Enrichment of Automatically Generated Texts using Metaphor

Raquel Hervás[1], Rui P. Costa[2], Hugo Costa[2], Pablo Gervás[1] and Francisco C. Pereira[2]

[1] Instituto de Tecnologías del Conocimiento
Universidad Complutense de Madrid, Spain
raquelhb@fdi.ucm.es,pgervas@sip.ucm.es,
[2] CISUC, Department of Informatics Engineering
University of Coimbra, Portugal
{racosta,hecosta}@student.dei.uc.pt,camara@dei.uc.pt

Abstract. Computer-generated texts are yet far from human-generated ones. Along with the limited use of vocabulary and syntactic structures they present, their lack of creativeness and abstraction is what points them as artificial. The use of metaphors and analogies is one of the creative tools used by humans that is difficult to reproduce in a computer. A human writer would not have difficulties to find conceptual relations between the domain he is writing about and his knowledge about other domains in the world, using this information in the text avoiding possible confusion. However, this task is not trivial for a computer. This paper presents an approach to the use of metaphors for referring to concepts in an automatically generated text. From a given mapping between the concepts of two domains we intend to generate metaphors for some concepts relating them with the target metaphoric domain and insert these metaphorical references in a text. We also study the ambiguity induced by metaphor and how to reduce it.

1 Introduction

The great challenge for Natural Language Generation (NLG) is known to be one of choice rather than ambiguity. Where natural language understanding has to deal with ambiguity between different possible interpretations of an input, NLG has to decide between different possible ways of saying the same thing. In recent years, natural language generation is slowly considering other domains of application where the choice available for formulating a given concept is much wider. Applications such as the generation of poetry [1] or fairy tales [2] present a wider range of decision points during the generation process than medical diagnosis [3] or weather reports [4].

In domains where the generated texts are more narrative than technical, the differences from the point of view of quality and naturalness between human and computer-generated texts are bigger. Not only the linguistic information used by computers is more restricted, but they also lack creativeness and abstraction capabilities. Analogy and metaphor mechanisms are some of the creative tools used by humans that are difficult to reproduce in a computer. While the human mind deals perfectly with abstraction and conceptual relations between different domains, a computer must have all this kind of knowledge stored in the form of information and specific heuristics to deal with it.

In the present paper we address a small part of this problem. From available conceptual information of different domains it is possible to find semantic correspondences between the concepts belonging to them. These similarities can be used as a base to produce metaphorical references for some of the concepts, but they are not enough to generate an intelligible metaphor. Both non-common properties of the concepts and context of the discourse must be taken into account when generating a metaphor. We present solutions for the different tasks involved in the process, and study their results and limitations.

2 Related work

Two lines of research are reviewed to provide the basis for understanding the work presented here: structure alignment as means for identifying analogies between domains, and natural language generation technology.

2.1 Metaphor and Structure Alignment

It is widely accepted that many of the problems of metaphor interpretation can be handled using established analogical models, such as the structure alignment approach [5]. The general idea behind this approach is that Metaphor fundamentally results from an interaction between two domains (the vehicle and the tenor, in Metaphor literature). This interaction can be simplified as an isomorphic alignment (or mapping) between the concept graphs that represent the two domains. Thus, we see here a domain as being a semantic network (nodes are concepts; arcs are relations), and a mapping between two concepts (of two domains) results from the application of rules that rely on graph structure: if two nodes share the same connection to the same node, they form a potential mapping (triangulation rule [6]); if two nodes share the same connection to other two nodes that are forming a mapping, they form a potential mapping (squaring rule [6]). Since the domain mappings must be isomorphic (1-to-1), there may be many possibilities. Previous attempts at exploring metaphor generation [7] have followed a floodfill probabilistic algorithm based on Divago's Mapper as described in [8]. This alignment algorithm is extremely knowledge-dependent. On the other side, given the complexity of the task (graph isomorphism search), domains too large will become unpractical. To overcome this dilemma, the Mapper is designed not to bring the optimal solution. It uses a probabilistic approach at some choice points, thus potentially yielding different results in each run.

A mapping (say, from a concept X to a concept Y) produced by a structure alignment should emphasize some particular correspondence between two concepts, namely that, according to some perspective, the role that one concept has on one domain (say, the concept Y in the domain T) can be projected to its counterpart in the other domain (say, the concept X in Z). This also implies the implicit projection of the surrounding context (e.g. its function, properties) directly related the concept X to the concept Y. For example, when someone says "My surgeon is a butcher", some immediate functions and properties of "surgeon" are projected to "butcher", such as being "clumsy" (as opposed to "delicate") or using a "cleaver" (rather than a "scalpel"). These in turn provide other inferences, thus allowing for more elaborate descriptions, such as "clinical slaughterhouse" for "hospital". These properties of Metaphor become valuable for tasks such as NLG and its potential is clearly large. Algorithms such as those provided by Sapper [6], Mapper [8] or SME [5] help find suitable mappings that are the needed seed for its use. This in itself raises a number of challenges, some of which are being considered in our work.

