Blend City, BlendVille

João Gonçalves, Pedro Martins, Amílcar Cardoso
CISUC, Department of Informatics Engineering
University of Coimbra, Portugal
{jcgonc,pjmm,amilcar}@dei.uc.pt

Abstract

This paper presents BlendVille, a computational system based on the Conceptual Blending framework, where the search for the best blend is handled as an optimisation task by a Multi-Threaded (MT) Genetic Algorithm (GA). The new system departs from ideas explored in a previous framework, Divago. One of the most substantial differences from the latter lies in the usage of entropy to create new concepts with varying levels of information complexity. Additionally, Blendville explores the use of multiple analogies when projecting concepts into the blend space and in the evolutionary process. We investigate the behaviour of the new system, compare its output with its predecessor's and report on our findings.

Introduction

The Conceptual Blending (CB) theory (Fauconnier and Turner 2002) was proposed to explain mechanisms involved in the creation of meaning and insight in the every day mind. In the last years, one can witness an emergence of various computational systems based on CB with diverse origins, including international projects such as CoInvent (Schorlemmer et al. 2014) and ConCreTe (Żnidaršič et al. 2016).

This paper describes recent efforts, initiated within the latter project, towards the proposal of a new, written from scratch, computational approach to CB, which we named *BlendVille*. We build on the legacy of *Divago*, one of the first and most comprehensive implementations of CB,developed by Pereira (2005).

We start this paper with short overviews of both the Conceptual Blending theory and the Divago computational framework. Then, we expose our proposed blender, its inner workings and the optimality measures the system uses, after which we evaluate the impact of those measures on the system's output. Finally, we outline further work to improve our blender and conclude on our findings.

Background

Conceptual Blending (CB) was suggested as cognitive theory by Fauconnier and Turner (2002) to explain processes of conceptual integration occurring in human thought. Its potential to model mechanisms of concept invention has increasingly inspired research in Computational Creativity in recent years (e.g., the recent works by (Żnidaršič et al. 2016) in the ConCreTe project and (Schorlemmer et al. 2014) in the CoInvent project). A key element in the theory is the *mental space*, a partial and temporary structure of knowledge assembled for purposes of thought and action (Fauconnier and Turner 2002). The CB process takes two *input spaces* and looks for a partial *mapping* between elements of both spaces that may be perceived as similar or analogous in some respect. A third mental space, called *generic*, encapsulates the conceptual structure shared by the input spaces, providing guidance to the next step of the process, where elements from each of the input spaces are *selectively projected* into a new mental space, called the *Blend Space*. Further stages of the process elaborate and complete the blend.

As the input spaces can be blended in many forms, CB proposes a set of optimality principles to characterise good blends. These principles have a key role in the process and help to ensure that the resulting blend represents a coherent and integrated structure.

Another important notion is that of *frames*, which in the theory play a role in the structuring of mental spaces (Fauconnier and Turner 2002). Frames are mental structures that provide a kind of abstract prototyping of entities, actions or reasonings and may guide the process of blend construction to recognizable wholes (Pereira 2005). For instance, the mental space *Disney's Dumbo* encloses the idea of an elephant capable of flying, in which two invoked frames of thought are the frames *elephant* and *flight*.

Divago

Divago (Pereira 2005) is one of the first developed computational architectures based on the CB framework. It is composed of three major working modules (Fig. 1): the *Mapper*, the *Blender* and the *Factory*.

In Divago, input spaces are represented as computational versions of Conceptual Maps , i.e. graphs where nodes are concepts and arcs are relations between a source and a target concepts. The input spaces are stored as semantic networks in the form of text triples.

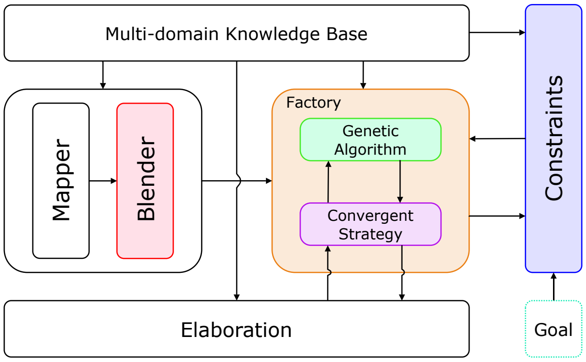

Figure 1: The architecture of Divago. The relevant modules to this paper are shown in colour. Best viewed in colour.

