Natural Language Generation from Concept Blends

Francisco C. Pereira*
* Creative Systems Group AILab - CISUC
Departamento de Engenharia Informática da Universidade de Coimbra
Pinhal de Marrocos 3030 Coimbra
camara@dei.uc.pt

Pablo Gervás†
†Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid
pgervas@sip.ucm.es

Abstract

Conceptual Blending (CB) is a framework for the integration of knowledge from different sources onto one, known as the *blend*. For example, the concept "edutainment" is the blend of "education" and "entertainment". This paper presents an approach on producing a natural language *reading* of the blends that are generated by Divago, a system that applies a computational implementation of CB to transform a search space. Represented through a semantic network (the concept map), the blend demands considerable effort in its interpretation. Obtaining its Natural Language description raises several issues and is an interesting challenge for the field of Natural Language Generation. In this paper, we propose a reading based on diagnostic features with respect to a reference concept. We make some reflections on the choice of these references, namely regarding the creative potential associated.

1 Introduction

In this paper, we propose a methodology for producing Natural Language texts from knowledge structures called *blends*, which are generated from the combination of other knowledge structures called *input spaces*. These structures (the blend and the input spaces) are represented as concept maps. A concept map should represent the explanation of one or more main concepts by means of a graph structure having directed arcs connecting pairs of nodes, where each arc represents a relation and each node a concept. For example, we can explain the concept of "eating" by association to other concepts like "digesting", "absorbing", "food", "energy" or "survival". Or a concept like "bird" by association to "wings", "feathers", "beak" and so on. From the point of view of Natural Language Generation (NLG), an immediate contribution from concept maps can be as an explanatory core for the lexicon that allows associative reasoning, like metonymy or metaphor. The non-literality emerging from these and other linguistic devices is, we think, an important brick for creativity in language.

The approach we are currently following is based on the principle that, when communicating a new concept in natural language, it is common to use previously established concepts as a support. For example, for describing the concept of "laptop computer", we do not have to enumerate all the specific features of computers in general, instead we take them all for granted (by just saying that "it is a computer") and focus the differentiating features (being portable, light, using a battery, etc.). This idea for reading blends is thus based on finding *salient* (Milosavljevic and Dale, 1996) or *diagnostic* (Costello and Keane, 2000) features and using them to describe the concept.

Conceptual Blending (Fauconnier and Turner, 1998) is a framework for integration of knowledge (e.g. "digesting a book" comes from an integration of "eating" and "reading") and we will explain it in some detail in section 2.2. It should be clear, though, that we are proposing the *reading* of the *blends* that are generated by our computational model of blending, not blends in general or generated by whatever forms the framework of Fauconnier and Turner may take. For our purposes, the first and most important notion to catch now is that of a *domain*, which is represented primarily by a concept map. Therefore, *reading a concept map* is the necessary first step towards *reading a blend*. The other step of exploration is that of *reading instances*, and we should point out that we have already done some work and experiments at this level[1], which took the form of exploring visual constructions obtained from a blended instance, instead of natural language descriptions.

In the next section, we give more information regarding the blending mechanism, after which we dedicate a section to describing existing work on the subject of comparing and describing objects. The general idea of our NLG system is presented in section 4. As the reader will notice, this is clearly a work under development, as also confirmed in the examples shown. We finish some discussion of important issues to be considered.

2 The Blending Mechanism

A detailed description of our implementation of Conceptual Blending is beyond the scope of this paper, however a

[1] See the "house-boat visual blending experience" (Pereira and Cardoso, 2002a)

few basic notions about the knowledge representation formalism and the algorithm for obtaining blends are needed to understand the general idea of the paper.

2.1 Concept Maps

Before entering this section, we should clarify that, from the point of view of AI, a concept map is "just" a semantic network, a rather classical knowledge representation structure, in general terms. Yet, we would like to provide the background and philosophy that lead us towards this choice as well as some specific methodology we are applying.

Semantic networks have been used for generating discourse from early on (Simmons and Slocum, 1972; Shapiro, 1982). Novak and Gowin (1984) were the first to focus concept maps under a study on education and communication in humans and they considered them a very intuitive and commonly used graphical representation of knowledge structure in domains. A concept map is a directed graph in which arcs connect pairs of nodes, each node corresponding to a concept, each arc to a relation. This definition covers a very big set of formalisms and applications we can find in AI, namely semantic networks, ontologies – e.g. WordNet (Miller, 1995) –, Semantic Web (Berners-Lee and Miller, 2002), among others. In our current focus, the Divago project (Pereira, 1998), we use concept maps to represent a *domain theory*, the explanation of the concepts of a domain. In principle, our notion of domain is not restricted to any pre-defined abstraction, but to the *user*'s choice, being any concept map she builds around a theme or goal. Yet, for normalization purposes, we are using an extra constraint: all the relations must come from the hierarchies of the Generalized Upper Model (Bateman et al., 1995), a general top-level ontology that has two hierarchies (elements and relations) that comprise abstract relations, properties, spatial relationships, among others. Figures 1 and 2 show an extract of the concept maps of the domains of horse and bird . Each relation is described by a predicate `rel(D, C1, R, C2)`, where the relation `R` is a labelled arc from the concept `C1` to the concept `C2` in domain `D`.

In comparison to Sowa's Conceptual Graphs (Sowa, 2000), it is clear that the language we use is much more limited: we only allow the valence of 2 arguments (every relation is binary), there is no possibility for specifying a quantifier for each relation, context or descriptor, to name a few differences. This makes clear that making the natural language reading of Conceptual Graphs (CGs) is more straightforward (if the full potential of these is used) than in our concept maps, given the ambiguity that may arise in these.