2.2 Natural Language Generation

The general process of text generation [9] takes place in several stages, during which the conceptual input is progressively refined by adding information that will shape the final text. During the initial stages the concepts and messages that will appear in the final content are decided and these messages are organised into a specific order and structure (*content planning*), and particular ways of describing each concept where it appears in the discourse plan are selected (*referring expression generation*). This results in a version of the discourse plan where the contents, the structure of the discourse, and the level of detail of each concept are already fixed. The *lexicalization* stage that follows decides which specific words and phrases should be chosen to express the domain concepts and relations which appear in the messages. A final stage of *surface realization* assembles all the relevant pieces into linguistically and typographically correct text. These tasks can be grouped into three separate sets: *content planning*, *sentence planning*, involving the second two, and *surface realization*.

The appropriate use of referring expressions to compete with human-generated texts involves a certain difficulty. According to Reiter and Dale [9], a referring expression must communicate enough information to identify univocally the intended referent within the context of the current discourse, but always avoiding unnecessary or redundant modifiers. When looking for a reference for a specific concept in the text, it is possible to decide between using a pronoun, the plain name of the concept, its proper noun (if any), a description using its attributes, a description using its relations with other concepts, etc. The range of choice depends directly on the available knowledge.

Reiter and Dale [10] describe a fast algorithm for generating referring expressions in the context of a natural language generation system. Their algorithm relies on the following set of assumptions about the underlying knowledge base that must be used. Every entity is characterised in terms of a collection of attributes and their values. Every entity has as one of its attributes some type. The knowledge base may organise some attribute values as a subsumption hierarchy. For each object, there must also be some way of determining if the user

- the person for which the system is generating text - knows whether a given attribute-value pair applies to it. This serves to determine whether mention of a particular characteristic will be helpful to the user in identifying the object. To construct a reference to a particular entity, the algorithm takes as input a symbol corresponding to the intended referent and a list of symbols corresponding to other entities in focus based the intended referent, known as the *contrast set*. The algorithm returns a list of attribute-value pairs that correspond to the semantic content of the referring expression to be realised. The algorithm operates by iterating over the list of available attributes, looking for one that is known to the user and rules out the largest number of elements of the contrast set that have not already been ruled out.

2.3 Metaphor Generation

Little research has been devoted to the generation of metaphors and their use in an automatically generated text. Pereira et al. [7] aimed at improving the stylistic quality of the texts generated by the PRINCE system by extending its capabilities to include the use of simple rhetorical figures. PRINCE (*Prototipo Reutilizable Inteligente para Narración de Cuentos con Emociones*) is a natural language generation application designed to build texts for simple fairy tales. The goal of PRINCE is to tell a story received as input as close as possible to the expressive way in which human storytellers would. To achieve this, PRINCE operates on the conceptual representation of the story, determining what is to be told, how it is organised, how it is phrased, and which emotions correspond to each sentence in the final output.A lexical resource and structure mapping algorithms are used as outlined above to enhance the output texts with simple rhetorical tropes such as simile, metaphor, and analogy.

Pereira et al. identified several problems in the results obtained. From the point of view of interpretation by a reader, metaphorical references and analogies could be confusing if the explicit and implicit information on which they are based is not carefully managed. In addition, once they had identified a metaphor for a specific concept all the appearances of this concept were substituted by the metaphorical reference. As a result the texts were overloaded with analogies, and they presented a significant departure from the original meaning that was difficult to understand. From the point of view of text generation, PRINCE also lacked the linguistic tools required for a correct use of the metaphorical structures. The internal representation of references in PRINCE did not allow the system to refer to any concept using sets of attributes or nominal phrases, so the metaphorical references were reduced to replacing the initial concept with the word assigned to the vehicle concept.

3 Generating and Using Metaphors

The process of generating a metaphorical utterance for referring to a specific concept involves several tasks that must be faced separately. Considering that the initial concept belongs to a given domain, the first step is to find another domain where to look for the desired metaphors. Once it is found, and the mapping between the two domains is established, for each concept susceptible of being referred to using a metaphor a set of possible metaphorical references must be generated. This set of metaphorical references must be studied and evaluated in terms of clearness and suitability, so that inappropriate metaphors can be filtered out. Finally, for each occurrence of the concept in a given context within the text it is necessary to decide whether to use one of the metaphors generated to refer to the concept at that stage or not, always avoiding loss of meaning or unnecessary ambiguity.