The *Mapper* computes analogy mappings between concepts from two input spaces, using a structural alignment algorithm based on Sapper (Veale and Keane 1997). The input spaces are in the form of semantic networks, same structure used by our blender. The mappings calculated are grouped in a single set and represent the analogy of concepts to be used by the Blender module. The analogy is formed by finding the largest (in cardinality) isomorphic sub-graphs of both input spaces. Once the isomorphism is found, aligned pairs of nodes from both subspaces are mapped together to produce an analogy mapping.

The *Blender* takes one analogy and performs a selective projection into the blend space, which leads to the construction of a *blendoid*, an intermediate graph that subsumes the set of all possible blends. This blendoid feeds a GA in the *Factory* module, which explores the space of all possible combinations of projections of the input spaces taking into account the generic space.

The Factory uses an implementation of CB optimality principles, intended to ensure a coherent and integrated blend, as fitness function of the evolutionary process (the *Constraints* module). When an adequate solution is found, the Factory stops the execution and returns the best blend.

A city of blends

The architecture of BlendVille is shown in Fig. 2, where the search for the best blend is handled as an optimisation task by a GA that evolves a population of competing blends. By comparing with Fig. 1, we can see that BlendVille roughly plays the role of the modules Factory and Constraints in Divago. Its inputs are (i) a list of input spaces in the same format as Divago, (ii) a list of analogies and (iii) a list of frames. The fitness of each blend is computed through a weighted sum of a number of measures, discussed later. The reason for using a GA is that the space to search for an optimal blend is highly complex. This is mainly due to the high variability in the semantic structure of a blend (relations, edge directions and concepts).

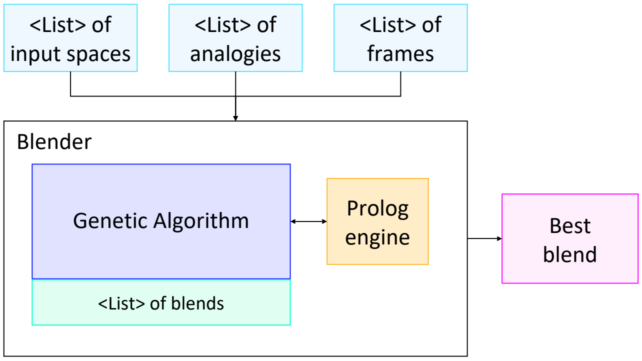

Figure 2: BlendVille's architecture and input knowledge. Best viewed in colour.

Three major differences between BlendVille and the Divago's Factory are: the usage of multiple analogies (sets of mappings) in the projection of concepts from the input spaces to the blend space; the investigation of different metrics/constraints for guiding the blending process; and the performance improvement of the blending process itself. The first two are discussed on a section of their own, further below.

To ensure a better performance, BlendVille uses a Multi-Threaded (MT) GA, allowing the evolution of a greater number of blends, the handling of complex semantic structures (input spaces, analogies, frames) and the parallel execution of multiple Prolog calls, used to check frame matching.

Input Spaces

The input spaces are represented as semantic graphs of concepts and relations (concept maps). Each input space graph represents a directed pseudo-graph, described as a set of triples $relation(concept_{source}, concept_{target})$, where both relations and concepts are text strings (arrays of characters). Our blender supports multiple input spaces, identified by their title (*horse*, *bird*, *boat*, etc.) which we term *domain* or *namespace*. In a declarative language (Prolog), the input space corresponds to a knowledge base of facts, where each fact is an edge of the graph, the predicate is the label of this edge (relation) and both concepts are the atoms of the fact. As an example, the sentence "Socrates is a man" corresponds to the fact *isa('Socrates','man')* and in the semantic graph is represented as an edge labelled *isa* going from the vertex *Socrates* to the vertex *man*.

Analogies

Also required is a set of analogies. Each analogy relating two input spaces is defined as a set of mappings $m_i = \{c_i, c_j\}$, each of which associating two concepts c_i and c_j, one from each space. In each analogy, a concept can only be present in one mapping. However, a concept can be mapped to different concepts in different analogies. We expect that this rationale will favour a higher blend diversity, when compared to a blender system which uses a single set of mappings (one analogy) as in Divago. Currently, we use an additional tool with an algorithm similar to Divago's Mapper to generate analogies from the input spaces.

Frames

Frames are handled in the form of either semantic graphs or logical clauses, compatible with the Prolog language. They allow the blend to be matched against specific composite concepts, situations, abstract ideas or similar cognitive contexts.