It is important to say, though, that our specific choice for the concept map representation as opposed to CGs is much more attached to its simplicity (with regard to the blending process) than to any other aspect. Yet, we are currently studying and implementing further evolutions to our representation, namely by allowing quantifiers and context specifiers.

```
rel(horse, horse, existence, wilderness).
rel(horse, farm, isa, human_setting).
rel(horse, house, isa, human_setting).
rel(horse, snout, pw, horse).
rel(horse, mane, pw, horse).
rel(horse, tail, pw, horse).
rel(horse, leg, pw, horse).
rel(horse, leg, purpose, stand).
rel(horse, hoof, pw, leg).
rel(horse, horse, taxonomicq, ruminant).
rel(horse, horse, eat, grass).
rel(horse, horse, ability, run).
rel(horse, horse, carrier, human).
rel(horse, human, ride, horse).
rel(horse, horse, purpose, cargo).
rel(horse, horse, purpose, traction).
rel(horse, horse, purpose, food).
rel(horse, mane, color, dark).
rel(horse, mane, size, long).
rel(horse, mane, material, hair).
rel(horse, hoof, quantity, 4).
rel(horse, leg, quantity, 4).
rel(horse, eye, pw, snout).
rel(horse, ear, pw, snout).
rel(horse, mouth, pw, snout).
rel(horse, eye, quantity, 2).
rel(horse, ear, quantity, 2).
rel(horse, eye, purpose, see).
rel(horse, ear, purpose, hear).
rel(horse, mouth, purpose, eat).
rel(horse,horse, sound, neigh).
rel(horse,horse,motion_process,walk).
```

Figure 1: Extract of facts from the *horse* domain

2.2 Conceptual Blending

In the simplest version of Conceptual Blending, as described by Fauconnier and Turner (1998), four distinct *mental spaces* are considered: the *generic space*, which should contain the generic background knowledge of a cognitive agent; two *input spaces*, which should comprise specific knowledge (e.g. the concepts that define horse and bird) that is to be integrated in the blend; and the *blend*, which contains a (potentially) novel concept (or set of concepts), and has a structure that results from the other three spaces and from its own emergent logic. For example, the fictional character of Count Dracula can be analysed as resulting from the blend of "vampire bat" and "evil person"[2], thus having properties and structure from both input spaces. Yet, while giving *life* to the character, some new features emerge, like "hates garlic" or "only dies if his heart is pierced with a wooden stake". This process they call "running the blend". The process of generating the blend demands a mapping correspondence between elements from the input spaces, which, according to a set of principles (the optimality constraints) are subject to be transferred (projected), potentially bringing

[2] We do not argue this is how Dracula was imagined, yet it serves as an example of the blending process.

```
rel(bird, bird, smaller_than, human).
rel(bird, lung, pw, bird).
rel(bird, lung, purpose, breathe).
rel(bird, paradise_bird, isa, bird).
rel(bird, bird, ability, fly).
rel(bird, feathers, pw, bird).
rel(bird, beak, pw, bird).
rel(bird, beak, purpose, eat).
rel(bird, bird, sound, chirp).
rel(bird, eye, pw, bird).
rel(bird, bird, existence, house).
rel(bird, bird, existence, wilderness).
rel(bird, eye, quantity, 2).
rel(bird, wing, quantity, 2).
rel(bird, claw, quantity, 2).
rel(bird, wing, pw, bird).
rel(bird, wing, conditional, fly).
rel(bird, wing, purpose, fly).
rel(bird, claw, purpose, catch).
rel(bird, leg, pw, bird).
rel(bird, leg, purpose, stand).
rel(bird, claw, pw, leg).
rel(bird, leg, quantity, 2).
rel(bird, parrot, isa, bird).
rel(bird, parrot, ability, speak).
rel(bird, bird, role_playing, freedom).
rel(bird, straw, pw, nest).
rel(bird, oviparous, lay, egg).
rel(bird, bird, purpose, pet).
rel(bird, bird, purpose, food).
rel(bird, eye, purpose, see).
rel(bird, beak, purpose, chirp).
rel(bird, bird, motion_process, fly).
```

Figure 2: Extract of facts from the *bird* domain

attached their surrounding structure. This projection is selective in the sense that some elements are projected to an *identical* existence in the blend, some are projected to a *different* existence (normally their counterpart in the mapping) and some are not projected at all. In the Dracula example, the "big canine teeth" are projected to "big canine teeth", the "bat wings" are projected to the "man's cloak", and the "sonar" is not projected at all.

In our computational model of CB, currently under development, we apply our implementation of those processes and principles (Pereira and Cardoso, 2001, 2002a,b, 2003a) to generate the blend. Each of the spaces is called a domain, and is represented via a concept map, rules, frames, integrity constraints and a set of instances. For the present purposes, the reader need only imagine each domain as being a concept map.

