the variance of Cosine Similarity scores, [$R^2 = .108$, $F(3, 224) = 9.04$, $p < .01$]. Metaphoricity significantly predicts Cosine Similarity scores, [$\beta = -0.30$, $t(224) = -3.48$, $p < .01$], where higher Metaphoricity ratings are associated with lower values of cosine similarity. The regression analysis for the Joint subspace does not yield any significant results, with $R^2 = .016$, $F(3, 224) = 1.19$, $p = n.s.$

Given the poor performance of the above models (which is quantified in the low R^2 for all four model configurations) compared with Word2Vec, we conclude that neither the Noun-only subspace nor the Joint subspace are suitable for capturing perceived metaphoricity of word pairs. Previous computational approaches to metaphor generation and interpretation [28, 31] have highlighted the fact that successful metaphors often result from situation in which the salient properties of one term (e.g., the target) are distinct from the salient properties of the other term. Therefore, the results may indicate that these two types of subspaces do not capture the necessary imbalance of salient properties between terms necessary to reflect metaphorical language. In other words, the Noun-only and Joint methods of delineating subspaces do not seem to select dimensions that are more salient for one term than the other, suggesting that Independent subspaces may provide a better way of capturing this information.

Independent subspaces The results of the multiple regression analysis for the Independently-constructed subspaces indicate that Metaphoricity accounts for a significant proportion of the variance in Cosine Similarity scores [$R^2 = .271$, $F(3, 224) = 27.73$, $p < .001$], and Metaphoricity significantly predicts Cosine Similarity [$\beta = -0.37$, $t(224) = -4.72$, $p < .001$].

Of the three multiple regressions reported here, this analysis accounts for the most variability in cosine similarity values, with an R^2 of .271, which is significantly higher than the regression for Word2Vec (where the R^2 was .249). Note that, interestingly, Familiarity is not significant in this analysis. The interpretation of this finding is discussed below in the General Discussion.

To visualize how the relationship between Cosine Similarity and Metaphoricity varies by utterance type (Conventional metaphor, Novel metaphor, and Literal word pair), a correlation analysis is shown in Fig. 2 for this best-performing model, with utterance types demarcated. Metaphoricity is inversely correlated with Cosine Similarity [$r = -.50, t = 32.49, p < .001$], such that word pairs rated as highly metaphorical tend to have low Cosine Similarity values. Novel metaphor word pairs, which participants rated highest for Metaphoricity, generally have low Cosine Similarity scores. This trend is shared by the Conventional metaphor word pairs, although the Metaphoricity scores tend to be slightly lower (this is confirmed by examining the averages of these two utterance types). Finally, Literal word pairs, which garnered the lowest ratings for Metaphoricity, tend to have slightly higher Cosine Similarity values overall.

In sum, whereas Cosine Similarity in Word2Vec is correlated with both Metaphoricity and Familiarity, the flexibility of our CDM model (specifically, the ability of our model to discover specific, conceptually-relevant spaces) allows

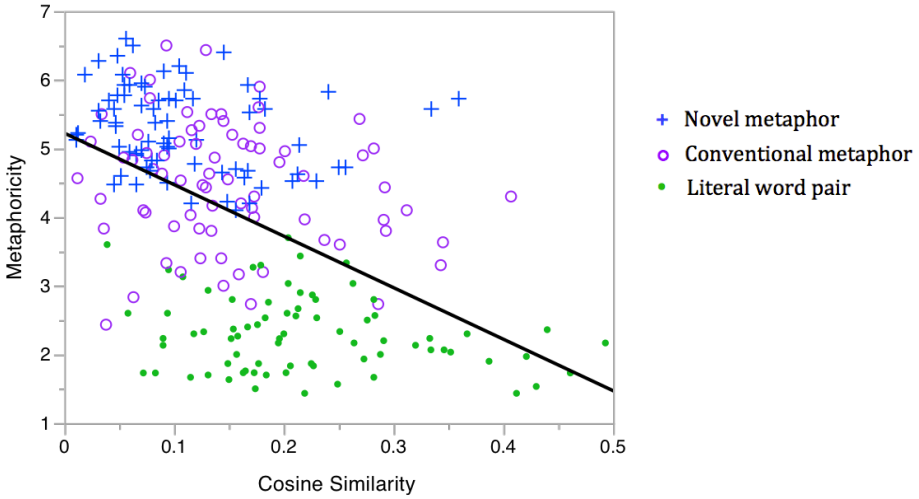


Fig. 2. Correlation between Cosine Similarity and Metaphoricity, including visualisation of Pair Types.

us to discover a space in which Cosine Similarity reflects only the *metaphorical* aspects of word pairs.

5 General Discussion

The set of results for Word2Vec and CDM offers important insight for the computational simulation of metaphoric language use. Firstly, for both Word2Vec and the Independent subspaces version of the CDM model, cosine similarity was able to account. Although only 24-27% of the variance was explained, it is important to consider that utterance type (which significantly influenced ratings) was *not* included in the statistical analyses above, because 1) the present research investigates the extent to which cosine similarity (alone) accounts for perceived metaphoricity between two terms, and 2) information about utterance type would not be available when applying our model in other contexts.

The performance of Word2Vec was used as a standard with which to compare the three variants of our CDM model. Both Familiarity and Metaphoricity were significant predictors of Cosine Similarity for Word2Vec, but for CDM, we were able to find a subspace that captures solely the perceived Metaphoricity of word pairs (because Metaphoricity was the only significant effect in the regression analysis). Furthermore, this Independent subspaces model performed best out of all of the models tested here, with an R^2 of .271 (while Word2Vec had an R^2 of only .249). Although this is only a modest improvement over Word2Vec, the difference does suggest that our CDM has a greater capacity to capture perceived metaphoricity. We therefore conclude that the Independent subspace

method offers both the most accurate and the most direct model of Metaphoricity (without confounding effects from Familiarity or Meaningfulness).

It is interesting to consider the comparative performance of the differently-configured CDM models, and explore why the Independent subspaces technique results in by far the best subspaces for mapping human judgments of metaphoricity to cosine similarity between word-vectors. In as much as a "property theoretic view of metaphor" [28, p. 56] has been laid out in computational terms, the expectation is that a successful computational model of metaphor will capture the way in which salient properties of a source are mapped to specific instances of a target. The subspaces generated by our model are intended to represent a conceptually relevant contextualisation of a lexical space: the dimensions of these subspaces consist of sets of words which taken independently offer only anecdotal glimpses into the way language is used, but which collectively can be understood as a certain *way of speaking* about a conceptually coherent topic.

So in a metaphorically relevant subspace, we hope to discover a *de facto* overlapping of some but not all of the properties of source and target. Rather than discover spaces where the mutual properties of two conceptual domains are already to some extent emphasised – as is the case with our Joint subspaces – we seek spaces where only a degree of overlap between the salient properties of each conceptual domain can be found, and where precisely this feature of a space is significant. This explains the efficacy of our Noun-only and, moreover, Independent subspaces in mapping human judgments of metaphor. In the case of the Noun-only subspace, we establish a context emphasizing the salient properties of the noun; to the extent that a verb expresses a conceptually paradigmatic action in this context, the cosine similarity between noun and verb word-vectors will be high, becoming lower as the noun-verb relationship becomes more metaphorical in nature. This phenomenon is considerably more evident in our Independent subspaces, where cosine similarity shows an inverse correspondence to the salient properties of each component of the dyad which have been merged into a single hybrid context.

It is worth noting that distributional semantic models have typically been applied to tasks involving the identification of word *similarity*, with the underlying intuition regarding these spaces being that similar words occur in similar contexts. Similarity here must be understood in a different light than the *familiarity* inherent in a word pairing: we might expect familiarity to correlate roughly with a tendency towards juxtaposition, and so a statistical measure of familiarity might emerge simply from calculating the PMI between two co-occurring words. Nonetheless, we must also note that words that tend to occur together will necessarily also tend to occur together *in the same context*, and so we might expect familiarity to emerge as a kind of artifact of this tendency in spaces geared towards similarity. It is therefore not surprising that a fairly standard distributional semantic model such as Word2Vec captures a degree of familiarity in measures of cosine similarity.

With this in mind, we might imagine a way forward towards building more nuanced subspaces particularly geared to prise apart judgments of metaphoricity. We could, for instance, investigate techniques for building subspaces that focus

primarily on the source component of a word pair – the verb, in the cases studied here – in order to draw out the salient properties of the source and then measure the degree to which these properties are typically transferable to a target. Finally, in future work we hope to use our findings regarding the geometric properties of subspaces to discover how people are likely to interpret new word pairs. In contrast to the research presented above, where human ratings were used to explain the variance in cosine similarity scores, this future direction will use cosine similarity scores for novel dyads to *predict* the degree to which human participants will perceive a given dyad as being metaphorical.

Acknowledgments

This research is supported by the project ConCreTe, which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733. This research has also been supported by EPSRC grant EP/L50483X/1.

References

1. Agres, K., McGregor, S., Purver, M., Wiggins, G.: Conceptualising creativity: From distributional semantics to conceptual spaces. In: Proceedings of the 6th International Conference on Computational Creativity. Park City, UT (2015)
2. Arzouan, Y., Goldstein, A., Faust, M.: Brainwaves are stethoscopes: ERP correlates of novel metaphor comprehension. Brain Research 1160, 69–81 (2007)
3. Barnden, J.: Uncertainty and conflict handling in the ATT-Meta context-based system for metaphorical reasoning. In: Third International Conference on Modeling and Using Context. pp. 15–29 (2001)
4. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! In: ACL 2014 (2014)
5. Bowdle, B.F., Gentner, D.: The career of metaphor. Psychological Review 112(1), 193 (2005)
6. Clark, S.: Vector space models of lexical meaning. In: Lappin, S., Fox, C. (eds.) The Handbook of Contemporary Semantic Theory. Wiley-Blackwell (2015)
7. Feldman, J., Narayanan, S.: Embodied meaning in a neural theory of language. Brain and Language 84, 385–392 (2004)
8. Gibbs, R.W., Bogdanovich, J.M., Sykes, J.R., Barr, D.J.: Metaphor in idiom comprehension. Journal of Memory and Language 37(2), 141–154 (1997)
9. Goldstein, A., Arzouan, Y., Faust, M.: Killing a novel metaphor and reviving a dead one: ERP correlates of metaphor conventionalization. Brain and Language 123(2), 137–142 (2012)
10. Grice, H.P.: Logic and conversation. In: Cole, P., Morgan, J.L. (eds.) Syntax and Semantics, pp. 41–58. Academic Press, New York (1975)
11. Gutiérrez, E.D., Shutova, E., Marghetis, T., Bergen, B.K.: Literal and metaphorical senses in compositional distributional semantic models. In: Proceedings of the 54th Meeting of the Association for Computational Linguistics (2016, to appear)
12. Jankowiak, K., Naskręcki, R., Rataj, K.: Event-related potentials of bilingual figurative language processing. In: Poster presented at the 19th Conference of the European Society for Cognitive Psychology. Paphos, Cyprus (2015)

13. Kiela, D., Clark, S.: A systematic study of semantic vector space model parameters. In: Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) @ EACL 2014. pp. 21–30. Gothenburg (2014)
14. Kiela, D., Hill, F., Clark, S.: Specializing word embeddings for similarity or relatedness. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 2044–2048 (2015)
15. Kintsch, W., Bowles, A.R.: Metaphor comprehension: What makes ametaphor difficult to understand? Metaphor and Symbol 17(4), 249–262 (2002)
16. Kutas, M., Federmeier, K.D.: Thirty years and counting: Finding meaning in the n400 component of the event related brain potential (ERP). Annual Review of Psychology 62, 621 (2011)
17. Lai, V.T., Curran, T.: ERP evidence for conceptual mappings and comparison processes during the comprehension of conventional and novel metaphors. Brain and Language 127(3), 484–496 (2013)
18. Levy, O., Goldberg, Y.: Linguistic regularities in sparse and explicit word representations. In: 18th Conf. on Computational Natural Language Learning (2014)
19. McGregor, S., Agres, K., Purver, M., Wiggins, G.: From distributional semantics to conceptual spaces: A novel computational method for concept creation. Journal of Artificial General Intelligence (2015)
20. McGregor, S., Purver, M., Wiggins, G.: Words, concepts, and the geometry of analogy. In: Workshop on Semantic Spaces at the Intersection of NLP, Physics and Cognitive Science (2016)
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of ICLR Workshop (2013)
22. Mikolov, T., tau Yih, W., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 246–251 (2013)
23. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Conf. on Empirical Methods in Natural Language Processing (2014)
24. Schütze, H.: Dimensions of meaning. In: Proceedings of the 1992 ACM/IEEE conference on Supercomputing. pp. 787–796 (1992)
25. Shutova, E., Teufel, S., Korhonen, A.: Statistical metaphor processing. Computational Linguistics 39(2), 301–353 (2012)
26. Turney, P.D., Patel, P.: From frequency to meaning: Vector space models of semantics. Journal of Artificial Intelligence Research 37, 141–188 (2010)
27. Utsumi, A.: Computational exploration of metaphor comprehension processes using a semantic space model. Cognitive Science 35(2), 251–296 (2011), http://dx.doi.org/10.1111/j.1551-6709.2010.01144.x
28. Veale, T.: A service-oriented architecture for metaphor processing. In: Proceedings of the Second Workshop on Metaphor in NLP. pp. 52–60 (2014)
29. Wolff, P., Gentner, D.: Evidence for role-neutral initial processing of metaphors. Jnl. Experimental Psychology: Learning, Memory, and Cognition 26(2), 529 (2000)
30. Wolff, P., Gentner, D.: Structure-mapping in metaphor comprehension. Cognitive Science 35(8), 1456–1488 (2011)
31. Xiao, P., Alnajjar, K., Granroth-Wilding, M., Agres, K., Toivonen, H.: Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations. In: Proceedings of the 7th International Conference on Computational Creativity (ICCC). Paris, France (2016)

An Exploratory Model of Remembering, Telling and Understanding Experience in Simple Agents

Pablo Gervás

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

Abstract. Given the importance of narrative for the way humans perceive the world and exchange information about it, it is surprising how little we know about the procedures by which reality is represented as narrative. This is in part due to the well known fact that humans are bad at being aware of their own though processes. It is also influenced by the fact that the ability to generate and process narratives is so pervasive that everybody takes it for granted. Although this is not a worrying issue in general terms, it is a significant problem for recent efforts to construct computational models of this narrative ability. The present paper describes an elementary computational model of a society of agents driven by a need for information, where the ability to represent and communicate reality as a sequential stream of symbols can be shown to provide advantages in terms of maximising the amount of information compiled by a given agent over a given period. This model is not intended as a plausible model of human cognition. The human narrative ability has broader range and significantly higher complexity. However, the model is phrased in terms of elementary principles that can also be seen to underlie more complex models. The paper discusses consequences and insights arising from this model that may be relevant for wider consideration of narrative.

Keywords: computational creativity, narrative, computer simulation

1 Introduction

Narrative has been considered as an elementary cognitive ability relevant for human beings [19, 4, 10]. Yet the process by which a particular experience of reality gets transformed into a narrative in the classic sequential sense that we consider a "story" is poorly understood. Part of the problem faced here is that for a very long time there has been no obvious representation for the experience of reality that can be considered prior to our traditional rendering as narrative. However, the advent of the digital age has progressively changed this. Computers do not represent memories in terms of narrative. Rather they allow for direct storage of experience in terms of flows of perception – audio,

video – or abstractions intended for conceptual representations – temporally [20] and/or geographically tagged data [13]. Although these alternative means of representing and storing experience present their own peculiar affordances, it has nevertheless become apparent that there is a large conceptual gap between the representation of the world used by computers and those traditionally used by humans. There are a number of efforts to bridge this gap by means of technologies for recognising actions from video [21], summarization of videos [14] and semantic annotation of videos [5]. However, these are still relatively basic procedures that generate a conceptual representation of the experiences captured in the video at a level that remains far from the narratives that a human would have used to describe the same experience. Given access to a video, or even to the results of these various processing techniques to annotate it, any human would be able to produce a brief and concise narrative about what happens in it. Yet this task is way beyond the current capabilities of machines.

An important obstacle that faces this challenge is the fact that humans are notoriously poor at identifying the processes that they apply in processing reality [16]. The field of narratology has devoted significant efforts over the years to studying narrative [1]. However, this effort has traditionally considered narrative in its final form with little reference to the way it is constructed from human perception. In recent years there has been a significant effort to relate narrative to the study of human cognition [10]. It is clear that this line of research constitutes a major challenge, given the levels of complexity involved in both narrative and human cognition. The grand picture to be considered is enormously complex and full of open questions. Existing examples of narrative can only very rarely be paired with any kind of alternative record of the experience that led to them. This constitutes a significant obstacle for applying a data-driven approach to solve this problem computationally, as these approaches require instances of both the input that lead to the communication impulse and the narrative that arose from it. Efforts to obtain insights into the processes that lead to the production of narrative have resulted in the appearance of creative writing as a specific discipline, different from traditional approaches to literature in the humanities. The current disconnect between these two radically different views of the same cultural artefact – narrative – has been identified as an open question that needs solving [11].

Whereas this lack of consensus as to the nature of the processes actually involved in the production of narrative presents a considerable difficulty for the computational modelling of narrative production, there is a possibility that computational efforts may provide useful insights that might help to clarify some of the elementary aspects that characterise the problem. Disciplines such as social psychology have long accepted the role of computer simulation as a useful tool for addressing research problems that are difficult to represent linguistically or mathematically [17]. This approach has been particularly successful in providing insights on problems that involve social interaction [2]. In the particular case of narrative, existing effort of computational modelling have focused on the traditional view of the author as an individual that works in isolation [6]. Only

recently has the social aspect of narrative as means of communication between an author and an audience been considered in computational terms [8]. An important difficulty is that the consideration of narrative is complex enough if the problem is considered exclusively in terms of the process that the writer needs to apply. Consideration of the wider social context while maintaining an acceptable level of complexity in the representation of both narrative itself as a product and the cognition of the author increases the complexity of the problem beyond what can sensibly be represented in a computer simulation. Yet the social aspects clearly play a very significant role and they may be worth studying for their own sake.

The present paper describes an exploratory model where some basic abilities of narrative-generating agents are represented in a small society driven by an accepted need for collecting information about the world, and obeying basic constraints such as a limited range of perception and a limited life span in the context of a record of time that extends over generations of agents. The need to know about a wider world, when an agent's ability to perceive the reality outside him is limited spatially by a given range perception and a given life span, creates circumstances where the ability to share fragments of experience with other agents constitutes an advantage. The consideration of the mechanics of communicating experience in this fashion, in relation to elementary operations of perception and cognition, establishes criteria both for how stories themselves need to be constructed, and how agents decide what and when to share stories with other agents.

2 Previous Work

For the purpose of this paper we want to model the way in which cognitive agents construct sequential discourses that encode a fragment of their personal experience to be conveyed to other agents, the way in which other agents interpret such discourses to enrich their own stored knowledge about the world around them, and the way such behaviours affect the management of information over a network of such agents as a whole. To support the approach followed in the paper, three areas of previous work need to be considered: the use of computer simulation in social psychology, basic approaches to agent-based modelling, and existing computational models of narrative composition.

2.1 Computer Simulation for Social Phenomena

As the model we intend to build will capture the social nature of discourse as a communication device, we need to consider previous work on the use of computer simulation for social phenomena.

Ostrom [17] argues that computer simulation can provide an alternative symbol system in which to express theories in social psychology. He argues that simulations should be undertaken especially when the complexity of the theoretical processes exceeds the ability of the theorist to hold all relevant postulates in

mind and to accurately generate predictions. He also describes five complexities inherent to social behaviour that are difficult to address using symbol systems other than computer simulation. They mostly concern the difficulty in observing a latent variable – a construct that cannot be observed directly but must instead be inferred on the basis of its observable manifestations. This is particularly true of the process of construction and interpretation of narrative from reality, in which most of the elements that would need to be captured in a computational model of the process correspond to latent variables. Ostrom's five complexities are: the fact that a single latent variable may have multiple manifestations, the influence of qualitative cognitive and social structures, the connection between latent variables and their overt expression, the interaction between multiple latent variables, and the fact that these phenomena evolve over time. These issues need to be considered for the current study.

Neches [15] outlines three possible views of the role of computer simulation in cognitive psychology: an extreme one where computer simulation is seen as a superior formalism for theory specification, and two more pragmatic ones, one where it is seen a means of exploring or validating psychological theories, and one where it is seen as a source of useful concepts. The third view relies on the view that a computer implementation of a theory may provide insights on the mechanisms involved in the phenomenon under study, by making us aware of the constraints that govern them. This third view is the most interesting for the present paper.

2.2 Agent-based Modelling

Because we want to address the way in which each agent constructs and interprets discourse, we need to consider agent-based modelling.

Helbing [9] provides an overview of agent-based simulation in which he explains that such simulations, when applied for scientific purposes, intentionally make simplifications to focus on the particular aspects under study. In this way, they may restrict to modelling very few of the properties known to be relevant to a given phenomenon, in the hope of achieving a more realistic representation of those properties. Helbing outlines a number of principles to be followed in agent-based modelling. These include: the need to describe the evidence to be explained, the importance of clarifying the purpose of the simulation, and the need to formulate a hypothesis as to the underlying socio-economic processes or fundamental mechanisms leading to the behaviour of the system – making sure that these mechanisms should be at least one level more elementary than the evidence to be understood.

In the spirit of the third approach described by Neches, we intend to develop a computer simulation that does not pretend to be an accurate model of human behaviour but rather models some very specific and very elementary aspects that are known to play a role in the communication between humans by means of discourse. Instead of building a complex model in the hope of obtaining predictions applicable to real-life situations, we hope to achieve a simple model that exhibits interesting properties that may be shared with its more complex counterparts

and may provide insights as to basic constraints that may also underlie those. The approach should be seen as a computer simulation version of the synthetic psychology advocated by Braitengerg [3].