The supported frames are of three types, according to the elements (relations or concepts) matched in the blend and the existing elements present in one or more input spaces:

local frames compare the existence of elements between the blend space and the input spaces. When comparing relations, these are counted for a given label and must be connected to a concept (eg. count all the *isa,pw* relations connected to the *horse* concept) or in all the graph (count all the *isa,w* relations in the graph). The local frames ignore both edge transitivity (memoryless) and directionality, thus their naming. Examples of these frames are the Divago's aprojection/bprojection frames, whose purpose is to compare the existence of a custom set of concepts in the blend; and Divago's aframe/bframe which counts relations of a given mental space in blend. The frames nomenclature (a or b) is because each frame is related to a specific input space, eg. a to the *horse* and b to the *bird* input space.

pattern frames match the maximum number of concepts in the blend space, subject to the specified restrictions. With the usage of variables and atoms in a clause, the blender is able to detect a pattern of interconnected relations. The score for an individual pattern frame is 1 when the clause applies fully. Otherwise, Prolog iteratively chunks the frame clause's predicates in individual clauses of one predicate and counts the number of predicates which are successfully solved. Then, the score will be $k/(n-1)$, with k the number of successfully solved predicates and n the total amount of predicates of the frame. An example of a pattern frame is shown in Fig. 3.

delta frames are semantically equivalent to pattern frames with the difference that the instantiation of variables must be different between a mental space and the blend space. For instance, using the frame in Fig. 4 as a delta frame, the frame is maximised when the variables W, A, P are instantiated with different concepts from the blend space than those instantiated from the input space. The score for a delta frame is directly proportional to the amount of different instantiated concepts between the blend space and a mental space. When more than one instantiation for the variables is possible, the final score is the maximum of the individual scores. Our rationale is that in this case there is at least one frame which maximises the difference between the blend space and a mental space and as such, there is a pattern which differs between a mental space and the blend by having the greatest amount of concepts. An example of a delta frame used in the experiments is shown in Fig. 4, which detects entities W with ability A and part P with purpose A. Hence, this frame gives importance to the emergence of entities with parts whose abilities and purposes are different than the ones present in the combined input space.

```
ability(A,’bird/fly’), pw(’bird/wing’,A),
pw(’horse/leg’,A), ability(A,’horse/run’),
pw(A2,A1), pw(A1,A), purpose(A2,’horse/hear’).
```

Figure 3: Pattern frame of an A composed of a wing and a leg part, with the ability to fly and run, as well as a sub-part whose purpose is to hear.

```
ability(W,A), purpose (P,A), pw(P,W).
```

Figure 4: Delta frame *new_ ability* of an entity W with the ability A and with a part P performing the purpose A.

Blend chromosome

The chromosome of each blend is defined by a local analogy and a local blend space. Both analogy and blend space are stored in the chromosome and therefore specific to an individual blend. When the GA starts, the blend space of every chromosome is an empty space. Then, the analogy chunk of each chromosome is initialized as either a full copy or a random subset of the mappings contained in a random supplied analogy. The blender then selects a random subset of mappings from the blend's analogy. For each mapping m_i, one of four operations is then executed: extract the first concept c_i; extract the second concept c_j; ignore the current mapping; or create a blend (fusion) c_k of both concepts. Depending on the chosen concept(s), a nearby set of relations touching the concept(s) are pulled from the input space into the blend space. The name of a blended concept c_k is the concatenation of both names separated by an underscore: $c_k = c_{i0}\_c_{i1}$. During each epoch, the GA applies a mutation to every blend in the population. Afterwards, all the blends are evaluated according to a fitness function.

Blend mutation

The mutation is applied to a blend in two steps: mutation of the blend mappings and of the blend space. The set of mappings is mutated as follows: setting them to fully match one of the supplied analogies; the random removal of one or more mappings from the blend's analogy; and the random insertion of one (or more) mappings from the supplied analogies.

The mutation does not create mappings which do not exist in the supplied analogies. If this happened there was no reason to require a list of analogies as the blender would wildly conceive random mappings. On the other hand, these mappings would likely to flawed, as the blender omits semantic and structural knowledge related to the concepts present in the input space. As such, our blender assumes there is an external adequate algorithm which supplies the analogies.