In figure 3, we show a diagram with the four spaces involved in the example we use in this paper: the blend of a "horse" and a "bird". The reader must understand this is a very simplified diagram (with only three concepts for each domain) in order to make it readable. Figures 1 and 2 have already given an idea of the input spaces, while figure 4 presents a (very small) extract of the generic space, which is expected to contain domain-independent knowledge such as *isa* taxonomies, integrity constraints[3] and

[3]The example shows the Divago notation for two: "X cannot have two different shapes at the same time", and "X cannot exist in two different quantities"

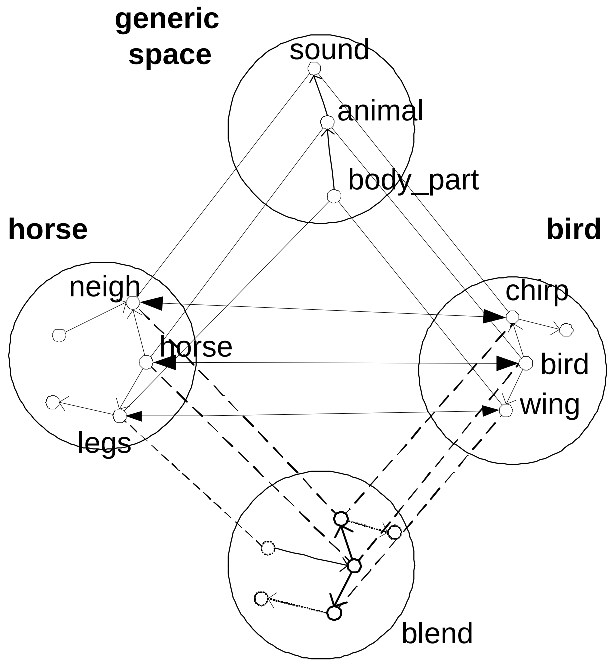

Figure 3: Example of a Conceptual Blending of horse and bird

frames[4].

It should be obvious that, given the mappings, there is potentially a multitude of projection combinations in the blend (e.g. "a horse that chirps and has wings", "a bird that neighs and has legs", "a horse that neighs and has legs", etc.). This explosion is controlled by the constraints (Pereira and Cardoso, 2003a), yet the search space is very large and complex so we apply a *Genetic Algorithm* to assure a parallel, non-deterministic search mechanism. Thus, we obtain a variety (though considerably reduced when compared to the unconstrained version) of different results (Pereira and Cardoso, 2003b).

In order to get an objective measure on the novel associations originated in the blend, the system finds the two lists of distinctive facts, one for each input domain. Each one of these lists, known as *extra* lists, contains basically all the relations in the blend that do not exist in the domain. Conversely, and also to assess the difference between the blend and the input domains, there are also *missing* lists (also one for each input), which contain the relations that are in the domain, but do not exist in the blend. As the reader will see, these lists are also fundamental for the NLG readings we are discussing here.

In figure 5, we can see an example of a generated blend (in this case, identified as being the individual *10210* of the genetic algorithm run) of "horse" and "bird" and, in figure 6, we show the respective extra and missing lists with respect to the "horse" domain. The syntax of these lists is straightforward: each element has the form `X/R/Y`, where `R` is the relation and `X` and `Y` are the pa-

[4]One is shown in the example, which can be interpreted as: " In the Blend, Concept X has an ability A, which didn't exist before in X's original domain, D1. It is P, a subpart of X, which enables this ability".

```
rel(generic,entity,isa,something).
rel(generic,situation,isa,something).
rel(generic,temporal_entity,isa,entity).
rel(generic,spatial_entity,isa,entity).

                          ⋮

rel(generic,person, isa, human).
rel(generic,human, isa, animal).
rel(generic,animal, isa, living_entity).
rel(generic,speed,isa,property).
rel(generic,size,isa,property).
rel(generic,color,isa,property).
rel(generic,weight,isa,property).
rel(generic,behavior, isa, property).
rel(generic,heavy, isa, weight).

                          ⋮

integrity(generic, [shape(X, Y),
                    shape(X, Z),
                    {Y\=Z}],[]).

integrity(generic, [quantity(X,Y),
                    quantity(X,Z),
                    {Y\=Z}],[]).

                          ⋮

frame(generic, new_ability(D1),
      [ability(X,A),purpose(P,A), pw(P,X),
      {current_blend(Blend),
       projection(Blend,D1,X,X),
       other_input_domain(D1,D2),
       projection(Blend,D2,A,A)},
       op(exists([projection/D1/X/X,
       projection/D2/A/A]))],
      [{rel(D1,X,ability,A)}],
      [new_ability(X, A)], []).
```

Figure 4: The generic domain

```
rel(ind10210, tail, pw, horse)
rel(ind10210, leg, pw, horse)
rel(ind10210, leg, purpose, stand)
rel(ind10210, hoof, pw, leg)
rel(ind10210, horse,taxonomicq, ruminant)
rel(ind10210, horse, eat, grass)
rel(ind10210, horse, ability, run)
rel(ind10210, horse, carrier, human)
rel(ind10210, human, ride, horse)
rel(ind10210, horse, purpose, cargo)
rel(ind10210, horse, purpose, traction)
rel(ind10210, horse, purpose, food)
rel(ind10210, mane, color, dark)
rel(ind10210, mane, size, long)
rel(ind10210, mane, material, hair)
rel(ind10210, hoof, quantity, 4)
rel(ind10210, leg, quantity, 4)
rel(ind10210, eye, pw, snout)
rel(ind10210, ear, pw, snout)
rel(ind10210, mouth, pw, snout)
rel(ind10210, eye, quantity, 2)
rel(ind10210, ear, quantity, 2)
rel(ind10210, eye, purpose, see)
rel(ind10210, ear, purpose, hear)
rel(ind10210, mouth, purpose, eat)
rel(ind10210, horse, sound, neigh)
rel(ind10210,horse, motion_process, walk)
rel(ind10210, horse, ability, fly)
rel(ind10210, wing, purpose, fly)
rel(ind10210, wing, pw, horse)
```

Figure 5: Blend generated for *horse* and *bird*

```
Extra:
[wing/conditional/fly, wing/pw/horse,
horse/motion_process/fly, horse/ability/fly,
wing/purpose/fly]
Missing:
[]
```

Figure 6: Extra and missing lists for the *horse* domain

rameters – e.g. the element `wing/conditional/fly` in the extra list means the "wing" in the blend has a "conditional" relation with `fly`, a relation that did not exist in the "horse" input domain, i.e. there is no `rel(horse, wing, conditional, fly)`.