The application described in this paper relies on the TAP (*Text Arranging Pipeline*) software architecture for the text generation functionality [11]. TAP is a set of interfaces that define generic functionality for a pipeline of tasks oriented towards natural language generation, from an initial conceptual input to surface realization as a string, with intervening stages of content planning and sentence planning. This process is applied to: the input that is to be processed, the intermediate representations used to store the partial results of progressively filtering, grouping and enriching the input into forms closer and closer to natural language in structure and content, and the set of tasks that take place as steps in that process. The particular instance of the TAP architecture employed here involves three basic modules: a *Content Planner*, a *Sentence Planner*, and a *Surface Realizer*. These modules are organised as a typical basic pipeline for text generation, where the information flows sequentially between the modules that deal with the different tasks involved in the process. Two specific modules have to be considered for the generation and use of metaphors. The Content Planner is the module in charge of deciding which conceptual information from the input would be exposed in the text. It is also in this module where the mapping between domains must be performed, as discussed below. The Reference Solver is a submodule of the Sentence Planner where a referring expression is chosen for each occurrence of a concept in the text. It is in this stage where the metaphorical references must be constructed, and where the decision of whether to use them or not is taken.

3.1 Metaphor in the Content Planner: Identifying the Target Domain and the Mapping

The task of selecting target domains and building the mappings has been located within the Content Planner module because this is the part of the generation pipeline that concentrates on handling a purely semantic representation of the data, in the sense that it has not yet been converted into messages susceptible of being relayed in a linguistic form. Therefore, it makes sense to carry out here the mapping operations, which take place over a semantic form of the domains with no reference to their possible linguistic communication. At subsequent stages of the pipeline, the semantic information required will not be available.

The task of identifying an appropriate additional domain as target domain for the metaphor is quite complex. Given that the metaphor is required to contribute to an act of communication, it is reasonable to say that in order to be appropriate as target domain in an metaphor, a domain must be sufficiently well known to the intended readers of the text so as to require no additional explanation. This narrows down the set of possible domains. It also makes the solution to the problem depend on the particular reader for which the text is intended. Since this requires some means of representing the intended reader as part of the process of generation, for the time being we consider the target domain as given. Further work must focus on exploring the role of reader representation on the choice of target domains.

The generation of each new mapping is firstly dependent on the choice of the pair of domains chosen, the source and the target. In the work here described, we intend to explore the linguistic reference of a concept `X` in terms of a target domain `D`. For example, how can we reference the concept “excalibur” in terms of the Star Wars saga? To find an answer to this question, we have to use as source the concept map that describes `X` (for “excalibur”, it might be a concept map expressing the relations: “excalibur is a weapon”, “excalibur is narrow”, etc.) and as target the domain `D` (for Star Wars, it would include relations such as “Han Solo loves Princess Leia” or “Light Saber is a weapon”).

For the current purpose, a Java implementation of the algorithm described in [8] has been developed. In this implementation, known as jMapper, the original algorithm has been slightly modified to improve its efficiency and scalability, although maintaining its general principles. To reduce the search space, the pairs of candidates are ranked in terms of potential similarity. This potential similarity is directly dependent on the number of shared relations that the two concepts have (e.g. dog is more similar to cat than to car, both have legs, breathe, are pets, etc.) and thresholds are established that avoid the exploration of unpromising portions of the search space.

Given the source and the target, the mapping algorithm (jMapper) thus starts looking for initial seeds to start with. This is based on finding pairs of concepts that share the same relation to a third concept (the triangulation rule). The ones that present higher ranks (more shared concepts) will start the process of looking for 1-to-1 correspondences as briefly described above (and in [8]). Several mappings can potentially emerge from this process, but jMapper (unlike previous versions of this algorithm) eliminates most of these during the generation, thus reaching in the end the largest one that could be found from the chosen seed. It is important, however, to notice that this is not an optimal algorithm, as the choice of other seeds could lead to different results. On the other hand, this version is considerably more efficient in terms of computational resources.

3.2 Metaphor as Referring Expression Generation During Sentence Planning

In order for the system to be able to use metaphors as references to concepts occurring in the input, it must first construct suitable metaphors, and then it has to decide for which particular occurrences of the concept in the text a metaphorical reference is suitable.

Constructing Metaphorical References The subtask of constructing metaphorical references must be carried out taking the two inputs domains and the mapping between them. This constitutes an additional task not usually contemplated in a natural language generation pipeline.

Once a mapping between two domains has been established, it is necessary to decide which information is useful to generate a metaphorical reference for a specific concept. For each pair of concepts mapped together, a list of their common features that have produced the correspondence is given. Apart from these features, the target concept may have extra attributes not belonging to the vehicle concept. This extra information must be used in the generated metaphor not only to describe the target concept, but also to distinguish it from the vehicle one (conceptually, if two concepts share all their features they are the same concept).

In the representation of the world we are working with, each appearing concept or referent is described by a set of properties. Some examples in two domains are shown in Table 1.