2.3 Existing Computational Models of Narrative Composition

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 [19, 18].

Bruner [4] addresses the role of narrative in the way people achieve knowledge of the world, arguing that experience and memory of human happenings is organized mainly in terms of narrative. Bruner presents ten features of narrative that help characterise the particular view of narrative that he is considering in his argument. These ten features are: narrative diachronicity, particularity, intentional state entailment, hermeneutic composability, canonicity and breach, referentiality, genericness, normativeness, context sensitivity and negotiability, and narrative accrual. These features capture a number of important characteristics of narrative that would ideally need to be considered in any computational account of narrative. However, not all of them need to be considered in a specific model, if that model is focusing on a particular aspect.

León [12] presents an architecture of narrative memory that combines pragmatic requirements arising from the need to represent aspects of narrative deemed relevant in computational approaches with cognitive considerations. As in the case of Bruner, a number of the features captured in this approach address characteristics of narrative that are beyond the simple modelling considered in this paper.

The ICTIVS model [8] describes the process of composition of discourse – understood as a vessel to convey a message from a composer to an interpreter – in a setting where the message is complex in nature and structure but the discourse to be employed is restricted to a linear sequence of propositions. The model includes a series of iterations where the composer progressively revises a tentative discourse that she attempts to interpret following procedures that the interpreter is expected to be using. The iterations stop when the interpretation of the result satisfies the expectations of the composer in terms of how the original intended message will be reconstructed by the interpreter.

Gervás [7] presents a computational approximation to the task of composition of a narrative discourse to describe a selected subset of the moves in a given a chess game. In this case, the record of the complete chess game is understood as a source representation of reality, and the composed discourse as a narrative representation of that reality as understood by a particular composer agent.

3 A Model of Remembering, Telling and Understanding Experience

According to the principles outlined by Helbing, we will consider a population of agents that act in the world – come to life, move, interact with other agents, die

–, perceive a small subset of the world that is close to them, store information on what they have experienced, and may communicate fragments of this stored information to other agents. Our hypothesis is that the underlying mechanism that governs the interactions between these agents is a pressure to maximise the perceived amount of information that each agent has managed to compile on the world at end of its life-span. Given basic considerations of limited perception, finite life-span, and limited resources in terms of time to invest in either exploring the world or communicating with other agents, procedures for sharing information with other agents as linear discourses should provide a competitive advantage when adopted by the society of agents. The behaviours resulting from such an approach should present significant similarities with the established way of exchanging information in the form of narratives.

3.1 Basic Model

The construction of a model such as we require involves the establishment of:

- a definition of the world to be experienced
- the definition of agents as participants in the world (and thereby as objects experienced by other agents)
- a definition of agents as cognitive actors (or subjects of experience who perceive a partial view of the world and store information about it in some format)
- the establishment of a communication mechanism whereby agents may share information

The World We consider the world to be a two-dimensional space of discrete cells, such that a single agent can stand in a given cell. Each cell can be identified by its horizontal and vertical coordinates with respect to a given reference point. Although many different configurations could be used, the initial tests have been run with an 8 x 8 cell, with the reference point established at the bottom left corner.

Agents as Participants Agents are identified by capital letters. Agents are capable of moving and talking. They can move over the world in vertical or horizontal directions, one cell at a time. An example of a succession of states of the world is shown in the first column of Table 1. The behaviour of each agent is controlled by a set of modules that encode the heuristics to decide when and where it will move and when it will talk. These modules can be configured in different ways, to allow for different degrees of mobility and garrulousness over the full set of agents. The values for these configurations parameters may play a role in determining the success of individual agents and/or the overall success of the social communication strategy. Specific modules also control reproduction behaviour and agent's demise. Each agent is spawned at a random position in the world.

	a	b	c	d	e	f	g	h
1								
2					A			
3						C		
4								
5								
6			B					
7								
8							D	

	a	b	c	d	e	f	g	h
1	X	X	X	X	X	X	X	X
2	X	X	X	X	A			X
3	X	X	X	X		C		X
4	X	X	X	X				X
5	X	X	X	X	X	X	X	X
6	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X

Burst of discourse (absolute)

```
time 00
C at f3
A is at e2

C move to f4
A disappears

C move to f5
B appears at e6
```

	a	b	c	d	e	f	g	h
1								
2				A				
3								
4						C		
5								
6				B				
7								
8						D		

	a	b	c	d	e	f	g	h
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X				X
4	X	X	X	X		C		X
5	X	X	X	X				X
6	X	X	X	X	X	X	X	X
7	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X

Burst of discourse (relative)

```
time 00
C at f3
A is nw

C move s
A disappears

C move s
B appears se
```

	a	b	c	d	e	f	g	h
1								
2				A				
3								
4								
5						C		
6					B			
7								
8						D		

	a	b	c	d	e	f	g	h
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X				X
5	X	X	X	X		C		X
6	X	X	X	X	B			X
7	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X

Table 1. Succesion of views of the state of the world, partial views of the world as observed by agent C and the discourse burst that C might produce to convey its experience to other agents.

Agents as Cognitive Subjects Agents are assumed to have perceptive abilities that allow them to perceive all actions that happen within a given radius of their current position. This is considered the range of perception. The range of perception is set at one cell in all directions – including diagonals – to ensure that agents that do not move have a limited perception of the world, and that agents wanting to report their experience at a given point in time have a limited amount of information to deal with.

Agents store their perception of the world around them as a partial view – determined by their current position and their range of perception – of the map of the world at a given point in time. As agents move, their perception will shift to a different portion of the world. Agents are capable of converting their perception of the world into this type of view. The overall record of a given agent will therefore be a sequence of snapshot of such partial maps. An example of a succession of such partial maps is shown in the second column of Table 1.

Communication Mechanism The purpose of the simulation is to explore the effects of the hypothesized pressure to compile information about the successive states of the world on the different strategies for constructing messages, sharing them with other agents, and interpreting those received from other agents.

It is clear that agents limited to perceiving the world as they move around are unlikely to reach high levels of coverage over the absolute space of all states of the world over time, due to their limited perception. The possibility of sharing their partial views with other agents would significantly increase the coverage they can collectively achieve.

In an ideal setting, an agent would be able to share all its memories with any agents within its range of perception. However this would be unrealistic. We need a model of communication that incorporates restrictions on the time devoted to communication. The amount of information conveyed should have some correlation with the time invested in communicating. This places pressure on the selection of what information to communicate and on the format used to communicate it, which are some of the features we are interested in.

This establishes some initial constraints on the format of communication of information. If at a given point in time agents were to communicate a full description of the NxN square that they perceived at some other moment in time, the listening agent would be restricted to a one shot view of the world, with little change of dynamic information. Even if the ratio between the processing ability for perception and interpretation were changed, the possibilities of covering appreciable segments of the world in this way would be small. So some procedure of optimising the encoding of experience must be put in place. This is one of the aspects of the simulation that can be tested during exploration.

Any such messages – given that they may refer to location, moments in time, and agents other than the current ones – must explicitly encode location, time and protagonist[1] in such a way as to allow an interpreting agent to place them

[1] We are at present conflating the roles of protagonist and narrator, as we assume that any such basic communications would necessarily be phrased in the first person.

in the appropriate context. An example of discourse bursts is shown in the third column of Table 1. A discourse using absolute values for locations constitutes a compact representations of the set of partials views it encodes. The use of speaker-centric relative values for describing locations is even more compact.

As agents need to construct this type of message, different solutions to this task may have different impact on the overall success of both individual agents and the collective as a whole. These aspects will be explored in the simulation.

Each agent must be capable of converting any messages received into its own representation of the world. Table 2 shows an example of how the knowledge of the world held by agent B, before and after processing the example discourse generated by C (see Table 1). Again, the procedures for this need to be explored in the simulation.

	a	b	c	d	e	f	g	h
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X
5	X				X	X	X	X
6	X		B		X	X	X	X
7	X				X	X	X	X
8	X	X	X	X	X	X	X	X

	a	b	c	d	e	f	g	h
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X
5	X	X				X	X	X
6	X	X		B		X	X	X
7	X	X				X	X	X
8	X	X	X	X	X	X	X	X

	a	b	c	d	e	f	g	h
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X	X	X	X	X
5	X	X	X			C	X	X
6	X	X	X		B		X	X
7	X	X	X				X	X
8	X	X	X	X	X	X	X	X

	a	b	c	d	e	f	g	h
1	X	X	X	X	X	X	X	X
2	X	X	X	X	A			X
3	X	X	X	X		C		X
4	X	X	X	X				X
5	X				X	X	X	X
6	X		B		X	X	X	X
7	X				X	X	X	X
8	X	X	X	X	X	X	X	X

	a	b	c	d	e	f	g	h
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X				X
4	X	X	X	X		C		X
5	X	X						X
6	X	X		B		X	X	X
7	X	X				X	X	X
8	X	X	X	X	X	X	X	X

	a	b	c	d	e	f	g	h
1	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X	X	X
4	X	X	X	X				X
5	X	X	X			C		X
6	X	X	X		B			X
7	X	X	X				X	X
8	X	X	X	X	X	X	X	X

Table 2. Succession of partial views for agent B (top row) and the result of enriching B's knowledge of the world (bottom row) after processing from C the discourse given in Table 1

4 Competing Configurations of the Model

Each of the elements of the model needs to be configured to behave in particular ways. The different possible configurations interact with one another. To explore

Subsequent efforts may delve into the details that may arise from considering more elaborate approaches.

the full set of possibilities is beyond the scope of this paper. However, some basic possible configurations are reported to demonstrate the potential of the approach.

4.1 Movement

Agents can move freely around the board. The way in which they decide to move will affect the amount of information that they have available at the end of their life-span. Agents may prefer to remain near fixed positions as this may maximise their personal perception of relative information coverage. Since they are unaware of other locations, they may reach the end of their life-span under the conviction that they know all there is to know about what happened in (their limited view of) the world. However, appearance of another agent coming from other parts of the world may disturb this brittle impression, as their stories may identify far away places (and may set the listener wondering about what may have happened there).

Alternatively, agents may decide to wander, traversing the world to explore what can be found beyond their range of perception. This type of agent is likely to discover a larger subset of the overall available space, but as a result it will develop a lower personal perception of relative information coverage.

4.2 The Role of Communication

Agents need to decide how they approach the task of composing the type of message that is being passed around. Although agents are in theory free to compose messages as they see fit, it is clear that comparative advantage will only be achieved if some agreement is reached between the way in which agents compose their messages and the way other agents interpret them.

A baseline procedure for carrying out this task can be imagined by considering analogies with the way humans go about similar tasks.

Agents may optimise the encoding their partial view of the world at a given point in time in two different ways: by restricting what is reported in a message to exceptional elements – themselves carrying out some action, an action by another agent being seen, another agent being visible, becoming visible or disappearing as a result of some action –, and by describing their perception in terms relative to themselves.

By following these two strategies, agents describe only the highlights of their partial view in terms relative to their position at that point in time. This allows for several moments in time to be so described within the space equivalent to a full perception of the view at a given moment (which would also include encoding for the empty spaces). In order for other agents to be able to interpret this correctly (see below), the first segment of such a representation is devoted to establishing the position and the moment of time of the telling agent at the start of conveyed span of experience. As these spans may be passed on from one agent to another, the next segment of the representation indicates who the protagonist agent of the span is. To avoid the use of overloaded terms that might

cloud the issue – such as story or narrative, which may bring in preconceived notions of the reader – we will refer to this medium of communication as a *burst of discourse*. An example of these various mechanisms at work can be seen in the discourse burst produced from a partial view of world shown in Table 1.

An important advantage of the relative approach to description of perception, is that the interpretation of busts of discourse of this type can be achieved by reusing the procedure by which a perceiving agent constructs the map view of experience that it would have if experiencing the events conveyed by the message.

5 Discussion

The approach described in the present paper constitutes a preliminary description of a computer simulation that is currently under construction. The basic infrastructure for handling a number of agents in the type of environment described is already available. Implementations of the type of epistemic agent described, capable of perceiving the world in the manner described and storing information obtained from those perception in the manner outlined is already available. Basic implementations of some of the possible approaches to composition of bursts of narrative discourse of the type described are under way.

The description of the model as presented in the paper already permits the identification of a number of problems that any computational model of narrative would have to take into account.

First, Ostrom's five complexities need to be considered as they affect the modelling of narrative. The description provided at the level of detail required for modelling the phenomenon in this fashion, ilustrates the fact that a number of crucial aspects – such as the specific procedures used for composition and interpretation of bursts of discourse – may be playing the role of latent variables, in the sense that they cannot be observed directly but only as they affect the results that are exchanged betwen the agents. In the particular case of studying narrative as it occurs in the world, the problem is further compounded by the fact that the actual perception (or conception if narrative is understood in broader terms to encompass fiction as it mostly does) that the composer wants to convey also becomes latent in a similar way: it is unavailable for observation. The proposed model has the advantage of providing a controlled experimental setting in which this perception of reality is indeed observable as it is explicitly modelled outside the agent. Nevertheless, Ostrom's concerns about the interaction between multiple latent variables and their evolution over time are also relevant for this approach, and computer simulation may provide the means of exploring them.

Second, Helbing's observation that an agent-based modelling approach need to formulate a hypothesis as to the fundamental mechanism that drives the behaviour of the system – at least one level more elementary than the evidence to be understood – has been followed in the present proposal by restricting the set of features explicitly represented in the system to elementary aspects of cognition and perception. The main hypothesis underlying the proposed approach is

that such features constitute a basic network of constraints that establish the fundamental form of discourse that humans can employ to communicate narrative. Although this constitutes a very crude representation of the much more complex phenomena that need to be addressed for narrative – such as interest, affect, emotion –, it provides a valuable guideline which may help in the search of meaningful baselines to implement the various procedures that would be required to obtain an operational version of the described architecture.

Third, the proposed approach is restricted to a small subset the ten features of narrative described by Bruner. Overall, the described model is indeed based on the way in which man achieves knowledge of the world – which Bruner lists as the original motivation for his discussion –, and considers explicitly the interaction between perception and transmission between agents. The underlying framework for representing the world and the solution used to represent agent's knowledge stores satisfy narrative diachronicity. The representation considered only deals with burst of discourse concerning particulars, so they lack the additional layer of acting as tokens of broader types. The remaining features that Bruner considers would correspond to more elaborate layers of understanding and representation than considered here, but it is our belief that even those more elaborate layers would need to operate within the constraints established by the more elementary features concerning this particular type of communication of experience between agents.

6 Conclusions

The present paper proposes a computational model of basic interchange of information about the world between agents that have their own perceptions but may learn about the world also from the perceptions of others. Elementary issues of the ratio between the information compiled versus the effort invested in its acquisition are interpreted into a set of constraints that drive possible solutions for communication mechanisms between agents into formats surprisingly similar to those observed in the human approach to telling stories. This is considered a promising insight.

Acknowledgements

This paper has been partially supported by the projects WHIM 611560 and PROSECCO 600653 funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program.

References

1. H.P. Abbott. *The Cambridge Introduction to Narrative.* Cambridge Introductions to Literature. Cambridge University Press, 2008.
2. Robert M. Axelrod. *The complexity of cooperation: Agent-based models of competition and collaboration.* Princeton University Press, Princeton, NJ, 1997.

3. V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology*. Bradford Books. MIT Press, 1986.
4. Jerome Bruner. The narrative construction of reality. *Critical inquiry*, pages 1–21, 1991.
5. Stamatia Dasiopoulou, Eirini Giannakidou, Georgios Litos, Polyxeni Malasioti, and Yiannis Kompatsiaris. *A Survey of Semantic Image and Video Annotation Tools*, pages 196–239. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
6. Pablo Gervás. Computational approaches to storytelling and creativity. *AI Magazine*, 30(3):49–62, 2009.
7. Pablo Gervás. Composing narrative discourse for stories of many characters: a case study over a chess game. *Literary and Linguistic Computing*, 29(4), 08/14 2014.
8. Pablo Gervás and Carlos León. *Integrating Purpose and Revision into a Computational Model of Literary Generation*, chapter Integrating Purpose and Revision into a Computational Model of Literary Generation. Lecture Notes in Morphogenesis. Springer, esposti, mirko degli, altmann, eduardo, pachet, francois (eds.) edition, 2016.
9. D. Helbing. *Agent-Based Modelling*. Understanding Complex Systems. Springer Berlin Heidelberg, 2012.
10. D. Herman. *Story Logic: Problems and Possibilities of Narrative*. Frontiers of narrative. University of Nebraska Press, 2004.
11. P. Howarth. Creative writing and schiller's aesthetic education. *The Journal of Aesthetic Education*, 41(3):41 – 58, 2007.
12. Carlos León. An architecture of narrative memory. *Biologically Inspired Cognitive Architectures*, 16, 04/2016 2016.
13. Jiebo Luo, Dhiraj Joshi, Jie Yu, and Andrew Gallagher. Geotagging in multimedia and computer vision—a survey. *Multimedia Tools and Applications*, 51(1):187–211, 2011.
14. Arthur G. Money and Harry Agius. Video summarisation: A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19(2):121 – 143, 2008.
15. Robert Neches. Simulation systems for cognitive psychology. *Behavior Research Methods & Instrumentation*, 14(2):77–91, 1982.
16. Richard E. Nisbett and TImothy Wilson. Telling More Than We Can Know: Verbal Reports on Mental Processes. *Psychological Review*, 84(3):231–259, 1977.
17. Thomas M Ostrom. Computer simulation: The third symbol system. *Journal of Experimental Social Psychology*, 24(5):381 – 392, 1988.
18. R. Schank. *Dynamic Memory : A Theory of Reminding and Learning in Computers and People*. Cambridge University Press, 1982.
19. R. Schank and R. Abelson. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. L. Erlbaum, Hillsdale, NJ, 1977.
20. Y. Tang, X. Ye, and N. Tang. *Temporal Information Processing Technology and Its Applications*. Springer Berlin Heidelberg, 2011.
21. Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224 – 241, 2011.

Towards an approach for the computationally assisted creation of insight problems in the practical object domain

Ana-Maria Olteţeanu[1⋆]

Abstract. Insight problems are creative solving problems used to assess creativity in human participants, and empirically study insight-related processes. However, not many such problems exist, and once a participant has been exposed to a particular problem, insight cannot be studied anymore in that context. This paper proposes an approach for creating more insight problems in the practical object uses domain. This approach uses cognitive understanding of some of the processes that take part in the solving of insight problems in this domain to create more such problems, and is amenable to computational implementation. The manual creation of a couple of such problems with this approach is described. Ways of making the approach computational are briefly discussed.

1 Introduction

Insight is an impressive phenomenon. Classical insight stories involve great artistic or scientific achievements – like the ones presenting (i) Archimedes shouting Eureka after having observed properties of water displacement that could help him solve the crown problem, (ii) Watson dreaming of spiral staircases before proposing the double helix structure of the DNA molecule and (iii) Kekulé's daydream of an Ouroboros snake eating its own tail, before coming up with the structure of the benzene molecule.

Insight, however, does not have to lead to big discoveries and involve historic level creativity [Boden, 2003]. Insight presupposes seeing an existing problem in a new way – a way in which it becomes solvable for the person seeing it. Insight is studied empirically using various types of insight problems created by humans. For example, [Dow and Mayer, 2004] have gathered a collection of problems which they split into the following categories: mathematical, spatial and verbal insight problems. A mathematical insight problem they mention is: *Which would be worth more, a pound of 10 dollar pure gold coins or half a pound of 20 dollar pure gold coins; or would they be worth the same? Explain your answer.* A spatial insight problem gives the participant Figure 1 and gives them the following task: *without lifting your pencil from the paper, show how you could join all 4 dots with 2 straight lines.* An example of a verbal insight problem from the same collection is the following: *The legendary runner Flash Fleetfoot was so fast that his friends said he could turn off the light switch and*

⋆ Corresponding author

jump into bed before the room got dark. On one occasion Flash proved he could do it. How?

Fig. 1. The four dots problem

In this paper we will discuss a different (though at times overlapping) category of insight problems – problems which involve the practical use of daily objects, like the candle problem [Duncker, 1945] and the two strings problem [Maier, 1931][1]. This category generally involves objects and creative ways of thinking about objects, their relations and their affordances, as well as creative modes of re-representing problems concerning objects and practical physical goals.

As fascinating as some of them might be, insight problems generally stop being useful for the empirical study of the insight process if the participant has encountered them even once beforehand. If the problem was once encountered, the participant might simply remember what the solution was, rather than struggle to re-represent the problem, or experience a moment of sudden re-representation. The experiment sessions involving problems which are already known by the participants will thus be devoid of chances in gathering data on authentic insight and creative problem solving processes. Considering that one such problem can take a while to solve, and experimenters generally plan for only a few such problems per experimental session, not having enough new problems can result in lost experimental sessions, and the necessity of gathering more empirical datapoints, to substitute for the ones in which a participant was acquainted with one of the problems.

Within the practical object uses domain, not many insight problems exist, and some of them are quite old problems, created by the minds alike those of Maier and Duncker. A bigger dataset of such problems would benefit work focused on the understanding of human creative cognition, and could further find its application in creative problem solving work in robotics, and ambient intelligence. Therefore, defining an approach to create more such problems, an approach which could rely on computational means for problem creation assistance, would be useful for both cognitive psychologists and for AI. We already have some knowledge about the cognitive solving process of such problems. Creating them will benefit from such knowledge and bring about computational applications for it.

This paper thus sets out to define an approach to creating insight problems in the practical object uses domain, which is amenable to computational assistance and/or implementation. This approach is cognitive in nature, looking at

[1] [Dow and Mayer, 2004] classify both of these as spatial problems. In the following we will refer to such problems as insight problems in the practical object domain, as to differentiate them from spatial problems involving abstract patterns, like the one in Fig. 1.

insight problems and their creation through the lens of insight related processes explored in a previous cognitive theoretical frameowork of creative problem solving [Olteţeanu, 2016,Olteţeanu, 2014]. In the following, existing classical insight problems will be analysed (Section 2); the approach will then be proposed (Section 3); the cases of applying that approach to the creation of two such problems will be explored (Section 4), and potential reliance on computational assistance in problem creation with existing or future tools will be discussed (Section 5).