We are thus faced with a fundamental problem: how to "read" the results? In a previous work, we have already approached the visual reinterpretation of the instances according to the new associations. We applied CB to the domains of "houses" and "boats", and re-generated the drawings we had for instances of those domains (e.g. "A boat has a window", being the representation of a "window" given by a square, implied the drawing of the "boat" would have a "hatch" of square form instead of round).

Now, we are interested in the understanding of the concept maps themselves via natural language descriptions.

3 Generating Descriptions and Comparisons

Two lines of existing research had to be taken into account before attempting the task outlined above: work on establishing comparisons between objects at the conceptual level, and work on generating natural language descriptions of objects from their conceptual definition. In some cases, actual research work had brought together both endeavours.

3.1 Establishing Comparisons between Objects

In their C3 model of Conceptual Combination (Costello and Keane, 2000), the authors present three constraints (*plausibility*, *diagnosticity* and *informativeness*) for the interpretation of noun-noun compounds. It is not within the scope of this paper to explain or compare them to our approach, yet we would like to retain diagnosticity for the present analysis. This constraint consists on identifying the differentiation (or diagnostic) aspects of a concept in comparison to its neighbouring concepts. For example, the diagnostic features of a cell phone are that it is a phone that is (normally) very small, wireless and one can speak with it if we are within the range of the provider's "cell". In this case, non-diagnostic features could be its colour, having a keyboard or being a communication device, which could at best say it is a phone. The idea of diagnosticity has much in common to the work of Milosavljevic (1997a,b) around salience properties of concepts. An interesting observation on salience is that it is dependent on the user-model (geographic location, level of expertise in the domain, etc.), thus differentiating properties depend on the relations users have with the concept (e.g. describing a "llama" to two kids by comparison to the animals they already know may bring salience to different sets of properties). We think the philosophy behind salience and diagnosticity is the same, yet its realization is different. In Milosavljevic (1997a,b), the authors hardcode salience ranking (some properties should be more salient than others, according to user models), as well as many other features of the system, which was justifiable given the specific application they had in mind, a dynamic hypertext system (Peba-II) for description of animals. In Costello and Keane (2000), the domain is that of nouns in general, and so finding diagnostic properties was done by automatic mining the corpus of all nouns in memory, not giving special value to specific properties or relations. Still, we think, if the nouns were animals that a kid knew about (as in the example just given), the results would be similar to those of Peba-II, and therefore diagnosticity would be the same as salience. An important feature in Milosavljevic's work, which is directly related to our present concern, is that Peba-II can describe an unknown entity, the *focused entity* (e.g. an echidna), by comparison to a known entity, the *potential confusor* (e.g. a porcupine) generating a text that enhances the *common salience properties* and the *contrasting salience properties*.

3.2 Generation of Descriptions in Natural Language

Research on natural language generation over the years has come to propose a pipelined architecture (Reiter, 1994) as the simplest engineering solution to generate texts meant to convey information. This solution is not optimal and the generation of other types of texts calls for different architectures (DeSmedt et al., 1995; Beale et al., 1998). However, for the present purposes, the modularity presented by a pipeline architecture outweighs any other disadvantages that it may have.

The pipelined architecture establishes a number of basic tasks to be carried out when generating a natural language text: *content determination* - finding what to say -, *document structuring* - organising what is to be said - , *sentence aggregation* - grouping together the parts that allow it -, lexicalization - selecting the words that will realize each concept -, *referring expresion generation* - choosing the right expression to refer to each element in its actual context - and *surface realization* - turning the result into a linear natural language sentence. These tasks tend to be grouped into bigger modules that operate in sequence over the initial input: a *text planning* component that deals with content determination and document structuring, a *sentence planning* component that deals with aggregation, referring expression generation and lexicalization, and a *surface realization* component. In this paper we focus particularly on text planning, in the sense that the innovative contribution lies in the way conceptual blending is applied to solve part of this task. The rest of the tasks are solved by resorting to state of the art solutions, in most cases by simple expeditive methods borrowed from existing work.

Natural language techniques have been applied in the past to generate textual descriptions of data that existed in a digital form that was not easily readable by humans, for domains like describing object-oriented models (Lavoie et al., 1997), information stored in a museum database (Dale et al., 1998), or time series data such as metheorological reports (Sripada et al., 2001).

Of particular interest is the body of work on natural language generation from conceptual graphs (Velardi et al., 1988; van Rijn, 1992; Smith et al., 1994; Svenberg, 1994; Wagner et al., 1995; Nicolov et al., 1996). These efforts take advantage of the well defined operations of conceptual graphs to explore the task of generation. Although some of the insights obtained in this way are applied in the present paper, the differences in complexity between this knowledge representation formalism and the concept maps used here required that further exploration of possible connections be postponed until a basic generation capability was integrated with Divago's process of conceptual blending.