The mutation of the blend space is divided in five steps, transforming its structure as follows:

- addition of edges from the input space, including new concepts;
- removal of edges and/or concepts from the blend space;
- inclusion of two blended concepts and relations linked with one or both concepts. When two concepts a and b are blended together, the new concept is named $concept_a|concept_b$, eg. *'horse/leg|bird/leg'*;
- inclusion of one concept c_0 from a random mapping m_r with some (or all) relations associated with the other concept c_1 of the mapping $m_r = \{c_0, c_1\}$, replacing in those relations one concept by the other. Concepts c_0 and c_1 may be randomly swapped;
- selection of a random concept in the blend space, changing it and its associated relations to an opposing concept, according to a chosen random mapping.

For each epoch, the above steps are independently and randomly applied to each blend of the current population. The presence of the mutation operator is sufficient (Tate and Smith 1993) to allow the emergence of a diversity of blends through the evolutionary process.

How enlightened is that blend?

Divago used six optimality principles: Integration, Topology, Unpacking, Maximisation/Intensification of Vital Relations, Web and Relevance (Pereira 2005). The optimality principles are used to exert pressure towards stable, consistent and integrable blends.

Martins et al. (2016) did a deeper analysis on both the impact and importance of each individual principle in achieving a "good blend". The authors, supported by empirical experiments, suggest that five principles: Integration, Topology, Unpacking, Relevance, and Intensification / Maximisation of vital relations - could be enough for achieving interesting blends.

Encouraged by the study, we decided on walking new grounds with the idea of *Simplicity Theory* (ST) (Dessalles 2013) in mind, with the aim of studying simpler and intuitive methods in conceiving interesting blends. In ST, Dessales asserts that the human mind is highly sensitive to any discrepancy in the complexity of information (Fig. 5). Motivated by the assumption that the human brain is sensitive to algorithmic complexity, the author alleges that the impact induced in people is proportional to how much simplicity is present in the information people are shown. This corresponds to the idea that much of cognition is regarding the compression or the elimination of redundancy (Chater and Vitányi 2003) in whatever information is the human brain processing. As entropy is related to the idea of information compression, our blender contains two forms of graph entropy as optimality measures. As such, the blender currently neglects most optimality principles of CB theory, accepting the fact that we may lose some consistency and coherence in the obtained blends. However, with the execution of experiments we expect conclusions to be made regarding a future study regarding a trade-off of some (or all) of theory's principles, perhaps including the study of new measures. Nevertheless, our blender requires other heuristics which either individually either combined, guide the blending process towards the direction of "good" blends.

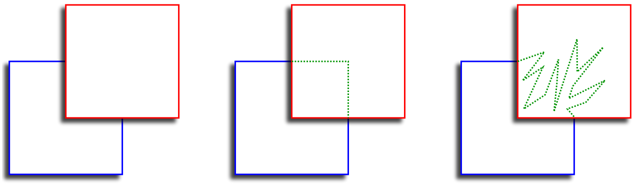

Figure 5: Simple and intricate interpretations for an occluded figure (blue). Best viewed in colour.

The assessment of the blends is done in four perspectives: topology, entropy, frame related and general informative measures. These are explained below:

Topology

We follow the definition of Topology used in Divago, where topology exerts a form of inertia in the blending process. For any mental space and any element in that space projected into the blend, it is optimal for the relations of the element in the blend to match the relations of its counterpart. A relation present in the blend space is topologically correct when it occurs in at least one of the mental spaces. The topology measure is defined as a ratio of topologically correct relations present in the blend space. The definition we used for topology is the same as defined in (Pereira 2005), section 4.2. For reference, it is given as follows:

Topology: for a set $TC \subseteq CM_b$ of *topologically correct* relations, defined as

$$TC = \{r(x, y) : r(x, y) \in CM_1 \cup \cdots \cup CM_n\}, \qquad (1)$$

where $CM_1 \cdots CM_n$ correspond to the concept maps of the n input/mental spaces. The topology measure is then calculated as the ratio:

$$Topology = \frac{\#TC}{\#CM_b}. \qquad (2)$$

Topology drives against change in the blend space because while preserving a similar topological configuration as the input space. According to its definition, the blend should preserve the same neighbourhood relations between every concept in the blend space.

Entropy

Various measures of graph entropy exist (Dehmer and Mowshowitz 2011). To measure one form of complexity (or compressibility) of the blend space, we implemented two entropy measures based on Shannon's entropy. These are calculated according to the relation labels of the blend space. The majority of the blend's concepts are unique and in turn, we decided to disregard entropy related to the concepts.

BlendVille contains two entropy measures for the blend's relations: 0-order and 1-order. The difference is due to the relevance given or not to sequences and the directivity of relations. We explain these below.