2 Classical insight problems – analysis

In this section, three examples of classical insight problems are taken as case studies: (i) the candle problem, (ii) the two strings problem and (iii) the cardboard problem. These are analysed from the perspective of what makes them insight problems, and what processes could have been used to create them. Principles are then extracted for the creation of new problems. During this analysis, we will assume the following:

(a) there is a simple, non-creative (or much less creative) version of the problem which does not require insight;
(b) that by observing the steps and possible stumbling blocks of the solvers, while they receive the version of the problem which does require creative skill, we can extrapolate some processes which make a problem require creativity;
(c) that by using a variety of such interpolated processes and knowledge about cognitive problem solving, we can begin to create more such problems, even if we do not yet cover the entire variety of skills which can be tested through them.

Candle problem

The candle problem by [Duncker, 1945] is stated as follows: *You are given a candle, a box of thumbtacks and a book of matches* (see Fig. 2). *You are supposed to fix the lit candle unto the wall in a way that does not allow the wax to drip below.*

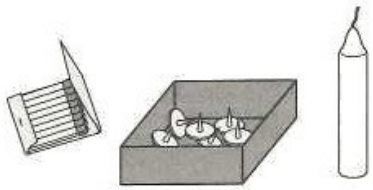


Fig. 2. The Candle problem

The candle problem is solved by taking out the thumbtacks from the box, and using the box as a container and platform for the candle, then fixing it (with thumbtacks) to the wall. The participants that get stuck while solving

this problem generally have trouble seeing the box of thumbtacks as a possible container; this is sometimes helped if the box is empty.

Considering the types of processes involved in creative problem solving, including re-representation, one can attempt to reverse engineer them, and check whether this is conducive to the creation of similar problems. The following aspects can be synthesized from the candle problem to help the creation of new problems:

- Hiding an object which is necessary for a (non-creative) solution of the problem within a different other object. This aspect shows up in this problem by the fact that a candle holder, which would have made the problem straightforward, is not present. Such a candle holder needs to be re-represented out of existing problem objects - from the perspective of the problem creator, one can look at this process as one of hiding the candle holder in a different object (or set of objects) with a similar affordance. Such a similar affordance however needs to be inferred creatively, perhaps in a process similar to that used when solving the Alternative Uses Test [Guilford et al., 1978].
- Hiding the affordance of an object by emphasizing a different affordance and (optionally) having that affordance already taken up or in use. This aspect shows up in the candle problem by the act of adding the thumbtacks inside the thumbtack box[2]. Adding them near the box would have been a case for emphasizing the affordance of the box as a container. Adding them within the box is a case for having the affordance already in use. The purpose of the latter is, of course, to trigger and thus help study the functional fixedness bias.

Two strings problem

The two strings problem by [Maier, 1931] presents to the participant a situation like the one in Fig. 3. The participant is told: *A person is put in a room that has two strings hanging from the ceiling. The task is to tie the two strings together, but it is impossible to reach one string while holding the other.*

The two strings problem is solved by making a pendulum from one of the two strings and from a heavy object laid on the floor, like the pliers, then launching one of the strings in pendular motion, as to be able to have it come on its own towards one's hand. The participants that get stuck when solving this problem usually fail to see: (i) that the object can be set in motion on its own, rather than requiring the motion of the solver; (ii) that the object could be created, as this requires making it using other objects in the room.

Considering the steps involved in solving this problem, one can assume the process of creating such a problem involves the following aspects:

- As a general strategy, make objects which need to be used for the solution lose part of their affordances. A specific technique, observed in this case, is to enable affordance loss by removing affordance related parts of the object. The object will thus need reconstruction, while the parts themselves are less likely to trigger the same affordance. In the two strings problem, this is done by removing the saliency of the affordance of a pendulum to be mobile, through splitting the pendulum in

[2] It is the thumbtacks that make this a thumbtack box anyway, and one might be biased to look at this box as a special container for the thumbtacks because of the very description of the problem, which includes the verbal tag "thumbtack box".

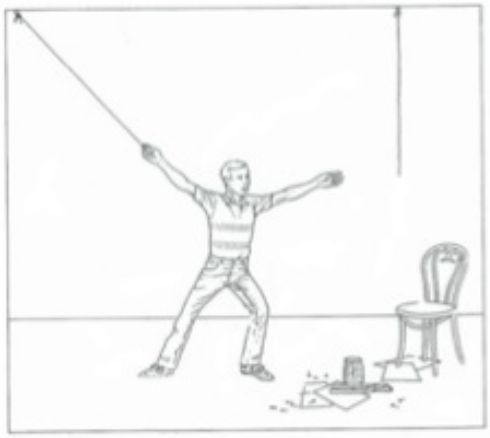


Fig. 3. The Two Strings problem

two parts. A self-based motion frame of reference might be emphasized in the verbal description of the problem (i.e. "it is impossible *to reach a string* while *holding the other*"), which focuses the participant on thinking of themselves, rather than other objects, as mobile.

– Split said object into parts and scatter the parts across the room. This is somewhat overlapping in the context of this problem with the principle above. However, the principle above can also involve removing solution-leading affordances of an object in different ways, by, for example, changing the frame of reference, representing the object in a space which makes the affordance less salient and constrains thinking about the object.
– Hide the object required to solve the problem (or object parts) within other objects. This principle is shared with the candle problem, with the added bonus that, here, not just objects but also parts of objects are exchanged for similarly affording objects. Here, the weight of the pendulum is hidden in heavy objects across the floor.
– Add objects that lead to other possible affordances, and thus other possible constructions – in this case the chair (getting up on the chair gets triggered), the nails (fixing one of the strings closer to the other by using the nails gets triggered).

Cardboard problem

The cardboard problem by [Duncker, 1945] is given as follows: *You are asked to help the experimenter attach this piece of cardboard to the loop on the ceiling. How do you proceed?* (our reconstruction of the depiction - Fig. 4).

This problem is solved by removing one of the paperclips, turning it around to make an S-shaped hook, then using one end of the hook to pierce the corner of the cardboard (the white paper corner can thus be kept in place too) and the other end to attach to the loop. Most people attempt to solve this problem using the nails and hammer on the table, various ways of standing the cardboard and attaching it with nails to the loop.

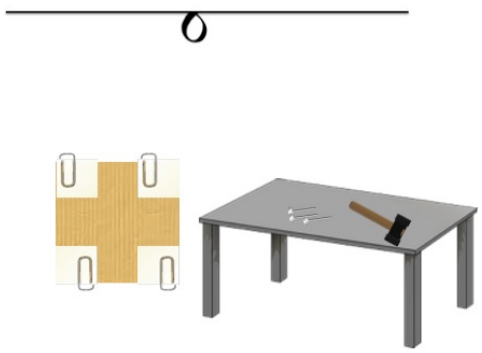


Fig. 4. The Cardboard problem

Creating problems like the cardboard problem could include the following aspects:

- Hiding part of the solving objects as parts of other objects. In this case, the paperclips are attached to the cardboard, and thus can be perceived as being part of it, thus part of the object which needs attaching to the loop, rather than of the tools which the attaching can be done with.
- Using one of the actions people would refrain from doing as part of the solution – i.e. destroying an existing object, disobeying rules or norms or arrangements perceived as implicit or unbreakable. In this case, not only the paperclips are used to hold together the white piece of paper at the corner of the cardboard (using them thus having the possible consequence of disassembling that part of the object), but also piercing the cardboard can be viewed as a way of damaging the cardboard irreversibly (in other problems, parts of objects can be pulled away by disassembling the object to pieces, however the object can also be reassembled).
- Have other objects with a similar affordance in the scene, as to interfere with finding the objects which would truly provide the solution. In the case of this problem, the affordance that nails and hammer have to fix something to a wall or a ceiling interferes with seeing the less salient affordance of the paperclip, setting the nails and hammer center stage as red herrings.

3 Approach

Various processes of solving seem necessary for the above described problems, and thus various principles of creating more such problems become apparent after this analysis. In this section, the extracted principles are discussed and summarized in an approach.

As mentioned in section 2, one of the parts of our analysis assumed that there is a simple, non-creative version of the problem, which does not require insight. For example, the candle problem would be in its simpler form if it would offer a candle holder as part of the existing objects, rather than the thumbtack box;

the two strings problem would be in its simpler form if one of the strings would be a pendulum already; the cardboard problem – if the paperclip was already twisted in the necessary S shape, or at least detached from the cardboard.

Consequently, this approach starts from the assumption that more creative problems can be constructed in the practical object domain, by taking simple day to day non-creative problems, considering one of their normal solutions, and then hiding the possibility of applying that solution from the solver via a set of re-representations and creative uses of objects (which then have to be traversed back by the solver) and problem templates (or sets of viable action plans).

A non-exhaustive list of some of the techniques that can thus be used in problem creation, in light of the previous case studies, includes:

(i) Diminishing the saliency of the objects required for the solution, by (a) putting them in a different context of affordances and possibly (b) having those affordances already allocated or used;
(ii) Hiding objects in a different form – by re-representing them as other objects which have similar properties and affordances (but for which said affordances might not be as salient as for the initial objects);
(iii) Decomposing the solution in different parts, and re-representing the parts in different structures or objects;
(iv) Representing needed parts as integrated parts of other objects;
(v) Using an object twice in the solution, with two different contexts of affordances. In this case, participants need to look at both sets of affordance contexts, to perceive the object in both of its potential roles, similar to being able to look at two ambiguous figures the perception of which can emerge from the same set of elements;
(vi) Adding to the problem other salient objects, the affordances of which might interfere with the solution;
(vii) Making use of natural or learned biases against breaking objects, crossing commonsense or common practice norms or aesthetic values.

Part of the techniques in this approach could be loosely summarized as the following (obj stands for object, aff for affordance, PT for problem template[3], sol for solution and $simProp$ for similar properties):

1. Embed in different affordance contexts:
   If $(obj_x \in PT_{sol}) \wedge \quad (\exists aff(obj_x) \neq aff_{sol}(obj_x) \vee$
   $\exists aff(obj_x, obj_k) \neq aff_{sol}(obj_x) \vee$
   $\exists PT_x | obj_x \in PT_x, aff(PT_x) \cap aff_{sol}) = 0)$
   then $display(aff(obj_x) | aff(obj_x) \neq aff_{sol}(obj_x) \vee$
   $display(aff(obj_x, obj_k)) \vee$
   $display(elementsOf(PT_x))$
2. Use creative object replacement:
   If $obj_x \in sol \wedge \quad \exists obj_a | simProp(obj_x, obj_a) \vee simProp(obj_x, partOf(obj_a))$
   Then $replace(obj_x, obj_a) \vee replace(obj_x, partOf(obj_a))$

[3] A problem template is a set of actions that will lead to a particular solution or affordance; such sets of actions are part of the commonsense knowledge of the subject. Creative use of problem templates is detailed in other works [Olteţeanu, 2014].

3. Decompose object: $decompose(obj_x) = partsOf(obj_x)$
4. Represent needed parts as parts of other objects:
   If $obj_x \in PT_{sol} \wedge partOf(obj_b, obj_x) \wedge obj_b \notin problem \rightarrow replace(obj_x, obj_b)$
5. Double use:
   If $obj_x \in PT_{sol} \wedge obj_a \in PT_{sol} \wedge simProp(obj_x, obj_a) \rightarrow remove(obj_x)$
6. Adding other salient objects or templates:
   If $\exists aff_x | sim(aff_x, aff_{sol}) \wedge aff_x \notin PT_{sol} \wedge aff_x \in PT_x, PT_x \notin PT_{sol}$
   $show(aff_x) \vee show(obj_{k1}, obj_{k2}...obj_{kn} \in PT_x)$

4 Creating new insight problems – two cases

In this section, two problems that were created using the above principles are discussed. These are: (i) the blown away teddy problem and (ii) the Jack and Jill weight problem.

The blown away teddy problem

The blown away teddy problem presents the participant with the following task: *The wind blew your son's teddy bear from the clothesline into your neighbour's garden. The neighbour is in holidays and the fence is too high to climb. How can you retrieve the teddy?* Fig. 5 shows the problem.

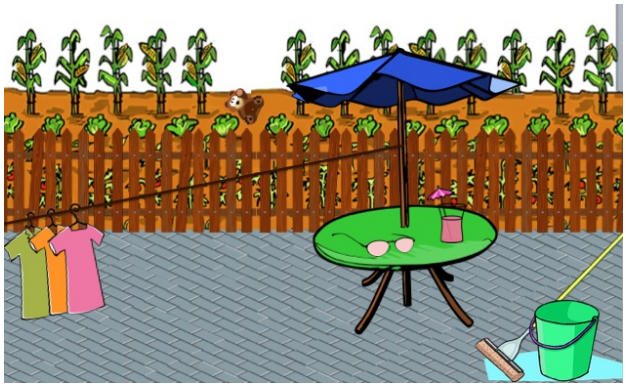


Fig. 5. The blown away teddy problem

This problem is solved by constructing a fishing rod, using the mop, the clothesline, and a clothes hanger attached to the clothesline.

The problem was constructed using the approach proposed in Section 3, in the following steps:

1. Start from a problem and an existing solution. The problem is that you need to obtain an object that is far away. The solution is to reach for the object.
2. Make solution creative. A fishing rod is used for fishing, but can have the alternative use of hooking something that is far away. The object replacement part of a system like OROC [Olteţeanu and Falomir, 2016] can be used to generate alternative uses.

3. Decompose object into parts, so that it will require composition. The parts of the fishing rod used here are the pole, the string and the hook.
4. Hide the parts of the solution in other objects with different salient affordances, or in parts of other objects. The string is presented as a clothesline (one to one object mapping). The hook is presented as part of the clothes hanger (part of object). The pole is presented as part of the mop (part of object). The participant can also attempt to use the pole that is part of the umbrella, if the participant perceives this as movable.
5. Embed the replacement (or re-represented) objects in different contexts of affordances. The clothesline is presented holding clothes to dry, attached to the umbrella pole. The hangers have clothes on them, and are also partly obscured visually by the clothes (the part which provides the affordance conducive to the solution is, however, visible). The mop is presented next to the bucket and a water puddle, which emphasize its cleaning affordances.

Parts of the problem are also ambiguous – how tall the fence is, the height of the table, the distance to the teddy. The experimenter can let the participant attempt various strategies within this ambiguity, to observe various types of constructions created, paths pursued and forms of creative reasoning. Other constraints can then be set in place and communicated to the solver (e.g. the teddy is too far to be reached just using the mop) in order to observe new strategies at play.

This process can be summarized as the following:

1. Initial problem:
   (a) Starting condition: $faraway(subject, teddy)$.
   (b) Goal: $has(subject, teddy)$.
   (c) Solution: $reach(subject, teddy)$
   (d) Starting template: $reach(subject, teddy) \rightarrow has(subject, teddy)$
2. Creative version of a *reach* template:
   $reach(subject, teddy) \rightarrow fish(fishing\ rod, teddy); beach, river, pool \notin problem$[4]
3. Decompose:
   $decompose(fishing\ rod) = (pole, string, hook)$
4. Creative replacement:
   (a) $creative\ replacement(string) = clothesline$
   (b) $creative\ replacement(pole) = mop\ handle$
   (c) $creative\ replacement(hook) = clothes\ hanger$
5. Embed in affordances or used affordances:
   (a) $embed(clothesline) \rightarrow show(attached(clothesline, pole), on(clothesline, clothes))$
   (b) $embed(mop\ handle) \rightarrow embed(mop) \rightarrow show(nextTo(mop, bucket))$
   $show(mopping(mop, water))$
   (c) $embed(clothes\ hanger) \rightarrow show(on(clothes\ hanger, clothes))$,
   $show(on(clothesline, clothes\ hanger))$

[4] This means we have to avoid a fishing related context for the rest of the scene, so the scene cannot be of a subset of scenes that might trigger fishing templates, like *beach* and *river*, nor scenes that are similar to them – like *pool*.

Jack and Jill weight problem

The Jack and Jill weight problem presents to the participant a situation like the one displayed in Fig. 6. The participant is given the following task: *Jack and Jill are arguing about whom weighs more. What could they do to find out for certain?*

Fig. 6. The Jack and Jill weight problem

This problem is solved by making a seesaw from the bucket and a surfing board and placing Jack and Jill at opposite ends.

The problem was constructed using the following steps and techniques from the approach:

1. Start from a problem and an existing solution. The initial problem template was that balancing scales are used to measure weight.
2. Change the problem so that the solution would be creative – use seesaw instead of scales; put the problem in the beach setting (change of context to one which less affords thinking about weights and balancing scales, like a kitchen), and use of a different object with similar properties.
3. Split the object in various parts – the seesaw was split into a pivot and a support plank.
4. Hide the objects which form the solution by re-representing them as other objects or object parts – the plank was turned into a surfboard (also adaptation to the current context) and the pivot into a bucket (similar adaptation). Through the adaptation to context, both objects thus can be envisaged as belonging to a normal beach scene, rather than triggering the attention of the participant as objects that have especially been added to the beach context because they are part of the solution.
5. Hide the objects in different contexts of affordances and possibly have those affordances be already in use – the bucket is turned with its side up, and in its container affordance (full of sand); the surfboard is being surfed on, and quite far away, which makes it visually less salient. Some participants might also have social qualms with solution steps that involve asking for an object which is being currently used and belongs to someone else.

6. Add objects which act as red herrings, providing a similar affordance as the one necessary for solving the problem, thus getting the participant started on a different track – the small plastic swimming pool can get the participant started in this case on an Archimede's principle template.

This process can be summarized as the following:

1. Initial problem defined:
   (a) Starting condition: $unknownWeight(x, y)$.
   (b) Goal: $findWeightDifference(x, y)$.
   (c) Solution: $balance(x, y)$
   (d) Starting template: $balance(x, y) \rightarrow findWeightDifference(x, y)$
2. Creatively change the initial template (problem+solution):
   $balance(x, y) \rightarrow seesaw(person_x, person_y), PT_x \neq kitchen$
3. Decompose:
   $decompose(seesaw) = (pivot, support\ plank)$
4. Creative replacement:
   (a) $creative\ replacement(pivot) = bucket$
   (b) $creative\ replacement(plank) = surfboard$
5. Embed in affordances or used affordances:
   (a) $embed(bucket) \rightarrow show(in(bucket, sand), near(bucket, toy\ spade))$[5]
   (b) $embed(surfboard) \rightarrow show(on(water, surfboard)), on(surfboard, surfer))$
6. Addition of red herring objects:
   $sim(measure\ weight, measure\ volume) \wedge measure\ volume \notin PT_{sol} \wedge$
   $measure\ volume \in PT_{Archimedes} \wedge bathtub \in PT_{Archimedes} \wedge$
   $sim(bathtub, pool\ with\ water) \rightarrow show(pool\ with\ water)$

An extra point can be made about embedding the entire problem in a new contextual setting. The problem template of balance is in this case creatively transformed to a template about seesaws, and then imported in the contextual setting of a beach. The beach location could have been chosen as a consequence of having already picked one of the two replacement objects for the seesaw parts – the surfboard. Laying this object in its own contextual setting made the natural choice a beach, and the replacement for the second object, which did not have that many constraining properties, could be fixed to another object in the same setting – the bucket.

5 Discussion – towards a computational approach

As shown in the above test cases: (i) insight problems in the practical uses of objects domain can be analysed from the cognitive perspective of re-representation, and creative inference, (ii) such principles can be put together in an approach towards creating more insight problems and (iii) the approach can be used to create more insight problems in the specified domain.

[5] The PT of using a bucket to dig and play with sand is supported by the neighborhood of objects such a spade, a sand castle, etc.

This approach might not reflect a reverse engineering of all the types of insight processes, or might not yet result in all the types of problems that can be created in the domain. However, starting from creating some such problems and evaluating them with human participants will have the impact of understanding and controlling the process of creating insight problems more thoroughly, and thus, in the future, providing a wider database of problems, based perhaps on a wider array of processes.

Creating insight problems might seem like a lofty computational pursuit. However, as shown above, the processes implied by this approach are substantial enough to allow for formalization. An interesting next step would be to tackle the issue using computational assistance when creating such problems. Part of the tools needed for such an approach already exist, at least in prototype form.

Take item (ii) of the list of techniques provided by the approach here – hiding objects in a different form – by replacing them with objects which have similar properties and affordances (but for which solution related affordances might not be as salient as for the initial objects). The creative object replacement (OR) part of OROC [Olteţeanu and Falomir, 2016] can be used to generate items from the practical objects domain which have similar affordances as the initial items. OROC makes creative inferences about new affordances of known objects, based on the similarity between said objects to other objects on shape and material properties. The object composition (OC) part of the same system can in part take care of items (iii) and (iv) on the list, specifically by decomposing various objects which are part of the solution in object parts, then finding similar objects with similar properties (OR) in its knowledge base; these objects, or the ones they are part of, can be used to substitute initial objects which are a salient part of the solution.

Other parts of this approach, like (i), (v) and creative transfer of a simple problem can be based on OROC knowledge, but also require knowledge of problem templates. Such knowledge should include contexts of affordances in which various objects get engaged, functional subsets of objects which are employed in such templates, qualitative or quantitative measures of similarity of template affordance, and some measure for when templates achieve similar but not quite the same results – thus interfering with the human judgement because of the similarity component, but not being able to help participants solve the problem.

In conclusion, a few case studies of classical insight problems have been analysed, in order to extract a set of principles which can be used in the creation of new insight problems. A non-exhaustive approach towards mechanisms that can be used to create such problems, and that is amenable to computational implementation has been proposed. Then, the applicability of the approach has been analysed, by describing it in the context of two newly created insight problems. Some existing tools that can be used to assist with this process have been briefly discussed. As future work, we plan to start (i) implementing this approach and/or relying on computational assistance, and (ii) experiment with problem template acquisition, problem template transfer and creative replacement of problem templates.