For the current endeavor, an existing surface realization module, FUF/SURGE, (Elhadad, 1993) has been used to take care of the final stage. This delegates the task of dealing with the linguistic knowledge required to linearize and realize the text plans resulting from the previous stages into their final form as natural language sentences. The work presented here deals therefore with the task of plannning an NL realization for a given concept map.

The communications between the different modules involved takes place in terms of the generic data structures defined for generic architectures for NLG systems in the context of the RAGS project (The Rags Project, 2001).

4 Reading Concept Blends

In our first approach, we generated a reading by transcribing individually each relation of the resulting concept map. The resulting factual description of a domain, a list of sentences, with one sentence for each fact, made arid reading. We thus arrived to the conclusion that, unless with intensive knowledge and processing, this method was bound to arrive at very unnatural outcomes. The dead-end we apparently had encountered lead us to reversing the direction, i.e., instead of starting with the maximum possible of sentences (one for each relation/fact), we decided to look for minimal descriptions, and eventually elaborate them towards a "natural" description.

Usually, humans refine these descriptions by grouping together sentences about related facts, constructing chains of discourse that can be read as a related paragraph. This task is simplified by taking into account additional information such as a *discourse goal* - what goal is to be addressed by the generated text -, a *user model* - what are the particular needs of the user for which the text is intended -, or a *discourse history* - which information has already been conveyed by the system to the user. This information can be used to deterministically plan the required text. Unfortunately, in attempting to provide a reading for a concept map in no specific context, such information is available only in a sketchy form. At best we can say that the discourse goal, taken here as default, is to describe the domain expressed in the concept map in a way that makes it easy to apprehend for a human reader. We are adding a further constraint to the extent that we are trying to obtain the minimal description that will identify to the user the relevant differences between the focused entity and the potential confusor. The conceptual map that describes the domain that has been chosen as referent (Milosavljevic's potential confusor) is in fact acting as a reference for what the user already knows. In this sense, it can be understood as encoding the information about the user that is relevant for the task.

4.1 Text Plannning

The text planning component must provide a plan for the text that is to be generated. In our case, this plan must be a semantic specification of what is to be said to describe the concept blend that has resulted from a process of Conceptual Blending as described above.

The rationale we are trying to bring here is that a "natural" procedure that humans regularly apply for describing concepts is by recurring to the diagnostic (or salient) features, and this leads us to the stated goal of "reading a blend". Once we identify a reference concept (or Milosavljevic's potential confusor) that is close enough to the blend, we are more likely to build a "natural" reading. This reading can be creative in as much as the system is able to find unexpected references. For example, a blend of a horse and a bird can be read as "a horse that has a pair of wings to fly" or as "a plane without an engine and that uses food instead of fuel" (both can carry humans and have a pair of wings and fly), the former taking a horse as reference and the latter taking a plane. Of course, these more "creative" readings demand a space of choices with higher complexity (in the plane example, some features are ignored, some are emphasized). Here, taking a less creative starting point, we propose the use of the input spaces as references for the description, and this will consist on the use of the extra and missing lists as a source for diagnostic features.

Apart from exploring the referred creative aspects, the practical goal of the application of diagnosticity in the NLG readings of the blends is twofold: to provide an intuitive reading of the blend, which, if made by hand in a relatively small concept map, demands considerable effort; and to help in the validation of the system, when confronted with Costello's empirical experiments with people, in which the description follow the same kind of reasoning (e.g."A X Y is a Y with this and that feature of X, without this and that feature of Y...").

In the ideal case, the text planning module would receive as input: the concept maps that describe the domains that have been blended, the blend, and the statistics generated by Divago while running the blend. As explained above, we intend to describe the blend by comparison with one of the original domains. One of the two possibilities must be chosen. We have opted for taking the domain that is most similar to the blend. Alternative options (like focusing on the most different domain) are possible, and they will be explored in detail in further work. However, at the present level of refinement of the NLG modules, they resulted in output that was too verbose to be processed fruitfully, in the sense that it entailed re-describing a high proportion of the most similar concept.

During the blending process Divago carries out various operations that entail comparing the input domains and searching for analogies between them. The results of these operations are present in its output in the form of additional statistics regarding the observed characteristics. These statistics are generated with respect to both input domains. Once we have chosen the most similar one, we concentrate on the statistics that describe the differences between it and the blend: the list of facts in the blend that were not in the original concept map (the extra list), and the list of facts that were in the original concept map and have dissapeared from the blend.

Additionally, in order to produce adequate natural language output, the system must be provided with a name for the concept that has been chosen as reference, and a name for the blended concept. These are used by the system to identify data from each domain when generating the description.

The system operates by structuring the output in terms of a simple schema for comparison. First the comparison is established between the blend and the chosen referent (i.e. "A horsebird is a horse"), then the description is generated by processing the extra and missing lists.

The process of converting those lists into semantic descriptions of the intended content follows a number of steps that progressively refine the input.

The fist step involves transforming the information conveyed by the facts in the extra and missing lists into blocks of information susceptible of being converted at a later stage into linguistic expressions. The representational simplicity of the concept maps used by Divago is oriented toward easing the task of conceptual blending, but it is at times ill-suited for linguistic expression. For instance, a concept map may contain separate facts to represent the following information about a bird: that a bird has wings, that the number of its wings is two, that the purpose of its wings is flying. In fact, the information linking the number or the purpose of wings is only implicitly linked to the bird by the co-occurrence of the corresponding facts in the description of the same domain. At this stage of the processing, such implicit information (that a bird has two wings for flying) is made explicit, and expressed in terms that the rest of the modules of the system will later recognise. This operation represents an elementary kind of aggregation at the conceptual level. The output of this process is already divided into units that will later correspond to linguistic clauses. Because the description is to be phased in terms of a comparison between the two main concepts involved (i.e. a "horse" and a "horsebird"), this process of aggregation is geared towards generating clauses in which the main concept acts as the head of the clause. This operation is akin to the *incremental construction* used for generating sentences from conceptual graphs in Nicolov et al. (1996), in the sense that the facts corresponding to the original concept map are progressively consumed to produce semantic representations that already correspond to sentences. The order of consumption is determined by priorities over the mapping rules, to ensure that those rules resulting in maximum aggregation are applied first.