0-order entropy: this measure of entropy coincides with Shannon's classical entropy, defined as follows: given a discrete random variable R with n symbols $r_0, \cdots r_n$ and probability density function $P(R)$ defined for each symbol r_i, the 0-order entropy $H(R)$ is:

$$H(R) = -\sum_{i=1}^{n} P(r_i) \log_e P(r_i). \tag{3}$$

The random variable R corresponds to the set of all the relations (ie. R ={*isa*, *pw*, *ability*, ...}) present in the blend space. Accordingly, $P(r_i)$ is the relative frequency of the relation r_i in the blend space. Hence, the probability density function $P(R_i)$ of a relation r_i is defined as the ratio between that relation's label absolute frequency and all the relations present in the blend graph:

$$P(r_i) = \frac{F_{r_i}}{\sum_{j=1}^{n} F_{r_j}}. \tag{4}$$

The reason for this measure being named 0-order is because it corresponds to a stateless description of the blend space, having no interpretation on the sequences of relations. This measure applied to a fully connected or disconnected graph would naturally result in the same measured value. However, it allows one assessment of the redundancy or uniqueness of labels of the relations in the blend.

1-order entropy: This measure is defined as an extension to the 0-order entropy, with the difference that it takes into account pairs of consecutive relations $\{r_{i0}, r_{i1}\}$ and their directivity (Fig. 6). For instance, the relations connected (through the horse concept) *isa(horse,animal)* and *pw(horse,leg)* generate the relation pair $\{isa, pw\}$. The calculus of the 1-order entropy is done analogous to the steps above, with summations adapted to include pairs of connected relations and their directivity, instead of single relations. The directivity of the pairs is defined as related to a common concept: if in the same direction; incoming; or outgoing to the concept (Fig. 6).

Frame evaluation

To have a meaningful interpretation and purpose, BlendVille uses frames to define the content and context of the blend, conforming to CB's view of mental spaces. The types of frames exposed in the previous section *Frames* are evaluated and given individual scores according to their type: *concept frames* (for a given mental space), *edge frames* (also for a given mental space), *delta frames* and *pattern frames*.

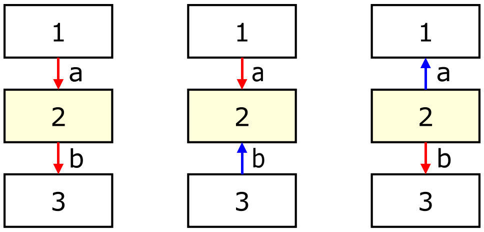

Figure 6: 1-order entropy patterns for the pairs of relations $\{a, b\}$: in the same direction, incoming or outgoing to the common concept "2". Best viewed in colour.

In the case of multiple delta Frames and/or pattern Frames, our blender does not currently prioritise one delta or pattern frames over another. We expect this to be improved in the future. In the situations where there are multiple delta/pattern frames to evaluate, the system gives a score to a frame proportional to the matching of each individual frame's predicates (range $[0 \cdots 1[$ or 1 if the frames matches fully) and proportional to the number of applicable frames.

General Informative

The last two measures are not related to semantics but are used to fine-tune the global structure of the blend graph and the relative contribution of the input spaces. These are the number of *graph islands* and the amount of *inter-space edges*.

graph islands counts what in graph theory is defined as the number of connected components (islands) in the blend space. It is calculated in linear time using breadth first search for all the concepts in the graph.

inter-space edges is the number of relations present in the blend space which connect concepts of distinct mental spaces. An example is the relation *pw('bird/wing','horse/horse')*. The exception is when a blended concept of different mental spaces, such as *'horse/leg|bird/leg'* is present in the relation. In this case, as the relation associates concepts of the same mental space as subject and object, we consider the relation as not inter-space.

Novelty and Usefulness

For assessing the quality of the generated blends and validating our blender against Divago, we used the same measures *novelty* and *usefulness* as defined by Pereira, themselves based on the work by Ritchie (2007). Ritchie considers *typicality* and *value* where Pereira defines *novelty* as the opposite of *typicality* and *usefulness* as a synonym of *value* (Pereira 2005). Both measures are defined next:

novelty describes a measurement of non typicality, surprise, a change in information. Novelty of a blend, which Pereira describes as the "*converse of Ritchie's typicality function*", is defined as a function of $d(b,x)$ and the size of the blend $size_b$.