References

[Boden, 2003] Boden, M. (2003). *The Creative Mind: Myths and Mechanisms*. Routledge.

[Dow and Mayer, 2004] Dow, G. T. and Mayer, R. E. (2004). Teaching students to solve insight problems: Evidence for domain specificity in creativity training. *Creativity Research Journal*, 16(4):389–398.

[Duncker, 1945] Duncker, K. (1945). On problem solving. *Psychological Monographs*, 58(5, Whole No.270).

[Guilford et al., 1978] Guilford, J., Christensen, P., Merrifield, P., and Wilson, R. (1978). Alternate uses: Manual of instructions and interpretation. *Orange, CA: Sheridan Psychological Services*.

[Maier, 1931] Maier, N. R. (1931). Reasoning in humans. II. The solution of a problem and its appearance in consciousness. *Journal of comparative Psychology*, 12(2):181.

[Olteţeanu, 2014] Olteţeanu, A.-M. (2014). Two general classes in creative problem-solving? An account based on the cognitive processes involved in the problem structure - representation structure relationship. In Besold, T., Kühnberger, K.-U., Schorlemmer, M., and Smaill, A., editors, *Proceedings of the Workshop on Computational Creativity, Concept Invention, and General Intelligence*, volume 01-2014 of *Publications of the Institute of Cognitive Science*. Institute of Cognitive Science Osnabrück.

[Olteţeanu and Falomir, 2016] Olteţeanu, A.-M. and Falomir, Z. (2016). Object replacement and object composition in a creative cognitive system. Towards a computational solver of the Alternative Uses Test. *Cognitive Systems Research*, 39:15–32.

[Olteţeanu, 2016] Olteţeanu, A.-M. (2016). From simple machines to Eureka in four not-so-easy steps. Towards creative visuospatial intelligence. In Müller, V., editor, *Fundamental Issues of Artificial Intelligence*, volume 376 of *Synthese Library*, pages 159–180. Springer.

Coherent Concept Invention

Marco Schorlemmer, Roberto Confalonieri, and Enric Plaza

Artificial Intelligence Research Institute, IIIA-CSIC
Bellaterra (Barcelona), Catalonia, Spain
{marco,confalonieri,enric}@iiia.csic.es

Abstract. We address the problem on how newly invented concepts are evaluated with respect to a background ontology of conceptual knowledge so as to decide which of them are to be accepted into a system of familiar concepts, and how this, in turn, may affect the previously accepted conceptualisation. As technique to tackle this problem we explore the applicability of Paul Thagard's computational theory of coherence. In particular, we propose a formalisation of Thagard's notion of *conceptual coherence* for concepts represented in the $\mathcal{AL}$ description logic and explore by means of an illustrative example the role coherence may play in the process of conceptual blending.

Keywords: conceptual blending, coherence, description logics

1 Introduction

Combinational creativity —when novel ideas (concepts, theories, solutions, works of art) are produced through unfamiliar combinations of familiar ideas— is, of the three forms of creativity put forward by Boden, the most difficult to capture computationally [2]. Putting concepts together to generate new concepts is, in principle, not a difficult task; but doing this in a computationally tractable way, and being able to recognise the value of newly invented concepts for better understanding a certain domain, is not as straightforward.

An important development that has significantly influenced the current understanding of the general cognitive principles operating during concept invention is Fauconnier and Turner's theory of *conceptual blending* [6, 7]. Fauconnier and Turner proposed conceptual blending as the fundamental cognitive operation underlying much of everyday thought and language, and modelled it as a process by which humans subconsciously combine particular elements and their relations of originally separate conceptual spaces into a unified space, in which new elements and relations emerge, and new inferences can be drawn.

The theory has been primarily applied as an analytic tool for describing already existing blends of ideas and concepts in a varied number of fields, such as linguistics, music theory, poetics, mathematics, theory of art, political science, discourse analysis, philosophy, anthropology, and the study of gesture and of material culture [20]. But it has been also widely recognised to be a theory that can serve as a basis for computational models of creativity [5, 9, 10, 15, 21].

To guide the concept invention process, in addition to the blending mechanism *per se*, at least two additional dimensions need to be considered, namely the origin and destination of concept invention, i.e., from where (and how) input concepts are selected and to whom the concept invention is headed. Confalonieri et al. have proposed a process model for concept invention in which these dimensions are taken into account [3]. Inputs are selected based on a similarity measure that is computed relative to a Rich Background, and blends are evaluated using an argumentation framework based on value preferences of the audience for which concepts are invented.

In this paper, we aim at showing how Thagard's computational theory of coherence [19] could also serve as an additional mechanism for triggering concept invention and evaluating newly blended concepts. In [18], Thagard suggested to use coherence as a model for the closely related cognitive process of *conceptual combination*, where the focus is primarily on language compositionality such as noun-noun or adjective-noun combinations [17]. Kunda and Thagard, for instance, show how conceptual coherence can be used for describing how we reason with social stereotypes [12].

Building upon Thagard's intuitions and principles for modelling coherence, we propose a formalisation of Thagard's notion of conceptual coherence for concepts represented in a description logic —we take the basic description logic $\mathcal{AL}$ as a start— and further explore its applicability to conceptual blending. But instead of interpreting coherence or incoherence based on statistical correlations or causal relations (i.e., on frequencies of positive or negative association), we determine coherence and incoherence as dependent on how concept descriptions are stated. Failure to find conceptual blends that cohere with some given background knowledge leads to a search for alternative conceptual blends that eventually increase the overall coherence of the blend with the background knowledge.

The paper is organised as follows: In Section 2 we give a brief overview of Thagard's computational theory of coherence, in Section 3 we introduce some core definitions regarding coherence and coherence graphs, and in Section 4 we provide a formalisation of conceptual coherence for the description logic $\mathcal{AL}$. Conceptual blending in $\mathcal{AL}$ is described in Section 5, and coherence is applied to blending in Section 6. We conclude in Section 7.

2 Thagard's Computational Theory of Coherence

Thagard addresses the problem of determining which pieces of information, such as hypotheses, beliefs, propositions or concepts, to accept and which to reject based on how they cohere and incohere among them, given that, when two elements cohere, they tend to be accepted together or rejected together; and when two elements incohere, one tends to be accepted while the other tends to be rejected [19].

This can be reformulated as a constraint satisfaction problem as follows. Pairs of elements that cohere between them form positive constraints, and pairs of elements that incohere between them form negative constraints. If we partition

the set of pieces of information we are dealing with into a set of accepted elements and a set of rejected elements, then a positive constraint is satisfied if both elements of the constraint are either among the accepted elements or among the rejected ones; and a negative constraint is satisfied if one element of the constraint is among the accepted ones and the other is among the rejected ones. The coherence problem is to find the partition that maximises the number of satisfied constraints.

Note that in general we may not be able to partition a set of elements as to satisfy *all* constraints, thus ending up accepting elements that incohere between them or rejecting an element that coheres with an accepted one. The objective is to minimise these undesired cases. The coherence problem is known to be NP-complete, though there exist algorithms that find good enough solutions of the coherence problem while remaining fairly efficient.

Depending on the kind of pieces of information we start from, and on the way the coherence and incoherence between these pieces of information is determined, we will be dealing with different kinds of coherence problems. So, in *explanatory coherence* we seek to determine the acceptance or rejection of hypotheses based on how they cohere and incohere with given evidence or with competing hypotheses; in *deductive coherence* we seek to determine the acceptance of rejection of beliefs based on how they cohere and incohere due to deductive entailment or contradiction; in *analogical coherence* we seek to determine the acceptance or rejection of mapping hypotheses based on how they cohere or incohere in terms of structure; and in *conceptual coherence* we seek to determine the acceptance or rejection of concepts based on how they cohere or incohere as the result of the positive or negative associations that can be established between them. Thagard discusses these and other kinds of coherence.

Although Thagard provides a clear technical description of the coherence problem as a constraint satisfaction problem, and he enumerates concrete principles that characterise different kinds of coherences, such as those discussed later in Section 4 for conceptual coherence, he does not clarify the actual nature of the coherence and incoherence relations that arise between pieces of information, nor does he suggest a precise formalisation of the principles he discusses. Joseph et al. have proposed a concrete formalisation and realisation of deductive coherence [11], which they applied to tackle the problem of norm adoption in normative multi-agent systems. In this paper, we shall focus on the problem of conceptual coherence and its applicability to conceptual blending.

3 Preliminaries: Coherence Graphs

In this section we give precise definitions of the concepts intuitively introduced in the previous section.

Definition 1. *A* coherence graph *is an edge-weighted, undirected graph* $G = \langle V, E, w \rangle$*, where:*

1. *V is a finite set of nodes representing pieces of information.*

2. $E \subseteq V^{(2)}$ *(where* $V^{(2)} = \{\{u, v\} \mid u, v \in V\}$*) is a finite set of edges representing the coherence or incoherence between pieces of information.*
3. $w : E \to [-1, 1] \setminus \{0\}$ *is an edge-weighted function that assigns a value to the coherence between pieces of information.*

Edges of coherence graphs are also called constraints.

When we partition the set V of vertices of a coherence graph (i.e., the set of pieces of information) into a set A of accepted elements and a set $R = V \setminus A$ of rejected elements, then we can say when a constraint —an edge between vertices— is satisfied or not by the partition.

Definition 2. *Given a coherence graph* $G = \langle V, E, w \rangle$*, and a partition* (A, R) *of* V*, the set of* satisfied constraints $C_{(A,R)} \subseteq E$ *is given by:*

$$C_{(A,R)} = \left\{ \{u, v\} \in E \;\middle|\; \begin{array}{l} u \in A \text{ iff } v \in A, \text{ whenever } w(\{u, v\}) > 0 \\ u \in A \text{ iff } v \in R, \text{ whenever } w(\{u, v\}) < 0 \end{array} \right\}$$

All other constraints (i.e., those in $E \setminus C_{(A,R)}$*) are said to be* unsatisfied.

The coherence problem is to find the partition of vertices that satisfies as much constraints as possible, i.e., to find the partition that maximises the coherence value as defined as follows, which makes coherence to be independent of the size of the coherence graph.

Definition 3. *Given a coherence graph* $G = \langle V, E, w \rangle$*, the* coherence of a partition (A, R) *of* V *is given by*

$$\kappa(G, (A, R)) = \frac{\sum_{\{u,v\} \in C_{(A,R)}} |w(\{u, v\})|}{|E|}$$

Notice that there may not exist a unique partition with a maximum coherence value. Actually, at least two partitions have the same coherence value, since $\kappa(G, (A, R)) = \kappa(G, (R, A))$ for any partition (A, R) of V.

4 Conceptual Coherence in Description Logics

Thagard characterises conceptual coherence with these principles [19]:

Symmetry: Conceptual coherence is a symmetric relation between pairs of concepts.

Association: A concept coheres with another concept if they are positively associated, i.e., if there are objects to which they both apply.

Given Concepts: The applicability of a concept to an object may be given perceptually or by some other reliable source.

Negative Association: A concept incoheres with another concept if they are negatively associated, i.e., if an object falling under one concept tends not to fall under the other concept.

Acceptance: The applicability of a concept to an object depends on the applicability of other concepts.

To provide a precise account of these principles we shall formalise *Association* and *Negative Association* between concepts expressed in a description logic, since these are the principles defining coherence and incoherence. We shall assume coherence between two concept descriptions when we have explicitly stated that one subsumes the other ("there are objects to which both apply"); and we shall assume incoherence when we have explicitly stated that they are disjoint ("an object falling under one concept tends not to fall under the other concept").

Definition 4. *Given a Tbox $\mathcal{T}$ in description logic $\mathcal{AL}$ and a pair of concept descriptions $C, D \notin \{\top, \bot\}$, we will say that:*

- *C coheres with D, if $C \sqsubseteq D \in \mathcal{T}$, and that*
- *C incoheres with D, if $C \sqsubseteq \neg D \in \mathcal{T}$ or $C \sqcap D \sqsubseteq \bot \in \mathcal{T}$.*

In addition, coherence and incoherence between concept descriptions depend on the concept constructors used, and we will say that, for all atomic concepts A, atomic roles R, and concept descriptions $C, D \notin \{\top, \bot\}$:

- *$\neg A$ incoheres with A;*
- *$C \sqcap D$ coheres both with C and with D;*
- *$\forall R.C$ coheres (or incoheres) with $\forall R.D$, if C coheres (or incoheres) with D.*[1]

Symmetry follows from the definition above, and *Acceptance* is captured by the aim of maximising coherence in a coherence graph. For this we need to define how a TBox determines a coherence graph, and, in order to keep the graph finite, we express coherence and incoherence only between non-trivial concept descriptions (i.e., excluding $\top$ and $\bot$) that are explicitly stated in the TBox.

Definition 5. *Let $\mathcal{T}$ be a TBox in $\mathcal{AL}$. The set of* non-trivial subconcepts *of $\mathcal{T}$ is given as*

$$\mathit{sub}(\mathcal{T}) = \bigcup_{C \sqsubseteq D \in \mathcal{T}} \mathit{sub}(C) \cup \mathit{sub}(D)$$

where sub is defined over the structure of concept descriptions as follows:

$$\begin{aligned}
\mathit{sub}(A) &= \{A\} \\
\mathit{sub}(\bot) &= \emptyset \\
\mathit{sub}(\top) &= \emptyset \\
\mathit{sub}(\neg A) &= \{\neg A, A\} \\
\mathit{sub}(C \sqcap D) &= \{C \sqcap D\} \cup \mathit{sub}(C) \cup \mathit{sub}(D) \\
\mathit{sub}(\forall R.C) &= \{\forall R.C\} \cup \mathit{sub}(C) \\
\mathit{sub}(\exists R.\top) &= \{\exists R.\top\}
\end{aligned}$$

[1] Note that since $\mathcal{AL}$ allows only for limited existential quantification we cannot provide a general rule for coherence between concept descriptions of the form $\exists R.\top$.

Definition 6. *The* coherence graph of a TBox $\mathcal{T}$ *is the edge-weighted, undirected graph* $G = \langle V, E, w \rangle$ *whose vertices are non-trivial subconcepts of* $\mathcal{T}$ *(i.e.,* $V = \mathsf{sub}(\mathcal{T})$*), whose edges link subconcepts that either cohere or incohere according to Definition 4, and whose edge-weight function* w *is given as follows:*

$$w(\{C, D\}) = \begin{cases} 1 & \text{if } C \text{ and } D \text{ cohere} \\ -1 & \text{if } C \text{ and } D \text{ incohere} \end{cases}$$

5 Conceptual Blending in $\mathcal{AL}$

We follow the modelling principles and techniques of [4], where the process of conceptual blending is characterised by the notion of amalgams [1, 14]. According to this approach, the process of conceptual blending can be described as follows:

1. We take a taxonomy of concepts described in a background ontology expressed as a Tbox $\mathcal{T}$.
2. A mental space of an atomic concept A is modelled, for the purpose of conceptual blending, by means of a subsumption $A \sqsubseteq C$ specifying the necessary conditions we are focusing on.
3. The new concept to be invented is represented by the concept description that conjoins the atomic concepts to be blended.
4. With amalgams we generalise the input spaces based on the taxonomy in our TBox until a satisfactory blend is generated.

Formally, the notion of amalgams can be defined in any representation language $\mathcal{L}$ for which a subsumption relation between formulas (or descriptions) of $\mathcal{L}$ can be defined, and therefore also in the set of all $\mathcal{AL}$ concept descriptions with the subsumption relation $\sqsubseteq_{\mathcal{T}}$.

To formally specify an amalgam we first need to introduce some notions. Let N_C be a set of concept names, N_R be a set of role names, and $\mathcal{L}(\mathcal{T})$ be the finite set of all $\mathcal{AL}$ concept descriptions that can be formed with the concept and role names occurring in an $\mathcal{AL}$ TBox $\mathcal{T}$. Then:

Definition 7. *Given two descriptions* $C_1, C_2 \in \mathcal{L}(\mathcal{T})$*:*

- *A* most general specialisation *(MGS) is a description* C_{mgs} *such that* $C_{mgs} \sqsubseteq_{\mathcal{T}} C_1$ *and* $C_{mgs} \sqsubseteq_{\mathcal{T}} C_2$ *and for any other description* D *such that* $D \sqsubseteq_{\mathcal{T}} C_1$ *and* $D \sqsubseteq_{\mathcal{T}} C_2$*, then* $D \sqsubseteq_{\mathcal{T}} C_{mgs}$*.*
- *A* least general generalisation *(LGG) is a description* C_{lgg} *such that* $C_1 \sqsubseteq_{\mathcal{T}} C_{lgg}$ *and* $C_2 \sqsubseteq_{\mathcal{T}} C_{lgg}$ *and for any other description* D *such that* $C_1 \sqsubseteq_{\mathcal{T}} D$ *and* $C_2 \sqsubseteq_{\mathcal{T}} D$*, then* $C_{lgg} \sqsubseteq_{\mathcal{T}} D$*.*

Intuitively, an MGS is a description that has some of the information from both original descriptions C_1 and C_2, while an LGG contains what is common to them.

An *amalgam* or *blend* of two descriptions is a new description that contains *parts from these original descriptions* and it can be formally defined as follows.

Definition 8 (Amalgam). *Let $\mathcal{T}$ be an $\mathcal{AL}$ TBox. A description $C_{am} \in \mathcal{L}(\mathcal{T})$ is an* amalgam *of two descriptions C_1 and C_2 (with LGG C_{lgg}) if there exist two descriptions $\overline{C_1}$ and $\overline{C_2}$ such that: $C_1 \sqsubseteq_{\mathcal{T}} \overline{C_1} \sqsubseteq_{\mathcal{T}} C_{lgg}$, $C_2 \sqsubseteq_{\mathcal{T}} \overline{C_2} \sqsubseteq_{\mathcal{T}} C_{lgg}$, and C_{am} is an MGS of $\overline{C_1}$ and $\overline{C_2}$.*

The number of blends that satisfies the above definition can be very large and selection criteria for filtering and ordering them are therefore needed. Fauconnier and Turner discussed optimality principles [7], however, these principles are difficult to capture in a computational way, and other selection strategies need to be explored. Since we use a logical theory such as $\mathcal{AL}$, one way to evaluate a blend is consistency checking. Another alternative, that we will investigate in this paper, is to evaluate blends in terms of conceptual coherence.

The LGG and the generalised descriptions, needed to compute the amalgam as defined above, are obtained by means of a generalisation refinement operator that allows us to find generalisations of $\mathcal{AL}$ concept descriptions.

5.1 Generalising $\mathcal{AL}$ descriptions

Roughly speaking, a generalisation operator takes a concept C as input and returns a set of descriptions that are more general than C by taking a Tbox $\mathcal{T}$ into account.

In order to define a generalisation refinement operator for $\mathcal{AL}$, we define the upward cover set of atomic concepts. In the following definition, $\mathsf{sub}(\mathcal{T})$ (Definition 5) guarantees the following upward cover set to be finite.

Definition 9. *Let $\mathcal{T}$ be an $\mathcal{AL}$ TBox with concept names from N_C. The* upward cover set *of an atomic concept $A \in N_C \cup \{\top, \bot\}$ with respect to $\mathcal{T}$ is given as:*

$$\begin{aligned} \mathsf{UpCov}(A) := \{C \in \mathsf{sub}(\mathcal{T}) \cup \{\top, \bot\} \mid A \sqsubseteq_{\mathcal{T}} C \\ \text{and there is no } C' \in \mathsf{sub}(\mathcal{T}) \cup \{\top, \bot\} \\ \text{such that } A \sqsubset_{\mathcal{T}} C' \sqsubset_{\mathcal{T}} C\} \end{aligned} \tag{1}$$

We can now define our generalisation refinement operator for $\mathcal{AL}$ as follows.

Definition 10. *Let $\mathcal{T}$ be an $\mathcal{AL}$ TBox. We define the* generalisation refinement operator γ *inductively over the structure of concept descriptions as follows:*

$$\begin{aligned} \gamma(A) &= \mathsf{UpCov}(A) \\ \gamma(\top) &= \mathsf{UpCov}(\top) = \emptyset \\ \gamma(\bot) &= \mathsf{UpCov}(\bot) \\ \gamma(C \sqcap D) &= \{C' \sqcap D \mid C' \in \gamma(C)\} \cup \{C \sqcap D' \mid D' \in \gamma(D)\} \cup \{C, D\} \\ \gamma(\forall r.C) &= \begin{cases} \{\forall r.C' \mid C' \in \gamma(C)\} & \text{whenever } \gamma(C) \neq \emptyset \\ \{\top\} & \text{otherwise.} \end{cases} \\ \gamma(\exists r.\top) &= \emptyset \end{aligned}$$

$\mathsf{House} \sqsubseteq \mathsf{Object}$	$\mathsf{Resident} \sqsubseteq \mathsf{Person}$
$\mathsf{Boat} \sqsubseteq \mathsf{Object}$	$\mathsf{Passenger} \sqsubseteq \mathsf{Person}$
$\mathsf{Land} \sqsubseteq \mathsf{Medium}$	$\mathsf{Person} \sqcap \mathsf{Medium} \sqsubseteq \bot$
$\mathsf{Water} \sqsubseteq \mathsf{Medium}$	$\mathsf{Object} \sqcap \mathsf{Medium} \sqsubseteq \bot$
$\mathsf{Water} \sqcap \mathsf{Land} \sqsubseteq \bot$	$\mathsf{Object} \sqcap \mathsf{Person} \sqsubseteq \bot$

Fig. 1. The background ontology of the House and Boat.

We should notice at this point that γ can return concept descriptions that are equivalent to the concept being generalised. One possible way to avoid this situation is to discard these generalisations [4]. Given a generalisation refinement operator γ, $\mathcal{AL}$ concepts are related by refinement paths as described next.