Once all the facts in each of the lists have been condensed into semantic formulations as concise as possible, a further step of aggregation is applied at the semantic level. For each list, the semantic descriptions are grouped together whenever two of these clauses share a subject or an object, forming a single clause with conjunctive expressions as subjects or objects.

Having obtained the most compact version possible of these lists, a single list is built by appending all the facts in the extra list and the list formed with the negations of all the the facts in the missing list.

The resulting list is then submitted to a further process of semantic aggregation to ensure the most concise solution possible.

The output of the text planning module is converted into a simple rhetorical representation where the resulting list of semantic representations of clauses is included as elaborations on the opening statement ("A horsebird is a horse").

4.2 Sentence Planning and Surface Realization

Little need be said about the sentence planning and surface realization components. Basic state-of-the art processing is applied to eliminate redundant mentions of the same concept during reference, substituting by referring pronouns wherever possible. The size and complexity of the text plans that are being obtained with the current inputs do not require further complication. This may change in the future as more elaborate cases are considered.

The lexicalization module enriches the terms that have been employed so far to include all the additional information that is required by the FUF/SURGE surface realization module. The list of sentence representations received by this module is converted into Functional Descriptions of FUF. Although the conversion that takes place in this module is straightforward look-up on a one-to-one semantic-lexical association table, this does not necessarily imply that there is no *true lexical choice* (Cahill, 1999) involved. For each of the lexical realizations that the system provides in its output, there are usually alternatives that it might have chosen. The fact that choices are limited at this stage is due to the strict criteria applied during content determination, where semantic constructions that might lead to lexicalizations in which the main concepts are not heads of the respective clauses are pruned from the text plan in favour of others. In this sense, the decisions that govern lexical choice are taken in previous stages, and the actual lexicalization of the each sentence plan is reduced to *lexical realization* (Cahill, 1999).

The output of the final module is a sequence of natural language sentences as shown in the examples.

4.3 Results

The procedure described here was applied to thirty conceptual blends generated by Divago from the concept maps for a "horse" and a "bird". In each case, the output was a concept map describing the blend and the set of statistics that describe the relationships between the resulting blend and each one of the original domains.

Not all the resulting blends correspond to concepts that are interesting to describe from our point of view. For some reason, Divago seems to prefer to project facts from the "bird" domain onto the "horse" domain, and the resulting blends are therefore all to be described by comparison with a horse. No single blend resulted where the most similar concept was a bird. This is related to the way in which the integrity constraints and the frames in the generic space interact when running the blend. In some cases the only facts that are projected from the bird domain onto the blend are impossible to relate with the facts about horses in the original domain (e.g. information about nests appears is projected the blend, by no explicit relation between "horsebirds" and nests appears). In such cases, the automatically generated descriptions come out as nonsensical.

In other cases, mismatches between the level of detail of the descriptions between the two domains result in un-

```
(1) A horsebird is a horse.  A horsebird can
fly, it has feathers, a beak, and wings for
flying and it moves by flying.
(2) A horsebird is a horse.  A horsebird has
four eyes, it can fly and it has wings for
flying and moves by flying.  A horsebird does
not have four legs.
(3) A horsebird is a horse.  A horsebird has
hooves for flying, it can fly and it moves by
flying.
(4) A horsebird is a horse.  A horsebird has
two wings and feathers.  It can fly, and it
moves by flying.
(5) A horsebird is a horse.  A horsebird can
fly.  It chirps, it has wings for flying and it
moves by flying.
(6) A horsebird is a horse.  A horsebird has
wings for flying, it can fly, it is smaller
than a human and it moves by flying.
```

Figure 7: Example readings for the *horse bird* blends

interesting projections (e.g. the "horse" domain does not mention that horses have lungs, whereas this information is included for birds; Divago considers attributing lungs to a horse as an interesting blend, since, according to the information in the "bird" domain and the frame in the generic space, this will give a horse the ability to breathe).

The selection of readings presented in figure 7 constitutes a selected sample of the total, in which those readings which were uninteresting for reasons such as the above have been eliminated.

Although these readings are nothing out of the ordinary from a linguistic point of view, they have already proved very useful as simple tools for providing a better understanding of the peculiarities of the blends that Divago is currently producing. The observations about the tendency towards one rather than the other domain, or the differences in level of detail between domains already provide insights that will allow further trimming and refinement of the algorithms.

From a more general point of view, better results will be obtained after several iterations of such refinements. The reason for the need of several iterations is that, in truth, what we have is a natural language generation process in which the text planning process is distributed between the actual operation of blending carried out by Divago and the text planning module. The first step of iteration might simply revise the processes of transcription onto semantic units, of aggregation at the various levels, or of lexicalization. However, better results may be obtained if the refinements are taken a step further, to affect the actual representations employed for each domain (the concept maps) or the set of integrity constraints and frames that are provided in the generic domain.