Let x be one of the n input spaces and b the blend space. Then, $d(b,x)$ is "*the sum of the relations that:* 1) *belong to* b *and that are missing in* x *with* 2) *those that belong to* x *and are missing in* b". Next, an edit distance $distance(b)$ is defined as:

$$distance(b) = \frac{min(d(b,x_1),\cdots,d(b,x_n))}{size_b}. \quad (5)$$

The measure novelty of the blend b is calculated as:

$$novelty(b) = \begin{cases} 1 & distance(b) > 1 \\ distance(b) & \text{otherwise} \end{cases}. \quad (6)$$

usefulness evaluates the blend according to a purpose. In Divago, the purpose was defined before an experiment, after which the usefulness of the blend is assessed. Furthermore, the purpose is defined as the blend having an exact similarity (both structural and semantic) to a specific conceptual map (semantic network).

Given a conceptual map t and a blend space b, $d(b,t)$ is the sum of relations that belong to t and are missing in the blend b. Then, usefulness is:

$$usefulness(b) = 1 - \frac{d(b,t)}{size_b}. \quad (7)$$

Next, we experimented our blender in various situations and assessed its results with the above measures. These experiments are exposed in the next section.

Experiments and discussion

The experiments were carried out with the conceptual maps *horse* and *bird*, available in Pereira's PhD thesis (Pereira 2005) in Tables 5.1 and 5.2 of page 100. The three analogies (sets of mappings) for the *horse* and *bird* experiment are also found in the same thesis, in Fig. 4.5, page 110. The *delta frame* used for detecting a new ability is given in Fig. 4. For the usefulness, the target blend was set as a conceptual map representing the *Pegasus*. The Pegasus is defined as containing the same conceptual map of the *horse* to which *two wings* were added as well as the *ability* to *fly*. Therefore, the five following relations were added:

ability('horse/horse', 'bird/fly'),
pw('bird/wing', 'horse/horse'),
quantity('bird/wing', '2'),
purpose('bird/wing', 'bird/fly'),
motion_process('horse/horse', 'bird/fly').

Testing the measures

The measures were evaluated either individually or combined. Their impact and validity in novelty and usefulness are described next. Unless otherwise noted, all experiments minimised the measure *number of graphs islands* (components) in order to obtain a fully interconnected blend space.

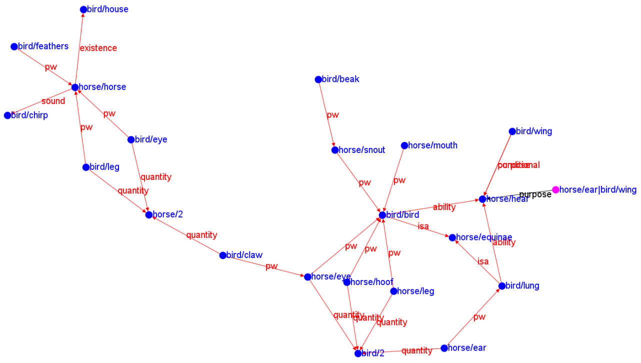

Figure 7: Example of a blend with low 1-order entropy. Edges in red are inter-space relations. Best viewed in colour.

Topology

When we used exclusively this measure as the fitness function, the blend ended as a projected copy of the input space. Although one could expect novelty to be inversely correlated with topology, the novelty measure - as defined by Pereira - is specific to the exact structure (semantic and relational) of the input spaces. On the other hand, topology is defined on the relations directly connected to each concept of the blend space. Therefore it does not guarantee topology at a higher structural level. We observed that when the blend's topology had in average score of 100%, novelty was contained in the interval of $[95\%, 100\%]$. This happened as a result of the stochastic nature of the GA, which allowed the blends to randomly evolve in both the blend space and the set of mappings. On all the topology experiments there was at least a dozen of inter-space relations connecting concepts of different mental spaces. By itself, this is enough to maximise novelty.

On the other hand, usefulness was between 25% and 50%, which somewhat demonstrates that on its own, there is no relation between topology and usefulness. This is expected, as the above definition of topology has no reason to justify the emergence of a *Pegasus* blend, even though the *Pegasus* mental space is defined as the union of the *horse* mental space with five specific relations of the *Pegasus*, with the particularity that 90% of *Pegasus*' relations are from the *horse* input space. This demonstrates that topology reflects a statistical description of the relations labels and ignores the global structure of both the blend and input spaces.

Entropy

These measures had no impact in the semantic structure of the blend space. However, they did have a definitive influence in the compression of the blend, as well as an effect in the presence of redundant structures.