Definition 11. *A finite sequence $C_1, \ldots, C_n$ of $\mathcal{AL}$ concepts is a* concept refinement path $C_1 \xrightarrow{\gamma} C_n$ *from C_1 to C_n of the generalisation refinement operator γ iff $C_{i+1} \in \gamma(C_i)$ for all $i : 1 \leq i < n$. $\gamma^*(C)$ denotes the set of all concepts that can be reached from C by means of γ in a finite number of steps.*

The repetitive application of the generalisation refinement operator allows us to find a description that represents the properties that two or more $\mathcal{AL}$ concepts have in common. This description is a common generalisation of $\mathcal{AL}$ concepts, the so-called *generic space* that is used in conceptual blending.

Definition 12. *An $\mathcal{AL}$ concept description G is a generic space of the $\mathcal{AL}$ concept descriptions $C_1, \ldots, C_n$ if and only if $G \in \gamma'^*(C_i)$ for all $i = 1, \ldots, n$.*

5.2 An Example: The House-Boat Blend

The process of conceptual blending in terms of amalgams can be illustrated by means of a typical blend example: the *house-boat* [7, 8]. The precise formalisation is not critical at this point, different ones exist [9, 15], but all provide similar distinctions.

The $\mathcal{AL}$ theories for House and Boat introduce the axioms modelling the mental spaces for *house* and *boat*.

$$\begin{aligned} \mathsf{House} &\sqsubseteq \forall \mathsf{usedBy}.\mathsf{Resident} \sqcap \forall \mathsf{on}.\mathsf{Land} \\ \mathsf{Boat} &\sqsubseteq \forall \mathsf{usedBy}.\mathsf{Passenger} \sqcap \forall \mathsf{on}.\mathsf{Water} \end{aligned}$$

The House and Boat theories cannot be directly blended since they generate an inconsistency. This is due to the background ontology stating that the medium on which an object is situated cannot be *land* and *water* at the same time (Figure 1). Therefore, some parts of the House and Boat descriptions need to be generalised in a controlled manner before these concepts can be blended. The generic space between a house and a boat—an object that is on a *medium* and *used-by* a *person*—is a lower bound in the space of generalisations that need to be explored in order to generalise these concepts and to blend them into a *house-boat*. The generic space is obtained according to Definition 12 by applying the refinement operator γ.

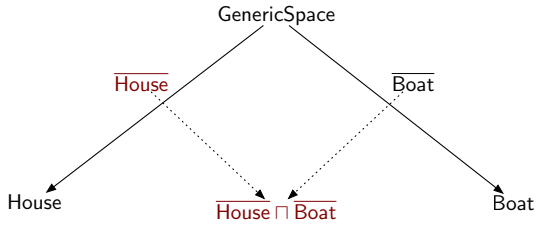


Fig. 2. A diagram of an amalgam HouseBoat from descriptions House and Boat and their respective generalisations $\overline{\mathsf{House}}$ and $\overline{\mathsf{Boat}}$. Arrows indicate the subsumption of the target by the source of the arrow.

Example 1. Let us consider the House and Boat concepts. Their generic space is: ∀usedBy.Person⊓ ∀on.Medium and is obtained as follows. In the House concept, the subconcepts ∀usedBy.Resident and ∀on.Land are generalised to ∀usedBy.Person and ∀on.Medium respectively. In the Boat concept, the subconcepts ∀usedBy. Passenger and ∀on.Water are generalised in a similar way.

From a conceptual blending point of view, the *house-boat* blend can be created when the medium on which a house is situated (land) becomes the medium on which boat is situated (water), and the resident of the house becomes the passenger of the boat. This blend can be obtained when the input concepts house and boat are generalised as follows:

$$\overline{\mathsf{House}} \sqsubseteq \forall \mathsf{usedBy.Resident} \sqcap \forall \mathsf{on.Medium}$$
$$\overline{\mathsf{Boat}} \sqsubseteq \forall \mathsf{usedBy.Person} \sqcap \forall \mathsf{on.Water}$$

The *house-boat* blend is obtained by conjoining the generalised mental spaces $\overline{\mathsf{House}}$ and $\overline{\mathsf{Boat}}$ (Figure 2). It is easy to see that $\overline{\mathsf{House}} \sqcap \overline{\mathsf{Boat}}$ is an amalgam according to Definition 8.

6 Evaluating the Coherence of Conceptual Blends

This section describes how coherence is used to evaluate blends. That is, how coherence graphs are built, and how the different coherence values are to be interpreted. The overall idea is to compute the coherence graph and maximsing partitions for each blend, and use the maximal coherence degree of the coherence graphs to rank the blends.

Let $\mathcal{T}$ be the TBox of the background ontology, let $A \sqsubseteq C$ and $B \sqsubseteq D$ be the axioms representing our mental spaces, and let $A \sqcap B$ be the new concept we would like to invent. The process of evaluating blends according to conceptual coherence can be described as follows:

1. Given the mental spaces, we generate a candidate blend according to Definition 8.

2. We form the coherence graph for $\mathcal{T} \cup \{A \sqsubseteq C, B \sqsubseteq D\}$, including node $A \sqcap B$, according to Definition 6.
3. We compute the coherence maximising partitions according to Definition 3 and we associate it to the blend.
4. We repeat this procedure for all the blends that can be generated from the mental spaces.

Once the maximising partitions are computed, the coherence of the blend could be measured in terms of the coherence value of the coherence-maximising partitions. The degree of the coherence graph directly measures how much a blend coheres with the background ontology.

Definition 13. *Let $G = \langle V, E, w \rangle$ the coherence graph of a blend B and let $\mathcal{P}$ the set of partitions of G. The maximal coherence value of B of G is* $\deg(B) = \max_{P \in \mathcal{P}} \{\kappa(G, P)\}$.

This maximal coherence value can be used to rank blends as follows.

Definition 14. *Let $\mathcal{T}$ be a TBox of a background ontology, let $A \sqsubseteq C$ and $B \sqsubseteq D$ be the axioms representing mental spaces, let $\mathcal{B}$ be the set of blends that can be generated from them. For each $b_1, b_2 \in \mathcal{B}$, we say that b_1 is preferred to b_2 ($b_1 \succeq b_2$) if and only if* $\deg(b_1) \geq \deg(b_2)$.

To exemplify how the coherence degree can be used to evaluate blends, we consider the *house-boat* example. According to the amalgams process of conceptual blending described in the previous section, several blends can be generated by blending the mental space of House and Boat. In particular, the concept $\mathsf{House} \sqcap \mathsf{Boat}$ is a valid blend.

The coherence graph blending the House and Boat directly is shown in Figure 3. As expected the concepts House and Boat positively coheres with the axioms representing the mental spaces and with the concept $\mathsf{House} \sqcap \mathsf{Boat}$, which is representing the blend. The incoherence relation between $\forall$on.Land and $\forall$on.Water is due to the fact that the concepts Water and Land incohere, since the background ontology contains the disjointness axiom $\mathsf{Water} \sqcap \mathsf{Land} \sqsubseteq \bot$. The coherence graph of House and Boat has a maximal coherence value of 0.84.

For the sake of our example, we generate new blends by generalising the axioms modelling our mental spaces. For instance, by applying the generalisations seen in the previous section that lead to the creation of the *house-boat* blend, we obtain the coherence graph in Figure 4.[2] The coherence graph of blending $\overline{\mathsf{House}}$ and $\overline{\mathsf{Boat}}$ has a maximal coherence value of 0.9. This graph yields a higher coherence degree since generalising $\forall$on.Land to $\forall$on.Medium prevents the appearance of the incoherence relation between $\forall$on.Land and $\forall$on.Water.

It is easy to see that the blend $\overline{\mathsf{House}} \sqcap \overline{\mathsf{Boat}}$ is preferred to $\mathsf{House} \sqcap \mathsf{Boat}$ since it has a maximal coherence degree that is higher.

[2] Concepts belonging to the background ontology are omitted.

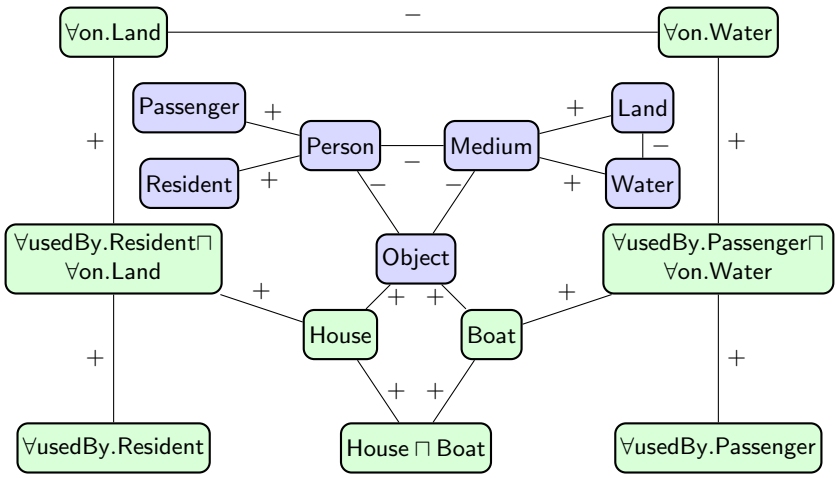


Fig. 3. The coherence graph of the House ⊓ Boat blend, showing the main concepts and their coherence relations. Blue and green coloured boxes represent concepts belonging to the background ontology and to the input mental spaces respectively.

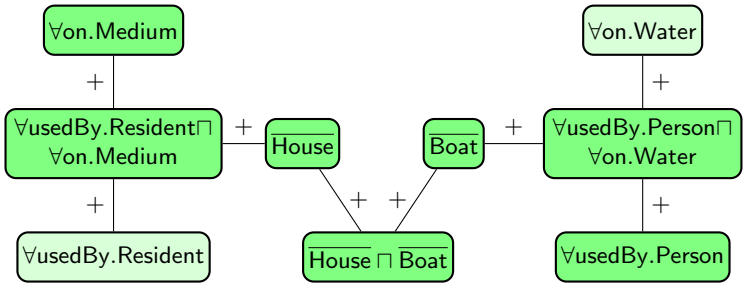


Fig. 4. The coherence graph of the $\overline{\text{House}} \sqcap \overline{\text{Boat}}$ blend, showing the main concepts and coherence relations. Generalised concepts are displayed in a darker tonality.

7 Conclusion

This paper should be seen as a first attempt to (a) provide a formal account of conceptual coherence for a particular concept representation language, and (b) to explore its applicability for guiding the process of conceptual blending.

With respect to (a), we proposed a formalisation of conceptual coherence between concept descriptions expressed in the basic $\mathcal{AL}$ description logic. This is only a starting point, and obviously this formalisation exercise should be carried out for more expressive concept representation languages. Usually, coherence and incoherence are not treated only in binary terms, but it is also natural to take certain degrees of coherence or incoherence into account. This, for instance, has also been the approach of Joseph et al. when formalising deductive coherence [11]. Although there is not an obvious way to do so with the formalisation of

conceptual coherence of $\mathcal{AL}$ proposed in this paper, we do not discard that this could be done for more expressive concept representation languages. One could imagine that description logics with number restrictions or nominals, such as $\mathcal{SROIQ}$ for instance, would allow for expressing degrees of concept overlap that could be interpreted as degrees of coherence or incoherence.

With respect to (b), we have so far only focused on how the coherence values of a graph of concept descriptions were evolving dependent on how these descriptions were changing in our amalgam-based conceptual blending process. However, we have not discussed yet another important aspect of coherence theory, namely how to interpret the two parts of a coherence-maximising partition: the set of accepted and of rejected concepts. The information that a particular concept description falls in the set of accepted concepts or in the set of rejected concepts could also be taken into account to decide the acceptance or rejection of newly invented concepts; or even of already existing concepts in the background knowledge, in the light of newly invented concepts. With the formalisation in $\mathcal{AL}$ given in this paper we could not see yet a clear way to provide such an interpretation of acceptance and rejection, but we think this aspect might become clearer as a wider range of concept representation languages is explored.

In this paper we attempted to see how coherence could be used as another tool for guiding the process of conceptual blending and for evaluating conceptual blends in the task of concept invention; an additional technique to those already proposed, such as optimality principles [16], logical consistency [13], and values of audiences [3]. We believe it is worth to further study the proper combination of these techniques and to carry out a comprehensive evaluation.

An implementation of conceptual coherence presented in this paper using the OWL API and Answer Set Programming is available at: `https://rconfalonieri@bitbucket.org/rconfalonieri/coinvent-coherence.git`.

References

[1] T. R. Besold and E. Plaza. Generalize and Blend: Concept Blending Based on Generalization, Analogy, and Amalgams. In *Proceedings of the 6th International Conference on Computational Creativity, ICCC15*, 2015.

[2] M. A. Boden. *The Creative Mind: Myths and Mechanisms*. George Weidenfeld and Nicolson Ltd., 1990.

[3] R. Confalonieri, E. Plaza, and M. Schorlemmer. A Process Model for Concept Invention. In *Proc. of the 7th International Conference on Computational Creativity, ICCC16*, 2016.

[4] R. Confalonieri, M. Schorlemmer, O. Kutz, R. Peñaloza, E. Plaza, and M. Eppe. Conceptual blending in EL++. In *Proc. of 29th Int. Workshop on Description Logics*, volume 1577 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[5] M. Eppe, R. Confalonieri, E. Maclean, M. A. Kaliakatsos-Papakostas, E. Cambouropoulos, W. M. Schorlemmer, M. Codescu, and K. Kühnberger. Computational Invention of Cadences and Chord Progressions by Conceptual Chord-Blending. In *Proc. of the 24th Int. Joint Conf. on Artificial Intelligence, IJCAI 2015*, pages 2445–2451. AAAI Press, 2015.

[6] G. Fauconnier and M. Turner. Conceptual integration networks. *Cognitive Science*, 22(2):133–187, 1998.

[7] G. Fauconnier and M. Turner. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books, New York, 2003.

[8] J. Goguen. An introduction to algebraic semiotics, with application to user interface design. In *Computation for Metaphors, Analogy, and Agents*, volume 1562 of *Lecture Notes in Computer Science*, pages 242–291. Springer, 1999.

[9)] J. A. Goguen and D. F. Harrell. Style: A computational and conceptual blending-based approach. In S. Argamon, K. Burns, and S. Dubnov, editors, *The Structure of Style*, chapter 12, pages 291–316. Springer, 2010.

[10] M. Guhe, A. Pease, A. Smaill, M. Martínez, M. Schmidt, H. Gust, K.-U. Kühnberger, and U. Krumnack. A computational account of conceptual blending in basic mathematics. *Cognitive Systems Research*, 12(3–4):249–265, 2011.

[11] S. Joseph, C. Sierra, M. Schorlemmer, and P. Dellunde. Deductive coherence and norm adoption. *Logic Journal of the IGPL*, 18(1):118–156, 2010.

[12] Z. Kunda and P. Thagard. Forming unpressions from stereotypes, traits, and behaviours: A parallel-constraint-satisfaction theory. *Psychological Review*, 103 (2):284–308, 1996.

[13] F. Neuhaus, O. Kutz, M. Codescu, and T. Mossakowski. Fabricating monsters is hard. towards the automation of conceptual blending. In *Proceedings of the Workshop "Computational Creativity, Concept Invention, and General Intelligence" 2014*, volume 01-2014 of *Publications of the Institute of Cognitive Science*, 2014

[14] S. Ontañón and E. Plaza. Amalgams: A Formal Approach for Combining Multiple Case Solutions. In *Proceedings of the International Conference on Case Base Reasoning*, volume 6176 of *Lecture Notes in Computer Science*, pages 257–271. Springer, 2010.

[15] F. C. Pereira. *Creativity and Artificial Intelligence: A Conceptual Blending Approach*. Mouton de Gruyter, 2007a.

[16] F. C. Pereira and A. Cardoso. Optimality principles for conceptual blending: A first computational approach. *AISB Journal*, 1(4), 2003.

[17] B. Ran and P. R. Duimering. Conceptual combination: Models, theories and controversies. *International Journal of Cognitive Linguistics*, 1(1):65–90, 2010.

[18] P. Thagard. Coherent and creative conceptual combinations. In *Creative thought: An investigation of conceptual structures and processes*, pages 129–141. American Psychological Association, 1997.

[19] P. Thagard. *Coherence in thought and action*. The MIT Press, 2000.

[20] M. Turner. Blending and conceptual integration. http://markturner.org/ blending.html. Last checked on June 20, 2016.

[21] T. Veale and D. O'Donoghue. Computation and blending. *Cognitive Linguistics*, 11(3/4):253–281, 2000.

Empirical Evidence of the Limits of Automatic Assessment of Fictional Ideation

A. Tapscott[1⋆], J. Gómez[1], C. León[1], J. Smailović[2], M. Žnidaršič[2], P. Gervás[1]
[1]Facultad de Informática, Universidad Complutense de Madrid
[2]Department of Knowledge Technologies, Jožef Stefan Institute

Abstract. Automatic evaluation of fictional ideation systems and their output is a topic relevant to Computational Creativity. Models and techniques have been proposed for this task, but their applicability is limited to the field of fictional ideation. In this paper we describe an evaluation procedure for fictional ideation, which compares human validation of the ideas with a number of automatically generated metrics obtained from them. We report on the observed limits of this procedure. The results suggest that, besides technical limitations, providing a stable evaluation method is fundamentally incomplete unless the full creative phenomenon is modelled, including aspects that are beyond current technical capabilities.

Keywords: Automatic evaluation, ideation, empirical study, narrative, computational creativity

1 Introduction

Evaluation of creative processes and artefacts is key to computational creativity. Explicitly reflecting on the relative value and novelty is crucial if machines are to produce content that would be *deemed creative* [6]. As such, addressing evaluation is fundamental for computational creativity that can successfully fulfill human needs.

This crucial aspect contrasts with the relative scarcity of systems explicitly generating rich evaluation of their own generated material or inner processes. Some systems arguably control the quality of their artifacts by carrying out a process that ensures a minimum relative quality, but an explicit evaluation arguably represents a qualitative advantage, both theoretical (as studied by computational creativity frameworks [29]) and practical ([4]).

Although the semantics of creativity are elusive and usually problematic, the vision that quality and novelty influence the perception of the creativity of an artifact (at least from the point of view of observation) is commonly accepted. Still, quality and novelty vary depending on the domain and context. Theoretical

⋆ Supported by the project WHIM (611560) and PROSECCO (600653), funded by the European Commission, Framework Program 7, the ICT theme, and the Future Emerging Technologies FET program.

discussion on this exists and it is seminal in the field [1, 2], while other works attempt to offer either formal or procedural techniques for evaluating creativity [18, 25, 30]. These efforts address the evaluation of creativity in generic terms, and they are of limited applicability for the evaluation of the quality of specific artifacts generated automatically. It might be the case that the assumption that there is a global definition of creativity applicable to every creative domain is not possible, but we still need more empirical evidence supporting whether this is so.

Moreover, even when working within a domain in which there is an agreed definition of characteristics assumed to play a role in creativity (let us say *quality*), addressing explicit automatic evaluation can be a costly task, even more costly than creating the generative system that is being evaluated. It is not uncommon that being able to generate appropriate artefacts is doable, while yielding an explicit, measurable evaluation is not (for instance, in images generated by evolutionary computing [15]).

This paper reports on an empirical study in which the output of an automatic ideation system is assessed by computational means. When compared to human evaluation, the conceptual and practical limits of the approach were evidenced. This led to an in-depth analysis of the challenges, which is provided in Section 5.

2 Previous Work

While all scientific exploration requires thorough evaluation of the steps taken, doing so in creativity represents a challenge. How to assess creativity itself is a commonly discussed aspect of the whole phenomena of creative generation. While most authors agree on the correlation between a number of features and the perception of creativity, there is no consensus either on what these features are or how they really correlate. Moreover, adding computers to the problem makes it even more difficult to know whether a system has been successful or not. There is still a debate on what parts should be evaluated, the influence of the programmer on the output, the very definition of creative behavior, the decision of whether to focus on the process or the artifacts (or both), and many others.

The few examples present in the literature describing actual evaluation of automatic creative systems usually focus on less ambitious, more measurable aspects. This makes these systems less useful from a general perspective, but they nonetheless provide insight on the current capabilities of computer systems to assess their own production.

There is, however, a number of proposals that try to provide guidelines to evaluate creative systems. For instance, Ritchie [24, 25] addresses the issue of evaluating when a program can be considered creative by outlining a set of empirical criteria to measure the creativity of the program in terms of its output. He makes it very clear that he is restricting his analysis to the questions of what factors are to be observed, and how these might relate to creativity, specifically stating that he does not intend to build a model of creativity. Ritchie's criteria

are defined in terms of two observable properties of the results produced by the program: *novelty* (to what extent is the produced item dissimilar to existing examples of that genre) and *quality* (to what extent is the produced item a high-quality example of that genre). To measure these aspects, two rating schemes are introduced, which rate the typicality of a given item (item is typical) and its quality (item is good). Another important issue that affects the assessment of creativity in creative programs is the concept of *inspiring set*, the set of (usually highly valued) artifacts that the programmer is guided by when designing a creative program. Ritchie's criteria are phrased in terms of: what proportion of the results rates well according to each rating scheme, ratios between various subsets of the result (defined in terms of their ratings), and whether the elements in these sets were already present or not in the inspiring set. Ritchie's criteria have been used in subsequent evaluations of creative systems output [7, 21, 8].

Pease et al. [19] discuss relevant factors to evaluating systems in terms of creativity. The proposed framework mainly takes into account input provided, output produced and process employed. Each of these categories are detailed in depth, detailing their required measures. Before detailing the measurement methods, Pease et al. provide assumptions regarding creativity, also admitting their 'somewhat arbitrary' nature. The evaluation tests proposed deal with two main aspects: how close does the test predict human evaluation of creativity and how possible and practical it is to apply the test to a system. Overall, this work suggests that the very definition of creativity is subjective and that evaluating systems in a general way is problematic.