5 Discussion and Further Work

Contrasting the *blend* with the input spaces is possibly the clearest way to understand its meaning, and thus provide a natural language reading. This observation could be extended to every unknown concept described via a representation similar or equivalent to a concept map. This leads us to the description of concepts by comparison, a field already being explored by others. We are not sure whether the work presented here provides any specific contribution to this field, other than being another practical example of its application. Yet, we would like to bring a short discussion around the creative aspects of the description of concepts by comparison.

The first aspect is related to the choice of the reference. Above, we gave the example of the two choices of reference for describing a "horse-bird" (as the one showed in figure 5): the input domain with shortest extra and missing lists (the "horse") and an apparently "distant" one, the "plane". Although without having implemented the "plane" data, we believe using the "bird" domain as a reference to that example of figure 5 seems to us a less natural or creative description. Namely because a plane and our "horse-bird" both fly, have wings, carry and are driven by humans, they end up being semantically closer than initially expected. The interesting fact now is that the *reading* with the plane as a reference still seems unnatural or, at the least, a non-literal interpretation of the concept map. In other words, and this is a (bold) conjecture demanding further study, choosing as reference the closest input space (the one with smaller missing and extra lists) provides the clearer insight on the blend, while the choice of a third entity (other than the input spaces) may bring alternative readings, enhancing other aspects or triggering metaphoric associations. Proving this hypothesis will demand the application of the reading mechanism described here to a wide range of blends, input spaces and "external" references.

Another aspect that may reveal creativity in such a system lies in the similarity abstraction between the blend and the reference. In the present stage, this similarity is based on *pure* set difference, i.e., we compare the sets of relations from the blend and from the reference. Yet, other levels of similarity, such as those based on structure may provide additional insights. For instance, in the "trashcan basketball" example, it is not correct to say that "trashcan basketball is a basketball game where…", yet these two "games" have some structural similarities that must be explored in the reading (e.g. there is something being shot towards a container; the goal in both is to get as much shots in the container as possible, etc.). Here we see *structure similarity* as happening when we can find a mapping between objects between the domains based on identical relational structure (e.g. having "person shoots crumpled paper" and "player shoots ball", we have a mapping between "crumpled paper" and "ball" and "person" and "player"). The textual interpretation of these cases seems more complex than in set difference comparison and, as in the discussion of reference choice, there may be a multitude of options to choose from. This discussion leads naturally to other less central intentions we have behind this work, which we will explore in further developments: to move from the static reference reading – "X is a Y with...", Y being the reference – to metaphoric reference where a concept is described with another's structure – e.g."In trashcan basketball, the player shoots the crumpled paper to the waste basket in order to score"[5] –, the latter demanding a more subtle reasoning. On this issue we intend to explore possible sinergies with the work of Jones (1994) regarding the generation of transparently-motivated metaphors.

From the point of view of natural language generation, the present work was faced with the obstacle of not having a set of human-generated natural language descriptions of the domains that were to be described. Where such a source is available, knowledge acquisition techniques can be applied either to the corpus of examples or to the experts that produced them, and a wealth of information about how the descriptions are constructed by humans can be obtained in this way (Sripada et al., 2001). The kind of information that may be obtained in this way includes text-level resources such as discourse-plans (Dale et al., 1998) which help to provide a pre-designed structure along which to organise the content. In the present case, no such corpus was available, so the text planning process is currently a freely creative process, with ongoing work progressively elaborating useful heuristics to guide it along the vast search space of possible organizations of the descriptions that can be extracted from the facts.

The question of how to evaluate the performance of natural language generation applications is still an open problem. For a related text planning problem, Sripada et al. (2001) propose an evaluation set up where the results are evaluated at the content level as well as at the text level. This involves having a conceptual representation of the texts generated, and having humans construct equivalent representations based on the same input data. Comparison of human and machine generated conceptual representations (as well as human evaluation of the output texts) would allow to calibrate the system both in terms of the quality of text generated and the appropriateness of the organization chosen for the content. As we said earlier, we plan to use Costello and Keane's examples, and compare their textual descriptions (e.g. "a parrot robin is a parrot that can talk"; "a rhinoceros horse is a horse with horns") with our readings. This will provide us with an estimate of how "natural" our automatically generated readings are.

From the NLG point of view, when covering also "external" domains (such as the "plane" and many others), we will be able to produce textual descriptions and finally study our conjectures around the creative aspects of this work.

In the Conceptual Blending system, apart from keeping the current developments on the implementations, we have now identified some fundamental work to do, namely towards the extension of the concept maps and using already built knowledge bases (like WordNet) to make wider range tests. Another interesting idea to follow is that of using the frames as references for NLG description. In Divago, a frame is a structure that allows the definition of top-level concepts or patterns. For example, a frame can specify the conditions a concept must satisfy in order to be a "transport means". This leads to the use of several references (from frames, external domains, input spaces) simultaneously to participate in the natural language generation of a blend description.

[5] See Coulson (2000) for more on the"trashcan basketball" example.

Acknowledgements

The authors are grateful to the anonymous reviewers for their helpful comments, which have greatly improved the original draft.

This work was partially supported by grant TIC2002-01961 of the Spanish Comisión Interministerial de Ciencia y Tecnología, and COST initiative 282 of the European Union.

References

J. Bateman, B. Magnini, and G. Fabris. The generalized upper model knowledge base: Organization and use. *Towards very large knowledge bases: knowledge building and knowledge sharing*, 1995.

S. Beale, S. Niremburg, E. Viegas, and L. Wanner. De-constraining text generation. In *Proceedings of the Ninth International worshop on Natural Language Generation, Niagara-on-the-Lake, Ontario*, 1998.