- **0-order entropy** directly affects the variety of relations of each type in the blend. This allows two limits: when 0-order entropy is maximum the blend has the highest amount of unique relations of a certain label (1×*isa*, 1×*pw*, 1×*ability*, etc.) and the opposing case,

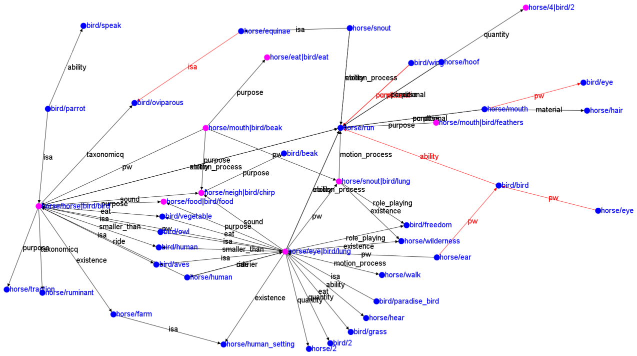

Figure 8: A blend with high 1-order entropy. Edges in red are inter-space relations. Best viewed in colour.

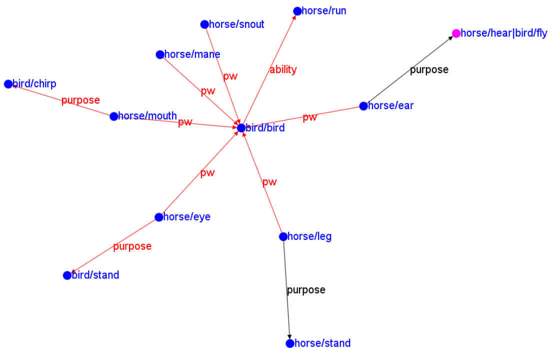

Figure 9: Example of a blend defined by the pattern frame in Fig. 3. Best viewed in colour.

when entropy is minimum the blend has only relations of one label, ie. $1 \times isa$ or $10^{99} \times isa$. However, 0-order entropy does not affect the total amount of relations present in the graph, but in fact the relative amount of relations of each type. Given a blend with 2 *isa* and 5 *pw* relations, its 0-order entropy is the same as another blend with 20 *isa* and 50 *pw* relations, or in another words, the blend is allowed to contain n groups of $\{2 \times isa \cup 5 \times pw\}$ relations without affecting the value of the measure. This is because 0-order entropy is defined on the *relative* probabilities $P(X)$ of the discrete variable X (the labels of the relations) occurring in the blend.

- **1-order entropy** - This measure affects the variety of sequences of relations present in the blend space. For instance, if a certain blend has the relation pattern pw(A,B) and $quantity$(A,C), this pattern is allowed to materialize elsewhere in the blend space without affecting 1-order entropy (Fig. 7). Therefore, this usage of this measure allows the appearance of repeating structures in the blend space or, on the other hand, the manifestation of diversified structures in the blend (Fig. 8).

An understandable observation is that the entropy measures have no obvious correlation with novelty, much less with usefulness. Even when the blend's entropy is equal to one or more of the input spaces (or the conceptual map of the Pegasus), that equality would not imply a structural and semantic similarity between those spaces. This is expected as entropy is a statistical description of information.

Frames

The local frames (a/bprojection) bring the concepts of their input space into the blend space. Similarly, the a/bframes allow the blend space to have a statistical distribution of relations equal to their specific input space. Thus, achieving a useful blend in the context of the Pegasus concept map requires the *aprojection* and *aframe* frames to be present in the blend, in order to emerge the structure of the horse mental space in the blend space. The inclusion of the delta frame *new_ability* in Fig. 4 allowed the blend to differ from the input spaces, materialising a new ability in the blend space which did not existing in any of the input spaces, increasing the blend's novelty.

The inclusion of the horse concepts (aprojection), horse relations (aframe), delta frame *new_ability* and topology enabled the creation of blends with a usefulness of 93%...95%. However, without more elaborated forms of expressing the frames required to represent the Pegasus mental space, it is not possible with our current blender to obtain the Pegasus conceptual map. This also exposes what we consider is an issue with the definition of usefulness.

Adding pattern frames to the fitness function favoured the manifestation of blends with more elaborated semantic interpretations. Using the pattern frame in Fig. 3, the delta frame *new_ability*, maximising the number of inter-space relations while minimising the 1-order entropy allowed the blender to generate interesting blends, one of such blends is shown in Fig. 9. That blend shows is a bird with a mouth of a horse, able to chirp, eyes used to stand up and with a horse mantle. Intriguing to note that the ears are used to fly instead of the wings, being these used to run. In separate experiments we witnessed blends that represented animals which could hear using their wings, being these attached to their snout. Kind of giving a new purpose to the definition of wing... in the form of a tympanic membrane.