Colton et al. [5] propose an extension of Ritchie's criteria [24] that attempts to determine the impact of the input data on the creative artifact produced by a system. This more agnostic approach attempts to obtain an objective measure by comparing the output of the system to the inspirational material used as input. This investigation attempts to discriminate systems that overfit or shuffle input data (fine-tuning) instead of producing genuine novel artifacts. Among other conclusions, the authors state that comparing creative systems might not be viable, suggesting their criteria to be used as guidelines for program construction rather than post-hoc evaluation.

The creative tripod framework, proposed by Colton [3], is built around the premise that a creative system must demonstrate skill, imagination and appreciation. These qualities are not required to be possessed by the system, but rather to be perceived as possessed by the system. This is an important remark by Colton to avoid debates around the definition of creativity. The framework also includes the programmer, the system and the consumer, however Colton is only interested in the program's behavior.

Pease and Colton [18] propose an alternative to the Turing Test to assess computational systems' creativity, the FACE (Frame, Aesthetic, Concept, Expression of concept) and IDEA (Iterative Development Execution Appreciation) model. The model includes creative acts and audiences, with relevant measures such as popularity, appeal, provocation, opinion, subversion and shock. Putting the focus on the reaction produced by the creative artifact, this model attempts

to avoid the shortcomings of the Turing Test by going further than merely assessing the capacity of a creative system to imitate human behavior. By including the audience into the model, this approach acknowledges the highly subjective nature of creativity evaluation.

SPECS [9], introduced by Jordanous as "a standardised and systematic methodology for evaluating computational creativity", represents a substantial effort to provide a standard for evaluating the creativity of a system in the field of computational creativity and address the multi-faceted and subjective nature of creativity. Its flexible nature allows SPECS to adapt to the demands of the researchers' field, applying the required demands and standards. The methodology informs researchers of their system's strength and weaknesses, providing useful feedback for achieving creative results.

2.1 Evaluation of Automatically Generated Narrative

Automatic generation of narratives has been a long-standing goal of Artificial Intelligence since its very beginning. There are a number of systems described in the literature, but the evaluation of these systems – be it its output, its creative process or whatever other aspect – is seldom found. This is most likely due to the fact that the average quality or variety of the generated stories is not really comparable to those written by most humans, not necessarily professional writers.

The Mexica system [23] includes procedures for the dynamic assessment of the novelty of a story in progress with respect to previously known stories. Novelty is considered in terms of how the stories differ in terms of the actions they include and their frequency of appearance.

In Pérez et al [22] three different characteristics are considered as relevant for measuring story novelty: sequence of actions, structure of the story, and use of characters and actions.

Peinado & Gervás [20] carried out an empirical study of how generated stories were perceived by a set of human volunteer evaluators. Human judges blindly compared one of the generated basic stories to two alternatives: one rendered directly from a stored fabula of the knowledge base and another randomly generated. Values were collected for: *linguistic quality* (how well is the text written), *coherence* (how well is the sequence of events linked), *interest* (how interesting is the topic of the story for the reader) and *originality* (how different is the story from others).

León & Gervás [11] propose a model, intended as a tool to drive automatic story generation, of how quality is evaluated in stories. This paper proposes a computational model for story evaluation in which an evaluation function receives stories and outputs a value as the rating for that story. The value for this function is computed from values assigned to: accumulation of contributions from individual events depending on the meaning of the event – aspects such as whether the reader wants to continue reading the story, or how much danger or love the reader perceives in the story –, appearance of patterns or relationships between the events of a story – aspects such as causality, humour or relative

chronology – and inference – which captures the ability to interpret stories by adding material to explain what they are told even if it is not explicitly present in the story. The evaluation function has been implemented as a rule based system.

Ware, Young et. al. [27] propose a formal model for narrative conflict with seven dimensions from various narratological sources meant to aid in distinguishing one conflict from another: participant, subject, duration, balance, directness, intensity and resolution. Their experimental results [28] suggest the model predicts these seven dimensions of narrative conflict similarly to human criteria. Their good results predicting human-perceived narrative conflict suggest a similar approach may be viable for measures related to creativity.

3 Evaluating Automatic Ideation

Original ideation is central to any creative process. Coming up with innovative ideas that potentially trigger the creation of new material is fundamental to human creativity. It is not uncommon to focus creative processes on the identification of a single, valuable idea that unlocks new paths leading to finished artifacts. Although human creative teams usually rely on pure ideation to foster creativity, there have only been a few small, ad-hoc studies of how to automate ideation until recent times. Section 3.1 describes an effort to provide a system able to produce novel ideas.

3.1 The What-If Machine

Llano et al. have recently proposed an automatic ideation system [13, 14, 12]. This computational system is designed to produce relatively valuable and novel ideas autonomously. This system, the *What-If Machine*[1], includes a module for analysing the ideas and generating narrative metrics, and a module for computing a predictive machine learning model. This model is trained against collected human evaluations of what-ifs, and is intended to learn a robust function from narrative metrics to perceived overall quality. Two main hypotheses guide the design of the What-if Machine and the presented research:

1. There is a strong correlation between the perceived *overall quality* and the perceived *narrative potential*, in the sense that if the audience perceives high narrative potential, it will also perceive a high overall quality. The overall quality is defined in terms of the analyzed response from humans (i.e. no specific model beyond what humans say about quality is assumed), and the narrative potential is assumed to be directly proportional to the amount and quality of the stories a certain what-if can trigger or inspire.
2. There is a set of computable metrics whose values correlate (directly or indirectly) with the overall quality and the narrative potential.

[1] The What-if Machine: `http://www.whim-project.eu/`.

The What-If Machine is, to the best of our knowledge, the only attempt to implement a computer system able to produce novel what-if ideas. The What-If Machine is a distributed computer system in which several modules collaborate in order to output rendered what-ifs. Five modules compose the system:

1. The **ideation module** produces, using a knowledge base, what-if ideas formalized as *mini-narratives*.
2. The mini-narratives are fed into the **narrative-based metric generation**, which generates values for a set of metrics which hypothetically have a correlation with human perception of quality. These metrics are based on narrative properties of the what-ifs.
3. The mini-narratives, now enriched with its corresponding metrics, are sent to a **crowd-sourcing evaluation module**, which applies machine learning to create and refine models for predicting overall quality against human ratings.
4. The **world view** creation, providing knowledge for what-if generation, story creation and metric computation.
5. The finished, filtered what-ifs are finally passed to a **rendering module**, which creates artifacts from the final what-ifs (stories, texts or images, for instance).

A subset of the What-If Machine (modules 1, 2 and 3) was used to generate the material for the study, which is described in detail in Section 4.

4 Study

A pilot study was performed to determine the feasibility of predicting the perceived *quality* and *narrative potential* in the artifacts created by a computable creative system. Both magnitudes have been introduced in the previous section, and in order to avoid influencing our subjects, no definition for them is provided in the questionnaires (as seen in Fig. 1). This naive approach is a result of our focus on the model and its capability to predict human assessment instead of introducing our own views or definitions. The study was conducted to obtain the human rating of perceived *quality* and *narrative potential*.

Using both measures, a machine learning process will search for correlations between some metrics (detailed in the next section) and the perceived *quality* and perceived *narrative potential*. This should allow us to determine what measures are relevant to predict human-perceived *quality* and *narrative potential* to produce what-ifs that present both qualities to human observers.

4.1 Metrics

Since we have no certainty about what metrics extracted from each what-if's mini-narrative may impact over the perceived *quality* and *narrative potential*, we focused on generating the maximum amount of computable features. The impact of these features on the perceived *quality* and *narrative potential* may be

obtained with machine learning techniques (we refer to these features as *metrics*). This approach is similar to the one used by Nowak for image classification [17] that generates a high number of arbitrary features from each image.

A mini-narrative is a structure that contains a set of **narrative points** linked to schemas like *setting* or *resolution*. Each **narrative point** is a set of **narrative statements** that provide information about characters or events through predicates (e.g. *dog is old* or *dog learns to play a piano*). **Narrative statements** may be related to one another (*caused by* or *inferred by* another statement).

The next list includes the set of implemented features along with their description:

- **Length**: mini-narrative **narrative points** amount.
- **SettingQuality**: Amount of schemas divided by 3.
- **ExplicitFact**: the amount of **narrative statements** in the mini-narrative.
- **RatioCharacters**: the character/statement ratio.
- **Originality**: hits returned by the full text of the mini-narrative in the *Bing* search engine.
- **OriginalityAccurate**: hits returned by the **exact** full text of the mini-narrative in the *Bing* search engine.
- **Divergence**: average hits returned by the mini-narrative statements in the *Bing* search engine.
- **DivergenceMinimum**: minimum hits returned by the mini-narrative statements in the *Bing* search engine.
- **Evolution**: amount of `learnTo` predicates found in the mini-narrative.
- **Handicap**: amount of negated `capableOf` predicates found in the mini-narrative.
- **InterestingLife**: amount of negated `doesFor` predicates found in the mini-narrative.
- **TotalStoriesGenerated**: amount of stories generated by the story generator from the current mini-narrative.
- **StoryCharacters**: average number of characters in the generated stories.
- **Names**: StanfordNLP [16] queries for the what-if's names.
- **NamesRatio**: *Names*/*ExplicitFact* ratio.
- **Valence**: Sum per statement, each statement codified as +1 if a fact is positive, -1 if negative and 0 otherwise).
- **ValenceAverage**: *Valence*/*ExplicitFact* ratio.
- **JointWordsProbability**: joint probability average for each set of words using *ngrams*. For this metric we use the Project Oxford[2] services.
- **JointWordsProbabilityMinimum**: the minimum joint probability for the set of words using *ngrams* from Project Oxford.
- **RealityDistortionRatio**: events in the mini-narrative that negate a fact from the *knowledge base* are considered a *reality distortion*. This metric provides the *reality distortion* amount/*ExplicitFact* ratio.

[2] `https://www.projectoxford.ai/`

- **FictionalAdditionsRatio**: any event in the mini-narrative that is missing from the *knowledge base* is considered a *fictional addition*. This metric provides the *fictional addition* amount/*ExplicitFact* ratio.
- **FictionalRatio**: *reality distortion* amount plus *fictional addition* amount/*ExplicitFact*.
- **ResolutionTriggerRatio**: *resolution events* solve *conflicts* from the mini-narrative. Provides the *resolution event* amount/*ExplicitFact* ratio.
- **MainCharacterEventsRatio**: *protagonist statements* are statements in which this actor plays any role. This metric provides the *protagonist statement* amount/*ExplicitFact* ratio.

4.2 Methodology

A set of 890 what-ifs were generated by the What-If Machine. All of their source mini-narratives were processed by the metric generation system. A total of 15 different questionnaires were created, each including 10 what-ifs rendered as text from the original set of 890. 150 what-ifs were included in the evaluation set. 101 volunteers received a link that randomly redirects to one of the 15 possible questionnaires through email. Given the simplicity of the questions, Google Forms was our platform of choice. The platform was robust and stable and all of the answers were successfully stored in a Google Sheet document automatically. There was no active supervision for each subject given the remote nature and limitations of the Google Forms platform.

4.3 Questionnaire

The questionnaire informed subjects about their participation in a study related to computer-generated content (Figure 1). Some demographic information was queried (age, gender and English level) and then they were asked to evaluate the overall quality (on a 0-5 Likert scale) of each what-ifs plus its narrative potential (yes/no binary answer). A text box accepting any comment was also provided in order to gather additional qualitative information.

> You are about to evaluate some of the preliminary results of the "WHIM: The What-If Machine" research project from the European Union. The overall objective of the What-If Machine is to automatically generate fictional ideas with cultural value. You will be presented a number of what-if style ideas and we kindly ask you to rate them according to the following features:
> – Overall quality: from 0 (no quality) to 5 (superb quality). – Narrative potential (yes/no). – Any observation you can provide.
> Completing the questionnaire should not take more than 10 minutes. We really appreciate your contribution to the project.

Fig. 1. Information presented to the user in the evaluation questionnaire.

4.4 Results

101 subjects participated in the study. Statistical analysis of the results revealed no significant differences between evaluators in terms of English level, age or gender. For instance, the quality (Q) for gender yielded $\mu(Q)_{male} = 2.66$, $\sigma(Q)_{male} = 0.75$; $\mu(Q)_{female} = 2.69$, $\sigma(Q)_{female} = 0.89$. The corresponding results for English and age are comparable.

Questionnaires provided 1,007 *Quality* and 1,004 *Narrative Potential* rankings for the 150 *What-Ifs* used. *What-Ifs* were ranked between 1 and 27 times. For the *Narrative Potential* (P) measurements, we mapped "Yes" to +1, "Not sure" to 0, and "No" to -1. Overall measures resulted in $\mu(Q) = 2,4$ and $\sigma(Q) = 1,3$ for *Quality* and $\mu(P) = -0,05$ and $\sigma(P) = 0,89$ for *Narrative Potential*. Individual *What-Ifs* aggregated ranking values were used for calculating:

- Pairwise correlations between perceived *Quality* and perceived *Narrative Potential*, perceived *Quality* or perceived *Narrative Potential* and the metrics, and between individual metrics.
- Global measure of attribute importance for these metrics in predictive modeling of the average perceived *Quality* or perceived *Narrative Potential*.

Pairwise correlations Metrics that provided the same values for all *What-Ifs* in the dataset were discarded. Correlation coefficients were calculated with the Pearson Product-Moment. There is a strong positive correlation between *Quality* and *Narrative Potential* averages (0.83) and medians (0.758). As seen in table 1, both measures correlate positively with some metrics, such as *MainCharacterEventsRatio* and *RatioCharacters* and correlate negatively with others, such as *ExplicitFact* and *Length*.

Table 1. The correlation coefficient between average/median *Quality (Q)* or *Narrative Potential (P)* labels and the metrics. The values are sorted by correlation coefficient values of the average *Quality*.

	Avg Q	Mdn Q	Avg P	Mdn P
MainCharEventsRatio	0.371	0.346	0.379	0.329
RatioCharacters	0.354	0.296	0.368	0.307
ResolutionTriggerRatio	0.342	0.303	0.305	0.261
TotalStoriesGenerated	0.312	0.250	0.321	0.264
JointWordsProbMin	0.308	0.289	0.367	0.314
...	...	...	...	...
ValenceAverage	-0.219	-0.188	-0.296	-0.249
ValenceSum	-0.258	-0.234	-0.323	-0.276
StoryCharacters	-0.283	-0.269	-0.327	-0.285
ExplicitFact	-0.379	-0.336	-0.406	-0.345
Length	-0.379	-0.336	-0.406	-0.345

Importance for Predictive Modeling In order to determine the importance of each metric in predicting perceived *Quality* and *Narrative Potential* we used the Relief measure [10, 26], which is a method commonly used for feature selection in machine learning. This measure does not assume independence among the metrics, but takes their possible interdependence into account. The more the Relief scores are positive, the more a metric contributes to prediction of a target value (in our case, the value of average Quality or the average Potential). The ones that scored close to zero or negative are irrelevant and those with negative values have even a negative impact.

According to the results in Table 2 it seems that most of the metrics have no use in predictive models of average Quality. For the average Narrative Potential, however, most of the metrics seem to be slightly informative . According to Relief ranks for the metrics results, usefulness of the metrics for average Quality is to some extent inversely proportional to their usefulness for the average Narrative Potential. The absolute values of the Relief scores depend on the characteristics of data and the parameters of the assessment, which makes it difficult to use absolute thresholds for judgements on the relevance of features. However, a strong correlation among the Quality and Narrative Potential values and a mismatch of the Relief scores of metrics for these two targets provide an indication that also the contributions of the positively scored metrics are likely to be too low to be considered relevant.

Table 2. Relief measure results for average Quality (Relief Avg Q) and average Narrative Potential (Relief Avg P). Rows sorted by Relief Avg Q. The best three results are in bold and the worst three are in italics.

Metric	Relief Avg Q	Relief Avg P
Handicap	**0.027**	*-0.009*
MainCharacterEventsRatio	**0.007**	0.004
NamesRatio	**0.001**	0.006
DivergenceMinimum	0.000	0.000
JointWordsProbabilityMinimum	0.000	*0.000*
Divergence	0.000	*0.000*
Originality	-0.006	0.013
...	...	...
FictionalAdditionsRatio	-0.075	0.028
InterestingLife	-0.116	**0.045**
TotalStoriesGenerated	-0.116	**0.045**
OriginalityAccurate	-0.126	0.024
FictionalRatio	-0.142	**0.039**
RatioCharacters	-0.142	0.039
SettingQuality	*-0.147*	0.024
Names	*-0.147*	0.024
ValenceSum	*-0.174*	0.033

5 Relative Limits of Evaluating Quality

The results previously presented evidence that there is a strong correlation between narrative potential and perceived overall quality of a what-if, which indicates that focusing on narrative plausibility as one of the main factors of quality can lead to better results. Moreover, some of the metrics are weakly correlated to narrative potential. However, these results are still inconclusive, and there is a number of aspects worth mentioning for their influence on the results.

Automatically generating stories and computing useful values for metrics is heavily dependent on the available knowledge. The outcome of the system is constrained by the use of ConceptNet. The amount of relations that can be safely used in ConceptNet is small and the richness and depth of the chains of properties is limited regarding to its use as a source for narrative processing. This makes it necessary to address knowledge management from a different perspective. The WHIM project currently includes a whole module for providing robust knowledge to the rest of the modules, and the impact of the application of this subsystem on the creation and evaluation of what-if ideas will be reported once the results are ready.

The generation process (for the what-ifs, the stories and the metrics) strongly influences the overall outcome. Many design decisions have been taken in order to provide a working, implemented prototype able to generate actual what-ifs, and these decisions set the kind of what-ifs generated, the complexity of the stories and many other aspects. The provided results are then the outcome of a specific implementation which does not claim any generality. However, the approach itself (namely the generation-metric computation-evaluation process) is presented as a generally applicable method for producing novel what-if ideas.

The used metrics for labeling narrative properties do not cover all computable features. There is a large number of aspects that can be extracted from a what-if, and the narrative-based feature extraction module of the What-If Machine does not currently provide coverage for all of them. This is considered to be not strictly relevant with regard to the methodology and scope of the study. To test the second hypothesis (the existence of a correlation between a certain set of metrics and the overall quality and plausibility), the metrics must be improved. For that purpose, the presented study gives valuable insight on which direction to go next.

The weak correlation between our metrics and the quality perceived by humans suggested that considering more sophisticated metrics was necessary. Some of them were considered:

1. **Humanization**: An approximation of how much human-like the main character is, assuming that fictional scenarios use characters that, while behaving like humans, can be non-human.
2. **Empathy**: How much empathy will a reader feel about the characters.
3. **Tragedy**: The amount of tragedy in the story.
4. **Reality**: How real and current the context is. An approximation of fictionally in terms of context.

5. **TimeSpan**: The time span the story covers. It could be minutes, days or years.

Modelling and implementing these metrics proved to be beyond technical capabilities because it required complex, rich knowledge bases (1, 4), reliable text understanding systems (5), sophisticated emotional models (2) or formal versions of narratological models (3). All of these resources are currently not available.

6 Conclusions

The current paper has presented a pilot study trying to gain insight on two hypotheses, namely that (1) human evaluation on overall quality of what-if ideas correlates to the perception of narrative potential and that (2) there is a set of computable metrics that also correlate to this perception. The study has evidenced that there is a strong correlation between quality and narrative potential for humans (1), but failed to prove such a strong correlation between the current metrics and the human ratings. These results have been analysed and discussed in terms of the limited potential of the current implementation of both the fictional ideation procedure and the method employed to evaluate it. Actual implementations lack the required complexity to approximate evaluations with a relatively acceptable level of accuracy, mainly due to the limited technical capabilities of current computational solutions.