T. Berners-Lee and E. Miller. The semantic web lifts off. *ERCIM News*, (51), 2002.

L. Cahill. Lexicalization in applied NLG systems. Technical Report ITRI-99-04, Information Technology Research Institute, University of Brighton, 1999.

F. J. Costello and M. T. Keane. Efficient creativity: Constraint-guided conceptual combination. *Cognitive Science*, 24(2):299–349, 2000.

S. Coulson. *Semantic Leaps: Frame-shifting and Conceptual Blending in Meaning Construction*. New York and Cambridge:Cambridge University Press, 2000.

R. Dale, S. J. Green, M. Milosavljevic, C. Paris, C. Verspoor, and S. Williams. The realities of generating natural language from databases. In *Proceedings of the 11th Australian Joint Conference on Artificial Intelligence*, Brisbane, Australia, 1998.

K. DeSmedt, H. Horacek, and M. Zock. Architectures for natural language generation: Problems and perspectives. In G. Adorni y M. Zock, editor, *Trends in natu-*

ral language generation: An artificial intelligence perspective, pages 17–46. Berlin: Springer, 1995.

M. Elhadad. *Using argumentation to control lexical choice: a functional unification-based approach*. PhD thesis, University of Columbia, 1993.

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

M. A. Jones. *Transparently-motivated metaphor generation*. PhD thesis, University of Delaware, 1994.

B. Lavoie, O. Rambow, and E. Reiter. Customizable descriptions of object-oriented models. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 253–256, Washington, DC, 1997.

G. A. Miller. Wordnet: A lexical database for English. *Communications of the ACM*, 38(11), 1995.

M. Milosavljevic. Augmenting the user's knowledge via comparison. In *Proceedings of the 6th International Conference on User Modelling*, Sardinia, Italy., 1997a.

M. Milosavljevic. Content selection in comparison generation. In W. Hoeppner, editor, *6th European Workshop on Natural Language Generation (6th EWNLG)*, pages 72–81, 1997b.

M. Milosavljevic and R. Dale. Strategies for comparison in encyclopdia descriptions. In *Proceedings of the 8th. International Workshop on Natural Language Generation (INLG '96)*, pages 161–170, Herstmonceux, England, 1996.

N. Nicolov, C. Mellish, and G. Ritchie. Approximate generation from non-hierarchical representations. In *Proceedings of the 8th. International Workshop on Natural Language Generation (INLG '96)*, pages 31–40. Herstmonceux, England, 1996.

J. D. Novak and D. B. Gowin. *Learning How To Learn*. Cambridge University Press, 1984.

F. C. Pereira. Modelling divergent production: a multi domain approach. In *European Conference of Artificial Intelligence, ECAI98*. IOSPress, 1998.

F. C. Pereira and A. Cardoso. Knowledge integration with conceptual blending. In *Proceedings of AICS 2001*, 2001.

F. C. Pereira and A. Cardoso. The boat-house visual blending experience. In *Proceedings of the Symposium for Creativity in Arts and Science of AISB 2002*. AISB, 2002a.

F. C. Pereira and A. Cardoso. Conceptual blending and the quest for the holy creative process. In *Proceedings of the Symposium for Creativity in Arts and Science of AISB 2002*. AISB, 2002b.

F. C. Pereira and A. Cardoso. The horse-bird creature generation experiment. In *submitted*, 2003a.

F. C. Pereira and A. Cardoso. Optimality principles for conceptual blending: A first computational approach. In *Proceedings of the Symposium for Creativity in Arts and Science of AISB 2003*. AISB, 2003b.

E. Reiter. Has a consensus NL generation architecture appeared, and is it psychologically plausible? In David McDonald and Marie Meteer, editors, *Proceedings of the 7th. International Workshop on Natural Language generation (INLGW '94)*, pages 163–170, Kennebunkport, Maine, 1994.

S. Shapiro. Generalised augmented transition networks grammars for generation from semantic networks. *Computational Linguistics*, 2(8):12–25, 1982.

R. Simmons and J. Slocum. Generating english discourse from semantic networks. *CACM*, 2(8):12–25, 1972.

M. Smith, R. Garliano, and R. Morgan. Generation in the LOLITA system: An engineering approach. In *Proceedings of the 7th. International Workshop on Natural Language Generation*. 1994.

J. F. Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

S. Sripada, E. Reiter, J. Hunter, and J. Yu. A two-stage model for content determination. In *Proceedings of ENLGW-2001*, pages 3–10, 2001.

S. Svenberg. Representing conceptual and linguistic knowledge for multi-lingual generation in a technical domain. In *Proceedings of the 7th. International Workshop on Natural Language Generation*. 1994.

The Rags Project. The RAGS reference manual. Technical Report ITRI-01-07, Information Technology Research Institute, University of Brighton, 2001.

A. van Rijn. Generating language from conceptual dependency graphs. In T. Nagle, J. Nagle, L. Gerholz, and P. Eklund, editors, *Conceptual Structures: Current Research and Practice*. Ellis Horwood, 1992.

P. Velardi, M. T. Pazienza, and M. De'Giovanetti. Conceptual graphs for the analysis and generation of sentences. *IBM Journal of Research and Development*, 32 (2), 1988.

J Wagner, R. Baud, and J. Scherrer. Using conceptual graphs operations for natural language generation in medicine. In *Proceedings of the 3rd International Conference on Conceptual Structures (ICCS '95)*. Springer, Santa Cruz, 1995.