Graph Islands

This measure is useful in creating strongly connected blends. It allows the penalisation of blends whose spaces have disconnected components, ie., with no detached relations "floating" around.

Inter-Space edges

Reinforcing the number of inter-space relations in the blend space tends to maximise novelty, as the blend space becomes filled with a mixture of connected concepts from different input spaces. In the horse-bird experiment which has two input spaces, the blend space ends with roughly 50% of the concepts of each input space. As novelty measures the amount of missing relations (and related concepts) from each input space, the emergence of relations connecting concepts from different input spaces naturally increases novelty.

Comments on Novelty and Usefulness

We can confirm that *novelty* as defined in Divago does indeed measure a modification of semantic structures in the blend, when compared to the input spaces. However, it only measures a total mismatch of a relation, not having into account, for example, when in a relation the only change from an input space to the blend space is the modification of a single concept, or of the label of the relation itself. We believe novelty should be proportional to the minimal possible change in the representation of the blend space.

We agree with Ritchie (2007) regarding usefulness. It should not be defined strictly according to a purpose, but in a more general and fuzzier perspective. This is the main reason why during the pegasus experiment our blender was not able to reach a score of 100% in usefulness. We believe that the change in usefulness should correspond to Ritchie's definition of *value*, which rates the worth or importance of the newly created artefact, not necessarily before its creation.

Future Work

We expect a great deal of work to be done. We envision the improvement of the measures *novelty* and *usefulness*. Regarding the entropy we expect at least two developments: multiple orders of entropy which will allow the emergence of structural redundancy at various levels; and the involvement of both semantics and frames in the calculation of entropy. The exploration of different types of frames, as well as the latest developments in image schemas, is also expected to be pursued.

Conclusion

In this work we proposed an evolutionary system modelled on CB theory. The blender implementation - BlendVille - receives input spaces, frames, analogies and outputs blends capable of displaying novelty. We feel that the system exhibits a form of creativity. Our system has its roots on Divago but follows a different recipe regarding the use of optimality principles in order to generate understandable, independent and coherent artefacts. The main ingredient is based on information theory, mainly the concept of entropy. Therefore, we think our system allows the emergence of a form of redundancy in the generated blends, in accordance with the idea present in Simplicity Theory.

Acknowledgements

This research was funded through EC funding for the project ConCreTe (grant number 611733) that acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission.

References

Chater, N., and Vitányi, P. 2003. Simplicity: a unifying principle in cognitive science? *Trends in cognitive sciences* 7(1):19–22.

Dehmer, M., and Mowshowitz, A. 2011. A history of graph entropy measures. *Information Sciences* 181(1):57–78.

Dessalles, J.-L. 2013. Algorithmic simplicity and relevance. In *Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence.* Springer. 119–130.

Fauconnier, G., and Turner, M. 2002. *The Way We Think.* New York: Basic Books.

Martins, P.; Pollak, S.; Urbančič, T.; and Cardoso, A. 2016. Optimality principles in computational approaches to conceptual blending: Do we need them (at) all? In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016).* Paris, France: Sony CSL.

Pereira, F. C. 2005. *Creativity and AI: A Conceptual Blending approach.* Ph.D. Dissertation, University of Coimbra.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Schorlemmer, M.; Smaill, A.; Kühnberger, K.-U.; Kutz, O.; Colton, S.; Cambouropoulos, E.; and Pease, A. 2014. Coinvent: Towards a computational concept invention theory. In *Proceedings of the 5th Int. Conference on Computational Creativity, ICCC-14, Ljubljana, Slovenia.*

Tate, D. M., and Smith, A. E. 1993. Expected allele coverage and the role of mutation in genetic algorithms. In *ICGA*, 31–37.

Veale, T., and Keane, M. 1997. The competence of sub-optimal structure mapping on hard analogies. In *Proceedings of the International Joint Conference on Artificial Intelligence.* IJCAI-97.

Żnidaršič, M.; Cardoso, A.; Gervás, P.; Martins, P.; Hervás, R.; Alves, A. O.; Oliveira, H.; Xiao, P.; Linkola, S.; Toivonen, H.; Kranjc, J.; and Lavrač, N. 2016. Computational creativity infrastructure for online software composition: A conceptual blending use case. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016).* Paris, France: Sony CSL.