References

1. Boden, M.: Computational Models of Creativity. Handbook of Creativity pp. 351–373 (1999)
2. Boden, M.: Creative Mind: Myths and Mechanisms. Routledge, New York, NY, 10001 (2003)
3. Colton, S.: Creativity Versus the Perception of Creativity in Computational Systems. Proceedings of the AAAI Spring Symposium on Creative Systems (Colton 2002), 14–20 (2008)
4. Colton, S.: The painting fool: Stories from building an automated painter. Computers and Creativity 9783642317, 3–38 (2012)
5. Colton, S., Pease, A., Ritchie, G.: The effect of input knowledge on creativity. Technical Reports of the Navy Center for (2001), http://www.inf.ed.ac.uk/publications/online/0055.pdf
6. Colton, S., Wiggins, G.: Computational creativity: The final frontier? ECAI (2012)
7. Gervás, P.: Linguistic creativity at different levels of decision in sentence production. In: Proceedings of the AISB 02 Symposium on AI and Creativity in Arts and Science, 3rd-5th April 2002, Imperial College. pp. 79–88 (2002)
8. Haenen, J., Rauchas, S.: Investigating artificial creativity by generating melodies, using connectionist knowledge representation. In: The Third Joint Workshop on Computational Creativity (2006), http://ccg.doc.gold.ac.uk/events/ecai06/proceedings/Haenen.pdf

9. Jordanous, A.: A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. Cognitive Computation 4(3), 246–279 (2012), http://dblp.uni-trier.de/db/journals/cogcom/cogcom4.html#Jordanous12
10. Kira, K., Rendell, L.: A practical approach to feature selection. In: Proceedings of the ninth international workshop on Machine learning. pp. 249–256 (1992)
11. León, C., Gervás, P.: The Role of Evaluation-Driven rejection in the Successful Exploration of a Conceptual Space of Stories. Minds and Machines 20(4), 615–634 (2010)
12. Llano, M.T., Colton, S., Hepworth, R., Gow, J.: Automated Fictional Ideation via Knowledge Base Manipulation. Cognitive Computation pp. 1–22 (2016)
13. Llano, M.T., Cook, M., Guckelsberger, C.: Towards the automatic generation of fictional ideas for games. Experimental AI in ... (2014)
14. Llano, M.T., Hepworth, R.: Automating fictional ideation using ConceptNet. Proceedings of the ... (2014)
15. Machado, P., Martins, T., Amaro, H., Abreu, P.: Beyond interactive evolution: Expressing intentions through fitness functions. Leonardo (2015)
16. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J.R., Bethard, S., McClosky, D.: The stanford corenlp natural language processing toolkit. In: ACL (System Demonstrations). pp. 55–60 (2014)
17. Nowak, E., Jurie, F., Triggs, B.: Sampling Strategies for Bag-of-Features Image Classification. pp. 490–503. Springer Berlin Heidelberg (2006)
18. Pease, A., Colton, S.: On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. AISB 2011: Computing and Philosophy pp. 15–22 (2011)
19. Pease, A., Winterstein, D., Colton, S.: Evaluating machine creativity. In: Workshop on Creative Systems, 4th (2001)
20. Peinado, F., Gervás, P.: Evaluation of Automatic Generation of Basic Stories. New Generation Computing, Computational Paradigms and Computational Intelligence. Special issue: Computational Creativity 24(3), 289–302 (2006)
21. Pereira, F.C., Hervás, R., Gervás, P., Cardoso, A.: A Multiagent Text Generator with Simple Rhetorical Habilities. In: Proc. of the AAAI-06 Workshop on Computational Aesthetics: AI Approaches to Beauty and Happiness, July 2006. AAAI Press (2006)
22. Pérez, R.y., Ortiz, O., Luna, W., Negrete, S.: A system for evaluating novelty in computer generated narratives. Creativity (2011)
23. Pérez y Pérez, R.: MEXICA: A Computer Model of Creativity in Writing. Ph.D. thesis, The University of Sussex (1999)
24. Ritchie, G.: Assessing creativity. In: Proceedings of the AISB Symposium on AI and Creativity in Arts and Science. pp. 3–11. York, UK
25. Ritchie, G.: Some Empirical Criteria for Attributing Creativity to a Computer Program. Minds & Machines 17, 67–99 (2007)
26. Robnik-Šikonja, M., Kononenko, I.: An adaptation of relief for attribute estimation in regression. In: Machine Learning: Proceedings of the Fourteenth International Conference (ICML 1997). pp. 296–304 (1997)
27. Ware, S.G., Young, R.M.: Validating a Plan-Based Model of Narrative Conflict. In: Proceedings of the International Conference on the Foundations of Digital Games. pp. 220–227. ACM Press, New York, New York, USA (2012)
28. Ware, S.G., Young, R.M., Harrison, B., Roberts, D.L.: Four Quantitative Metrics Describing Narrative Conflict.pdf. pp. 18–29. Springer Berlin Heidelberg (2012)

29. Wiggins, G.: A preliminary framework for description, analysis and comparison of creative systems. Knowledge-Based Systems 19(7) (2006)
30. Wiggins, G.: Searching for Computational Creativity. New Generation Computing, Computational Paradigms and Computational Intelligence. Special Issue: Computational Creativity 24(3), 209–222 (2006)

TypeAdviser: a type design aiding-tool

João M. Cunha, Tiago Martins, Pedro Martins, João Bicker, Penousal Machado

CISUC, Department of Informatics Engineering, University of Coimbra
{jmacunha,tiagofm,pjmm,bicker,machado}@dei.uc.pt

Abstract. The number of people who design typefaces has drastically increased in the last twenty years. However, not all typefaces work as they should, i.e., as a group of characters with shared attributes. We present a tool for helping type designers in their creative process, which explores the anatomic relations among characters of a typeface. Having computer-aided design as a goal, our tool helps in the early stages of designing a typeface by using semi-automatic letter-part sharing and allowing the users to compare their design with existing typefaces.

Keywords: Computational Design, Computer-Aided Design, Type Design

1 Introduction

In the last century there was a massive proliferation of typefaces, being difficult to know how many there are today [3]. The accessible type design tools and the ease of designing and releasing a typeface led to a huge number of typefaces, whose quality is quite diverse.

A typeface has a key role in terms of communication as can even change the intent of its message. However, its importance is often overlooked and the impact of the communication is negatively affected by choosing an unsuitable or even poorly designed typeface. As changing users' habits is something not feasible, we believe that we should instead focus on trying to improve the way users design typefaces. By doing this, we aim at increasing the overall likelihood of the production of better quality designs by both experienced type designers and users without a design background. One of the criteria to evaluate a typeface's quality is its *consistency* – the way typeforms match each other, i.e. how the characters are designed into a set of diversified and yet unified forms [2].

Having computer-aided design as a goal, we focus on helping the designer in the early stages of the design process by combining type design principles, derived during the research stage, with an exploration of the Creativity-Support Tools domain, using mixed-initiative approaches [15].

We present a prototype of a type design aiding-tool which is currently a prototype and is composed of two different components: *the Part-sharing* – based on the consistency criterion – and *the Adviser* – which uses Self-Organizing Maps to suggest similar designs to the user.

2 Related work

The development of type related digital applications is nothing new and the current approaches can be divided into different categories according to their authors/users or their end purposes. There are three main types of authors: designers, engineers and artists. This difference on background knowledge has a huge effect on the type of application: some are purely artistic; others are technical and often dependent on too rigid rules or structure, thus resulting sometimes in visually non-appealing outputs; and others are within the scope of software specifically developed for type design – none of which explores the domain of computational creativity to its advantage in a significant way. Being more interested in type design, we will avoid addressing applications which have a visual artifact as output and focus on those that have a font as final product.

CarveLCD [1] generates letters using a grid which defines the location of their outline-points. This grid is controlled by the user and its modification is applied to every character, maintaining some coherence among them. It often results in abstract shapes which are not easily read.

Other applications allow the user to change the value of pre-defined parameters which control a given characteristic in order to generate fonts (e.g. *Metaflop* [5], *Modular typographic generator* [8], *Typeconstructor* [10] and *Prototypo* [13]).

Schmitz [14] developed a tool which uses Genetic Recombination as a metaphor and allows the user to recombine a limited number of fonts generating hybrids. This generation is done by using their genes which store information about three features: font skeleton, line width and serif shape.

There are also some systems that use Evolutionary Computation for designing type. Some require user interaction (e.g. [11]) and others focus on being autonomous (e.g. *Evotype* [12] which generates glyph designs from scratch).

In spite of the existence of such applications and systems, not only none of them addresses all the issues we have previously identified but also their output is considered the final product – we act on the initial stage of the design process. Concerning the followed approach, most of the aforementioned applications rely on predefined typefaces or skeletons, whereas we aim to allow the user to input its own designs. Additionally, the majority of them presents as final output something that – as a text font – has low quality both in terms of legibility and coherence among characters.

3 Aiding type design: *Part-sharing*

Our tool aims at helping type designers by combining co-creativity with type design principles. The developed prototype is currently composed of two different components: the Part-sharing and the Adviser. The Part-sharing component is strongly based on type design principles and uses them to both help designer reduce time-consuming tasks and stimulate creativity.

1. Looking for consistency – By focusing on the principle of consistency, the first stage of the development of the Part-sharing component has three main

goals: identify the characteristics that make a typeface work as a whole; understand the reason for the existence of relationships among letters; and identify the possible patterns in their construction.

Our research approach consisted not only in bibliographic research but also in the conduction of interviews with type designers. These interviews allowed us to understand their way of working and the details they pay attention to when drawing a typeface. In addition, we also analyzed the different letters, in order to identify patterns in their construction and infer possible rules followed in the design of a text typeface. The collected material and reached conclusions were then used as a support and applied in the development of the application.

As observed in some of the projects presented in Section 2, it is possible to make a metaphor between typefaces and biology: different typefaces can be seen as different species as they have a different structure. On the other hand, as living creatures within the same species share a given karyotype, within a typeface the different letters also share a set of characteristics.

According to Meseguer [7], there are three distinct ways a designer can use to create a set of characters: (1) *step by step* – in which each character is divided in parts according to the calligraphic drawing and stays that way until the last stage of the drawing process; (2) *modular* – which is based on the repetition of shapes throughout the set of the characters (e.g. the stem of *i* is repeated on the *n* or *m*); (3) *shape derivation* – characters are drawn in a sequence and shapes are derived from others.

Both the repetition of modules and shape derivation have key roles in terms of giving coherence and maintaining the harmony among characters.

By drawing the first letters, it is already possible to foresee how the others will be as making a decision for one letter often affects the rest of the letters. The designer is then faced with the question of which elements of the drawn letters can be used in the new ones and which ones must be adjusted or modified.

In addition, it is possible to separate the characters into several groups based on their morphological similarity (e.g. in upper case the round shapes group is *O, Q C, G* and *S*; in lower case is *o, c* and *e*). This categorization is of great importance since similar structures can and should be designed as related forms [3]. Moreover, by firstly drawing one character of each group, it is possible to design the remaining characters with less effort and sparing time.

The process of designing in sequence was mentioned by all the interviewed designers when describing their way of working. They not only share the preference of drawing the lowercase first – upper case is much more limited in terms of creativity and differentiation – but they also have a set of favorite first letters.

Most designers start with the key letters which define the proportions and personality of the type, good starting points are the lowercase letters *a,e,g,n* and *o*. The letter *n* is normally one of the first, due to the fact that it is easy to obtain parts of the letters *h, m, u* and *r* from it. And with it, it is possible to define a lot of the rhythm and proportion of the typeface.

2. Our approach – Our initial ideia was to establish rules that would, in some way, allow to generate all the characters of a typeface based only on

some given by the user. The user would input some previously drawn vectorial glyphs and then identify their parts. After identifying the letter-parts, it would be possible to generate the rest of the letters. This generation would be based on the predefined rules and would use the details about the introduced glyph, given by the user. After the generation, the user would be able to change the generated shapes, in order to correct any imperfection, and afterwards export the results. The goal of this concept was to generate a font nearly ready to be used. The end version of the application works similar to what has already been described but its main goal is completely different.

As explained in Section 3 (*Looking for consistency*), a system with too rigid rules, would lead to something greatly repetitive and would probably result in something with a modular appearance. Such a product would go against one of our main goals: to have something serious in terms of typographic quality and useful as a text typeface. According to Goudy [6], the characters should be coherent among each other but at the same time each of them should have a quality of completeness and not seem to be made of pieces put together. Obviously, a modular-looking result would fall short of this idea. In addition, there is often the need of doing optic compensations before considering a typeface as final. These were not considered in the initial concept.

Another important issue is related to the introduced characters. As mentioned in Section 3 (*Looking for consistency*), there are some letters that allow to foresee how the rest of the font might look like (e.g. the *n* or *o*) – by partially defining the proportions, contrast, etc.. However, there are others that do not and are much more difficult to extract useful information from (e.g. *k*, *x* or *z*), consequently increasing the difficulty of generating the rest of the characters. Moreover, just a few letters are not enough to generate the all of the remaining characters, which denotes a clear problem in our initial concept.

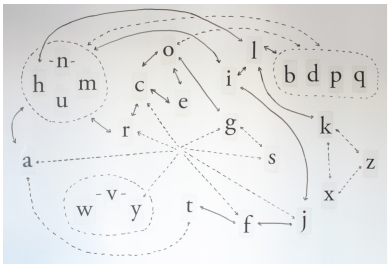


Fig. 1. Left side: Relationships among lowercase letters. The letters inside round shapes are greatly related; the lines symbolize a relationship between two letters; the dashed lines are less relevant relationships. Right side: Some of the skeleton characters.

For already mentioned reasons, the goal of the application was changed: instead of being considered as an application able to generate a complete and ready to use font, it should be seen as an aid-tool to the type designer, giving

support to the design task and stimulating creativity. It helps to reduce the time spent by the designers in the initial stage of the design process.

The way of working is highly similar to what has already been described. The user is able, at any time, to add new previously drawn glyphs and identify their parts. The shape-repetition is done automatically by the application. When the user is satisfied with the characters drawn using the tool, he can continue the design process with another software specially developed for type design, using the output of our application as a draft and an initial version of the typeface.

The current version is a prototype, not having all its functionalities fully implemented. In the part-sharing component, we have chosen to begin with the lowercase because it has a greater importance and usage – as already mentioned. Not only that but also all the interviewed type designers start the majority of their typefaces with the lowercase letters. The implemented rule system was based on the identified shape-sharing relationships among letters (see Fig. 1).

3. User input – The current version of the application was implemented using *Processing* and the library *Geomerative*. This library was used to read the introduced *svg* files and access the points and Bézier curves of the vectorial shapes. By introducing a vectorial file, the user is first asked to identify the letter it corresponds to. The user is then able to identify the several parts of the introduced glyphs. These parts are predefined in the application, i.e. the user has a list of possible parts to identify (based on the relationships, see left side of Fig. 1). This identification is done through point selection of all the points belonging to the part being identified. For example: having introduced a serifed glyph, the user can select all the outline points which are part of the serif and choose "serif" to save it as such. After identifying a letter-part, the application automatically checks which letters might use it in their construction and assigns it to them (see Fig. 2). The input is not limited: the user is able to input vectorial files for as many letters as he/she wishes and even replace a previously given one. This allows the user to work step-by-step in an incremental way.

4. Letter-part attribution and positioning – The attribution of the letter parts is done according to two criteria: *need* – does the letter need that part or does it already have one specially drawn for it? – and *suitability degree* – is there any other already defined letter-part which is more suitable or should it be replaced? For example, the top part of the l stem can be used to make an i stem but the top part of the j might be more adequate.

One of the issues that had to be solved was related to the positioning of letter-parts – i.e. how should a copied shape be positioned? The solution was to use a letter skeleton with the several letter-parts identified (see right side of Fig. 1). Contrary to some of the applications mentioned in Section 2, the skeleton is only used for letter-part positioning purposes. This is not an optimal solution as it does not account for every style/proportion – in fact it only works without error if they are the same as the skeleton. Despite this drawback, we do not consider it a problem, as the user is able to move the letter-parts.

The positioning according to the skeleton is also not trivial, as the positioning method is not the same in all letter components. An example of this can be seen

in the letters n and l. The left stem of the letter n can be positioned by aligning it to the left with the corresponding skeleton part. Differently, the stem of the l is centered in relation to its skeleton. This makes it necessary to previously define the positioning method for each letter-part.

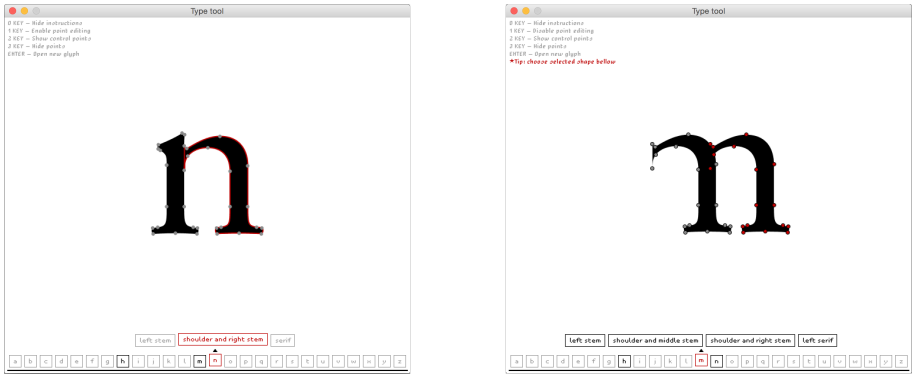


Fig. 2. Left side: Selecting letter-part. Right side: Letter m built using part-sharing

5. Editing shapes – In spite of not having a final typeface as a goal output nor aiming at developing an application to compete with software specialized in type design, we decided that the user should be able to edit the letter shapes. The importance of this functionality is even more evident when considering the positioning issue. Therefore, the user is able to both move shapes, by selecting all their points, move single points or even their respective control points. This makes it possible to change the generated shapes in order to correct any imperfection and make slight adjustments (e.g. optical compensations).

4 Aiding type design: *Adviser*

The second component is used as a guide and its purpose is to present the user with possible letter designs, based on a user given glyph. The suggested designs are from already released typefaces that are similar in style to the one introduced by the user. This functionality is particularly helpful for non-experienced users and has two main possible uses: (i) it allows the user to see how the characters of similar-looking typefaces were designed, thus being useful to guide the user when designing a typeface of a specific style; on the other hand, (ii) it helps the user in distancing the typeface being designed from others, thus avoiding an excess of similarity of the output.

To implement the Adviser, we use a Self-Organizing Map (SOM) [9], which is a technique normally employed in data visualization, as it reveals relationships between vast amounts of data. It consists of self-organizing neural networks which require no external supervision and are able to represent multidimensional data in a space of a lower number of dimensions. The output is a map which groups similar data items together, displaying similarities.

A SOM is used to produce a map of different character styles for each letter, organized in terms of similarity. This makes it possible to group similar typefaces and find the ones visually near to a given one. Moreover, it also allows a gradual distancing from a particular style – without getting too far from what is established by tradition for that specific style of typeface. It is important to bear in mind that breaking with tradition when designing a typeface is often not a good choice as it might have a negative effect on legibility and easiness of reading [4]. By using SOMs, a user can start by drawing one letter and then see different possibilities for other characters, ordered according to similarity degree.

1. Producing SOMs – Differently from the component described in Section 3, in the current version of the Adviser component we only used uppercase letters. This was mainly due to fact that, the upper case is much more limited in terms of differentiation and creative freedom when compared to the lowercase [7]. Given that the Adviser deals with variation of style, we considered that the uppercase was a better choice for a first assessment of the validity of using SOMs, avoiding the greater amount of character style variation of the lowercase.

We produced a SOM for each letter with a dataset of 4034 typefaces. The currently used SOMs followed the following experimental parameters: *number of iterations*: 800; *lattice size*: 25×25; *samples per character*: 250.

Following the production of the SOMs, we calculated the correspondence between the dataset typefaces and the best match SOMs node for each character, i.e. each SOM (e.g. SOM of the character *A*, see left side of Fig. 3) has a list of best match node for each typeface (e.g. the node that is most similar-looking to the glyph *A* of a given typeface). This establishes proximity relationships among the typefaces from the dataset.

2. Finding look-alikes – The Adviser component uses the produced SOMs to suggest the most similar-looking typefaces, based on the glyph given by the user. This process consists in finding the node of the character's SOM that has the highest degree of similarity and then provide the user with a list of typefaces. The similarity comparison is done using the Euclidean distance – also used in the SOMs production – which compares the input glyph with each node. The list of best matched typefaces is then produced by gathering the typefaces which were attributed to the best matched node and the nodes that are close to it. This results in a list ordered by similarity degree.

3. Guiding the user – The interface is the same as the one developed for the first component, as they are part of the same tool. When using the functionalities of the *Adviser component*, the user is able to click on every letter button, even though some of the letters do not have a defined glyph yet. This is due to the components way of working: it does not need a glyph to show a suggestion of a possible design.

The suggestion is shown with a shadow-like effect, being only a visual clue to the user (see right side of Fig. 3). There are two different modes of suggestion: single suggestion – only a glyph typeface is shown – and multiple suggestion – several are shown by overlaying them. Despite being totally functional, the

interface still needs some improvements in order to make both components fit well together in terms of usability and way of working.

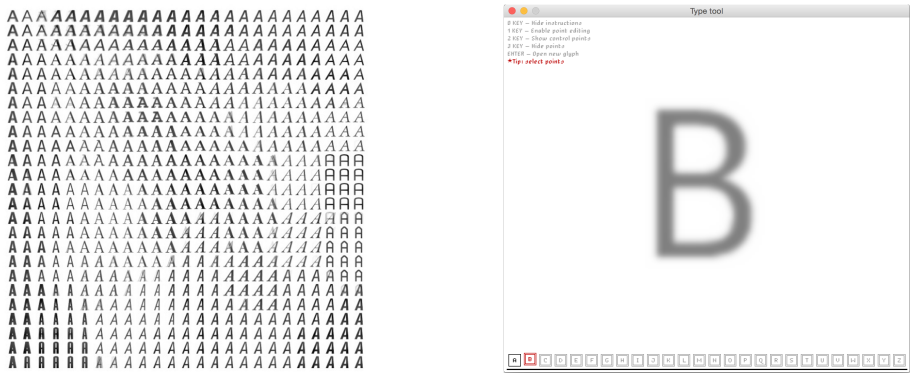


Fig. 3. Left side: SOM of the character *A*. Right side: Suggestion of *B* based on glyph *A* introduced by the user.

5 Conclusion & future work

We have presented and described the functionalities of a type design aiding-tool which has as main purpose to make the design process easier by reducing time-consuming tasks and stimulating creativity. It follows a mixed-initiative approach and is currently composed of two different components: *the Part-sharing* – strongly based on type design principles – and *the Adviser* – which uses Self-Organizing Maps to suggest similar designs to the user. The main goal is not to create completely finished and ready to use typeface but instead to allow the user to design a first draft by using the tool as a support. It is to be used in the early stage of the design process as the user will be able to export the typeface draft and continue the process in an application specially developed for type design, thus allowing the user to afterwards improve details and correct drawing problems such as those related to optical compensations.

Concerning the *Advisor component*, there are also issues to be solved. We consider that the use of SOMs fullfils its purpose as it allows similar typefaces to be found. Notwithstanding, the used dataset has some clear problems which need to be corrected: (1) Some of the typefaces used are too similar; (2) It contains different weights – this is pointless, as we are only designing the regular weight.

In addition, as every SOM is different and finds different similarities among the data, more attention should be given to both SOM production and SOM quality assessment as well as to the used experimental setup.

Regarding the process of finding similar typefaces using SOMs, we aim to further develop its way of working by focusing on multiple matching. The current version only finds similar typefaces based on a single glyph; with multiple matching it would be based on several glyphs, consequently aiming to find typefaces that match the maximum number of glyphs.

References

[1] Balzien, A.: CarveLCD. In: Pape, P., Jenett, F. (eds.) Gestalten mit Code, FH Mainz. http://generative-typografie.de (last accessed in June 2016)
[2] Celso, A.L.: Rhythm in type design (2005)
[3] Cheng, K.: Designing type. Yale University Press (2005)
[4] Cunha, J.M.: Dissertation on anatomical relationships among characters of a typeface (Dissertação sobre relações anatómicas entre caracteres de um tipo de letra). Master's thesis, University of Coimbra (2013)
[5] Eglie, S., M.M., Reigel, A.: Metaflop. http://www.metaflop.com/ (last accessed in June 2016)
[6] Goudy, F.W.: Typologia: studies in type design & type making, with comments on the invention of typography, the first types, legibility, and fine printing. Univ of California Press (1977)
[7] Henestrosa, C., Meseguer, L., Scaglione, J.: Cómo crear tipografías: del boceto a la pantalla. Tipo E (2012)
[8] Kaniowski, A.: Generative stuff. http://generativestuff.com (last accessed in January 2013)
[9] Kohonen, T., Schroeder, M.R., Huang, T.S. (eds.): Self-Organizing Maps. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 3rd edn. (2001)
[10] Koomen, J.: Haagse letters. http://haagseletters.nl (last accessed in June 2016)
[11] Lund, A.: Evolving the shape of things to come: A comparison of direct manipulation and interactive evolutionary design. Proceedings of Generative Art 2000 (2000)
[12] Martins, T., Correia, J., Costa, E., Machado, P.: Evotype: Evolutionary type design. In: Evolutionary and Biologically Inspired Music, Sound, Art and Design, pp. 136–147. Springer (2015)
[13] Mathey, Y., Babé, L.R., et al.: Prototypo. https://www.prototypo.io (last accessed in June 2016)
[14] Schmitz, M.: genotyp, an experiment about genetic typography. Proceedings of Generative Art 2004 (2004)
[15] Yannakakis, G.N., Liapis, A., Alexopoulos, C.: Mixed-initiative co-creativity. In: Proceedings of the 9th Conference on the Foundations of Digital Games (2014)

Associating Colors to Emotional Concepts Extracted from Unstructured Texts

Alberto Fernández-Isabel, Antonio F. G. Sevilla, and Alberto Díaz

NIL**, Facultad de Informática, Universidad Complutense de Madrid
c/ Profesor José García Santesmases 9, 28040, Madrid, Spain
afernandezisabel@ucm.es, afgs@ucm.es, albertodiaz@fdi.ucm.es

Abstract. Software creativity is a pervasive branch inside Artificial Intelligence where multiple approaches have place. In the case of developing a software artist the main difficulty resides in retrieving and expressing the feelings of the context properly. This paper addresses that issue introducing a framework able to produce color palettes and abstract paintings based on processing unstructured text. To achieve it, the tool extracts the concepts related to human mood from the text once it is analyzed. Then, they are checked to identify the color that represents their psychological meaning using related literature. The resulting picture is built according to the dimension of the canvas and the amount of colors from the palette obtained. A case study illustrates the applicability of the proposal using two texts selected from Wikipedia. They describe disparate concepts as love or death. A related work is considered to situate the approach in the context and establish comparisons.

Keywords: Natural language processing, concept extraction, abtract painting, painter software, semantic expression, information retrieval

1 Introduction

Abstract painting presents difficulties to produce objective evaluations. These are related to its foundations as subjective field. This subjectivity resides in the feelings, emotions and mood an individual (i.e., the artist) wants to manifest or illustrate, and the evaluation made by others with different backgrounds or sensibilities [10]. In order to alleviate these problems there are multiple approaches that evaluate the implicit psychology of colors and their possible meanings [5, 12]. These associations (though they are culture-specific [1]) allow establishing certain connections and relationships among the abstract art and the expressed feelings and their representations.

This literature can be adopted by intelligent software tools in order to reproduce the inherent creativity [21]. This fact might lead these tools to generate abstract paintings based on forms or colors. These can be organized into the picture for simulating some specific techniques used by important human artists

** Group of Natural Interaction based on Language

[14] or distributed over the canvas randomly focusing on more generic abstract painting processes [20].

Our proposal introduces the framework called CALyPSe (Conceptual Abstract and Lyric Painting System) that is able to produce color palettes and abstract paintings. The resulting pictures are generated from concepts related to human feelings and emotions extracted from an input text. These are checked to predefined notions which are associated to a specific color following the relationships introduced in [12]. The final painting is built according to the dimensions of the canvas and the color palette. This latter is generated considering the amount of concepts identified and harmony color techniques [4]. The organization of pixels which compose the complete image follows a randomly distribution.

The framework is composed by three different modules: *concept extraction*, *concept storage* and *painting generation*. They count with their respective manager which is responsible for connecting their internal tools and make them work together. The first module takes as input an unstructured text and extract the concepts from it through Freeling [18] and a *semantic analyzer*. In the second one its manager stores these concepts that can be related to a specific color in the *knowledge database*. These relationships might be produced directly or through synonyms. These latter are obtained using WordNet [15]. Optionally, the concepts gathered can be searched on Wikipedia, processing the first paragraph of their definition if the web page exists. The third module is oriented to develop the final painting according to the information collected.

A case study illustrates an experiment where abstract paintings based on obtaining concepts from texts are generated. Two different texts gathered from Wikipedia related to human emotions and feelings are selected. The first one describes *love* while the second talks about *death*. The tool generates two abstract paintings according to them. These are analyzed both qualitative and quantitative, focusing on two specific points: the colors used to produce the pictures with each one of the selected texts, and the amount of different concepts identified and their proportion.

The rest of the paper is organized as follows. Section 2 compares this proposal with related work. The CALyPSe framework is introduced in Section 3 delving into their modules and managers that are responsible for them. The case study in Section 4 shows the application of the approach. Finally, Section 5 discusses some conclusions and future work which concern to the issue.

2 Related work

The framework presented in this approach links natural language processing and information retrieval with the expression of emotions through abstract painting. This section discusses the existing tools and its foundations comparing them with their alternatives.

The framework accomplishes the extraction of concepts from an input text according to its dependency analysis through Freeling [18]. This tool automatizes the required steps using different layers of linguistic analysis. A similar alterna-

tive consists of the Stanford parser [11] which is also able to generate their own type of specific relations and dependencies applying stochastic analysis. Both tools are designed to ease their integration into a more complex pipeline.

The semantic information retrieval and lexical support for obtaining synonyms is provided by WordNet [15]. It is the standard lexical knowledge base where certain related items can be collected (e.g., verbs, nouns or adjectives).

Regarding the painting artist frameworks, there are multiple approaches but three main perspectives are related to our proposal: frameworks that mimic existing human artists styles, creative painting based on feelings and emotions, and collage generation.

Mimic human styles is a complex issue due to each painter develops a personal style. Nevertheless there are proposals that try to evoke some notions or standards related to a specific painter [14] or painting techniques [3]. Others more generic are focused on generating images simulating their own style as painters [6] or rendering images to simulate certain strokes techniques in paintings [7].

Creative painting is oriented to simulate human emotions through an intelligent software that follows some rules or background [9]. One of the famous frameworks in this field is The Painting Fool [8]. It is based on creativity associated to the elaboration of pictures through non-photorealistic rendering.

Collage generation consists of producing a picture integrating different images that could have sense together. The information to represent can be obtained from different sources. One of the most common is the processing of unstructured texts [13]. The different images which are added to the painting are usually extracted from the web.

There also are frameworks similar to CALyPSe. For instance, [17] is able to produce images according to a set of adjectives that describe the input.

CALyPSe framework generates its paintings taking as a reference the abstract art, while it identifies the concepts related to thoughts and emotions from an input text associating them to colors. The conception process of these paintings uses a random algorithm (instead of positioning or organizing the pixels) in order to simulate modern art guidelines [20].

3 Associating colors to emotional concepts

This approach presents CALyPSe which is a framework oriented to produce color palettes and abstract pictures that express the emotions and feelings included in an unstructured text. To accomplish it, the tool has a set of modules in order to classify the different tasks to perform (e.g., concept extraction or color comparisons) until the picture is generated.

It has an structure based on three main modules (see Fig. 1): *concept extraction*, *concept storage* and *painting generation*. The first processes the input texts and organizes the concepts captured from them by sentences. The second has the storage structures and implements the connection to WordNet database[15]. The third is in charge of collecting the knowledge stored and producing the resulting abstract picture.

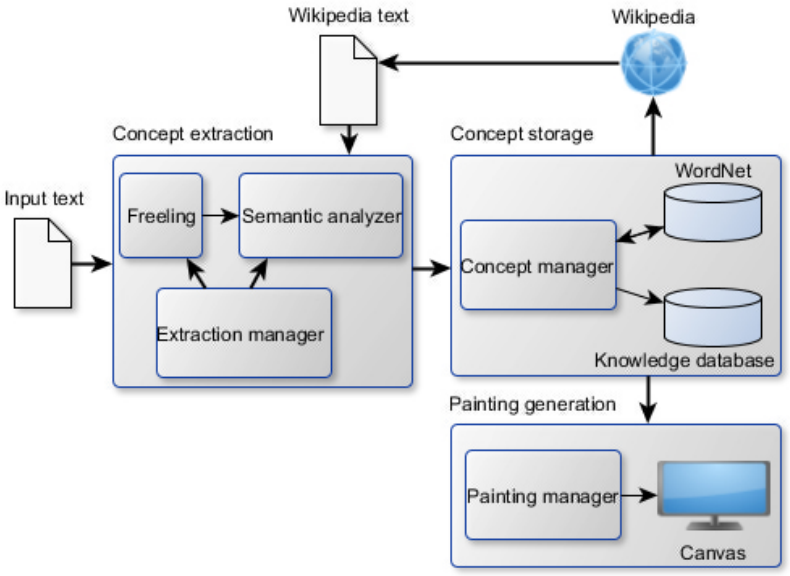


Fig. 1. Excerpt of the framework structure.

Section 3.1 introduces the *concept extraction* module, while the *concept storage* is described in Section 3.2. The issues related to the *painting generation* module are addressed in Section 3.3.

3.1 Concept extraction

This module is in charge of processing the current input text identifying its concepts. To achieve this operation it uses two items: Freeling [18] and a *semantic analyzer*. The *extraction manager* synchronizes both in order to generate the appropriate result (see Fig. 1).

Freeling [18] evaluates the input text achieving a dependency analysis. This allows producing its syntax structure and the identification of the different types of words (e.g., nouns, verbs or adjectives). This structure is captured by the *extraction manager*.

The *semantic analyzer* receives as input the syntax structure stored by the *extraction manager*. It examines the different elements and extracts the lemmas of the concepts discarding the pronouns and the articles (they are not appropriated to be compared to the concepts related to colors). The manager is also in charge of producing the final result of the module, organizing these lemmas by sentence following the order of precedence provided by the text.

Note that the module obtains the input texts through three different ways according to the features provided by the framework. It is able to analyze unstructured text, read files and collect their texts, or process a web page selecting

Key	Value (RGB)
Innocence	(253,233,242)
Love	(250,63,66)
...	...
Purity	(237,237,245)

Table 1. Excerpt of the *color structure* in the *knowledge database*.

Sympathy	Harmony	...	Love
3	2		3
Sympathy	Harmony		Love
Dog	Landscape		Girl
Friend			Dog

Table 2. Excerpt of the *concept structure* in the *knowledge database* associated to the text: *"The girl feels love and sympathy for her dog." Her friend also has sympathy for the dog. The landscape generates much harmony in them."*

only its paragraphs. This latter uses techniques related to web scraping [16] that allow extracting only the raw text.

3.2 Concept storage

It is the module where the information collected from the input texts is processed and stored. It is composed by a *knowledge database*, a WordNet [15] database connection and the *concept manager* that control the interactions among these elements (see Fig. 1).

The *knowledge database* presents two internal structures: *color structure* and *concept structure*. The first one stores 179 relations among concepts (i.e., keys) and colors collected from the literature that concerns to psychology of colors [12] (see Table 1). The second contains a set of head concepts which are the keys with an associated color in the first structure. Each one of the positions of the *concept structure* stores how many times the head element has been related to the concepts from the input text and a list of items. This list might contain the same concept of the head element or associated concepts. The list supports the simulation of associative learning based on words in sentences [19]. In order to achieve it, when a notion does not match with any head element or item of lists but others of the same sentence does, the unmatched concept is stored in the list of these latter (see Table 2). This establishes a semantic relationship among the concept and the rest of notions that could be identified in its same sentence.

WordNet [15] is used by the *concept manager* to find synonyms. It allows associating more notions from the text with the head elements of the *concept structure*.

The module presents an optional set of items related to Wikipedia in order to increment the color enrichment of the final painting. It is able to extend each

one of the concepts extracted from the text in order to find a description of it in its related Wikipedia web page. In the case the page exists, it is scraped and the first paragraph is gathered. This text is sent to the *concept extraction module* where its concepts are obtained. These are stored in the *concept structure* if they produce matches with its head elements. In order to avoid infinite cycles the concepts selected from this part has an special identification.

3.3 Painting generation

This module produces the color palette and the resulting picture. It is accomplished using both structures contained in the *knowledge database* (i.e., *concept structure* and *color structure*).

The *painting manager* is responsible for achieving the painting generation. It applies a set of rules related to color harmony [4] in order to generate the color palette. These rules are implemented through a basic filter. It allows discarding mixtures among combinations of similar colors (where the most important prevail) that do not produce an appropriate contrast to human eye (e.g., red color does not match to pink color).

Then, the *painting manager* produces the final composition obtaining the number of elements of the positions of the *concept structure* that are related to the filtered colors. This is obtained comparing the head elements to the keys of the *color structure*. Thus, the painting is built randomly according to the proportion between the number of elements and the dimension of the canvas.

4 Case Study

The case study illustrates how the tool produces two different palettes and pictures based on them from unstructured texts selected from Wikipedia. These texts introduce information about two specific human feelings or thoughts: *love* and *death*. The images obtained from them are compared in a visual way and the amount of concepts detected during the process are analyzed. The Wikipedia optional feature provided by the framework (see Section 3.2) is not considered in this case.

The framework starts reseting the *concept structure* of the *knowledge database*. Then the *extraction manager* scraps the corresponding web pages obtaining the input texts. These texts are provided to Freeling [18] in order to generate their dependency analysis. This output is processed by the *semantic analyzer* which is in charge of obtaining the lemmas of the identified concepts.

In the next step the concepts are analyzed in the *second module*. In it, the framework tries to link them to the head elements or to the items of their lists provided by the *concept structure*. In order to increment the matches, the tool uses WordNet [15]. It obtains synonyms that might be linked to the head elements. If the current concept is directly related or conversely one of its synonyms, the first one is stored in the list of items of the specific head element.

Input text	Head elements	Items in lists	Synonyms	% Matches
Love	8	1433	5	11%
Death	2	48	0	5%

Table 3. Summarization of matches in texts related to *love* and *death*.

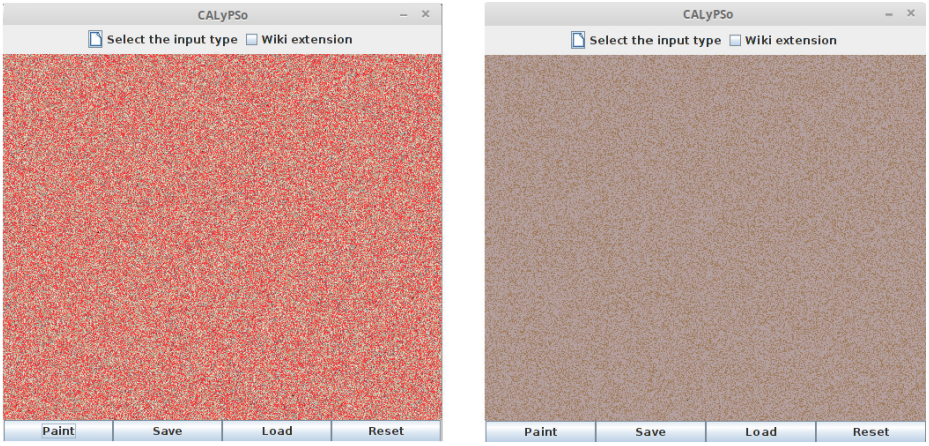


Fig. 2. Pictures generated by the framework related to *love* and *death* texts.

In the case of *love* text, some of its concepts can be checked easily (e.g., *affection* is associated to a head element that contains the *love* concept). The text that concerns to *death* has similar situations with other concepts (e.g., *solitude*) but no with the *death* concept. It happens because of the *death* concept is not considered in the *color structure* as a key (i.e., it is not evaluated in the literature [12]). The associative learning [19] solves this situation inserting the *death* notion into the list of items of the concepts related to head elements of its same sentence. Thus, when *death* appears again it will be matched.

Once the concepts are stored in the *concept structure*, the palette is produced. In both cases there are not similar colors to filter so that every head element is selected in order to generate the paintings. *Love* painting is composed by eight different colors (i.e., head elements) while the *death* presents only two colors (see Table 3). Nevertheless, the amount of pixels in both pictures suggests a high number of matching with the same head elements and lists of items.

Regarding the visual effect, the picture that depicts the *love* text is noticeably red with several spots of different colors (see Fig. 2). In the second picture concerned to *death* text, there are multiple spots with shades of gray. This makes the painting sad and dull. Therefore, it could be said that visually both pictures seem to illustrate some of the meanings of the concepts introduced in the texts.

Finally, delving into a quantitative analysis it can be found that in the text concerned to *love* there are more matches related to the head elements of the *concept structure* (i.e., more colors are used to draw the painting) than in the

death text. The same occurs in the case of the concepts of the lists of items (see Table 3). The amount of related synonyms is low in both cases or even irrelevant (*death* text does not provide concepts where their synonyms match to head elements). An enriched *color structure* in the *knowledge database* with a wider list of concepts associated to colors could enhance the problem in future experiments.

5 Conclusions

This paper has introduced the CALyPSe framework to generate color palettes and abstract pictures. These are produced from unstructured texts which are processed to gather their emotional concepts and associate them to colors using related literature [12].

The tool consists of three modules (*concept extraction*, *concept storage* and *painting generation*) and their respective managers. The first one is in charge of processing unstructured text using Freeling [18] and a *semantic analyzer*. The second stores the concepts related to colors (i.e., *color structure*) and the matchings among these and the notions coming from the input text (i.e., *concept structure*) in its *knowledge database*. It presents an optional feature that eases the acquisition of extra information. It is based on searching each concept on Wikipedia. The third is focusing on producing the color palette through color harmony techniques [4] and drawing the painting.

The case study shows the viability of the proposal using two different texts that have been selected from Wikipedia. They describe concepts as *love* and *death*. Their visual representations are compared observing the differences among them. A quantitative analysis is achieved focusing on the amount of concepts extracted from the texts that match to the head elements provided by the *concept structure*, or which ones are stored in the lists of items. Synonyms associated to the concepts are also considered showing a low hit rate.

More experiments support the proposal but it is still ongoing work with open issues. New concepts from literature related to the psychology of colors have to be inserted in the *color structure* of the *knowledge database*. This will allow obtaining higher percentages in the matches among concepts from texts and head elements. Another point to consider consists of adopting some painting techniques and image rendering [2] in order to produce more realistic pictures.

Acknowledgements

This work is funded by ConCreTe. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

This research is funded by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund (TIN2015-66655-R (MINECO/FEDER)).

References

1. Aslam, M.M.: Are you selling the right colour? a cross-cultural review of colour as a marketing cue. Journal of marketing communications **12**(1), 15–30 (2006)
2. Baxter, B., Scheib, V., Lin, M.C., Manocha, D.: Dab: interactive haptic painting with 3d virtual brushes. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, pp. 461–468. ACM (2001)
3. Baxter, W., Wendt, J., Lin, M.C.: Impasto: a realistic, interactive model for paint. In: Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering, pp. 45–148. ACM (2004)
4. Burchett, K.E.: Color harmony. Color Research & Application **27**(1), 28–31 (2002)
5. Cherry, K.: Color psychology: How colors impact moods, feelings, and behaviors. About Education (2015)
6. Cohen, H.: The further exploits of aaron, painter. Stanford Humanities Review **4**(2), 141–158 (1995)
7. Colton, S.: Stroke matching for paint dances. In: Proceedings of the Sixth international conference on Computational Aesthetics in Graphics, Visualization and Imaging, pp. 67–74. Eurographics Association (2010)
8. Colton, S.: The painting fool: Stories from building an automated painter. In: Computers and creativity, pp. 3–38. Springer (2012)
9. Colton, S., Pease, A., Charnley, J.: Computational creativity theory: The face and idea descriptive models. In: Proceedings of the Second International Conference on Computational Creativity, pp. 90–95 (2011)
10. Csikszentmihalyi, M., Robinson, R.E.: The art of seeing: An interpretation of the aesthetic encounter. Getty Publications (1990)
11. De Marneffe, M.C., MacCartney, B., Manning, C.D., et al.: Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC, vol. 6, pp. 449–454 (2006)
12. Heller, E.: Psychology of colour. Yedam (2005)
13. Krzeczkowska, A., El-Hage, J., Colton, S., Clark, S.: Automated collage generation-with intent. In: Proceedings of the 1st international conference on computational creativity, p. 20 (2010)
14. Lee, S., Olsen, S., Gooch, B.: Simulating and analysing jackson pollock's paintings. Journal of Mathematics and the Arts **1**(2), 73–83 (2007)
15. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
16. Munzert, S., Rubba, C., Meißner, P., Nyhuis, D.: Automated data collection with R: A practical guide to web scraping and text mining. John Wiley & Sons (2014)
17. Norton, D., Heath, D., Ventura, D.: Establishing appreciation in a creative system. In: Proceedings of the international conference on computational creativity, vol. 301, pp. 26–35. Citeseer (2010)
18. Padró, L., Stanilovsky, E.: Freeling 3.0: Towards wider multilinguality. In: LREC2012 (2012)
19. Prior, A., Bentin, S.: Word associations are formed incidentally during sentential semantic integration. Acta psychologica **127**(1), 57–71 (2008)
20. Schönlieb, C.B., Schubert, F.: Random simulations for generative art construction–some examples. Journal of Mathematics and the Arts **7**(1), 29–39 (2013)
21. Wiggins, G.A.: Searching for computational creativity. New Generation Computing **24**(3), 209–222 (2006)