Star Wars domain	Some Properties
storm_trooper	[warrior,man,person,evil]
light_saber	[hand_held,narrow,long,weapon]
princess_leia	[beautiful,young,royal_personage,independent,brunette,...]

King Arthur domain	Some Properties
knight	[warrior,man,person,medieval]
excalibur	[hand_held,narrow,long,weapon,magical,steely,...]
guinnevere	[beautiful,young,royal_personage,queen,blonde,...]

Table 1. Examples of properties for two domains

A usual reference for a concept would include some of these properties or attributes to describe it. The idea behind the metaphor is to omit some of these attributes from the reference to this concept by replacing the name of the original concept in the reference with the name of a vehicle concept such that these properties are part of the definition of the vehicle concept. Then the reader will understand as properties of the concept the ones explicitly mentioned along with the ones inferred from the metaphor. For example, the concept "lawyer" may have as attributes to be 'cunning' and 'well-turned out', and a possible reference to it will be "the cunning and well-turned out lawyer". If this concept is mapped with the concept "shark", that is known to be also 'cunning', the resulting metaphorical reference would be "the well-turned out shark".

Introducing Metaphor in Text Metaphor references can not be studied as an isolated phenomenon. In many cases, the context provided by a text is necessary to guide the reader through the assumptions that he must follow to grasp the meaning of the metaphor. But in an appropriate context this metaphor would be understood perfectly. Consider the given example of the lawyer and the shark where the metaphorical reference “the well-turned out shark” can be hardly understood if the reader does not know we are talking about “lawyers”. However, in the sentence “It was the well-turned out shark who won the trial” the word “trial” submerses the sentence in the legal domain and the metaphor is easily inferred. Once the metaphors have been generated, they become an additional option among all those available as possibilities of referring to that particular element. These possibilities usually include: a pronoun, its proper name, and the name of the class to which the element belongs.

The Referring Expression Generation module is in charge of the task of carrying out the selection of the correct reference of a concept at each and every one of its occurrences in the text. The references to a concept will usually be different at each occurrence, to ensure that the text is fluent and reads naturally. Part of its task includes not only selecting a particular type of reference but also ensuring that the chosen type of reference is appropriately enriched with properties that the concept satisfies so as to ensure that the reference is unambiguous in the context. Within the TAP instantiation employed here, this is carried out by an implementation of the Reiter and Dale algorithm, conveniently enriched to allow for the use of pronouns and proper names, which were not contemplated in the original algorithm.

The task of introducing the metaphorical references in the text deciding where in the text they will be appropriately understood is not exhaustively addressed in the present paper. It is sufficient to say that the algorithm for generating referring expression is extended to include metaphor as an additional option. The TAP reference solver includes heuristics to establish which additional properties of a concept must be mentioned to ensure unambiguous reference when using the name of the class it belongs. These heuristics are used to work out which properties should accompany a metaphorical reference to ensure it is easy to understand. The experiments described below are focused on using the metaphorical references only when all the properties that gave rise to the mapping have already been mentioned in the previous discourse.

4 Experiments

In order to test the metaphorical capabilities of our system we have resorted to the use of domain data generated in the past for previous research on Metaphor and Analogy, Tony Veale’s Sapper as reported in his PhD thesis [6]. These data have two distinct advantages. On one hand they constitute a set of coherent domain data already tested for the existence of structural analogies. On the other hand, they were generated independently of the current research effort so they are less likely to be biased towards obtaining interesting results with the proposed method.

Out of the complete data set used in Veale’s thesis, two well known domains have been used: King Arthur saga (target domain) and Star Wars (vehicle domain). The former has been chosen to represent simple referents in our generation system, including the most typical relations of the characters and elements of the domain. The latter is the domain from which metaphors are extracted to refer to concepts in the first one. Part of the knowledge originally encoded for these domains has been excluded, namely relation weights, and some specific kinds of concepts (compound narrative relations, e.g. `become_arthur_king`, `conceive_morgana_mordred`). Thus, for the moment, we are focussing on the properties of characters, objects and their first order relations within the story (e.g. `have`, `friend_of`, `teach`, `loves`, etc.).

Some of the associations returned as part of a mapping are solely based on very simple general relations such as *gender* or *isa*. Such analogies are considered to be uninteresting and they are discarded. In this example the obtained mapping is shown in Table 2. For each association the list of relations that have produced the mapping and the strength of the analogy is shown.

Cross domain association	Supporting Relations	Strength
obi_wan_kenobi ↔ *merlin*	[good,powerful,wise,old,magician,person,man]	0.52
storm_trooper ↔ *knight*	[warrior,man,person]	0.66
light_saber ↔ *excalibur*	[hand_held,narrow,long,weapon]	0.73
han_solo ↔ *lancelot*	[skilful,brave,handsome,young,man,person]	0.43
princess_leia ↔ *guinnevere*	[beautiful,young,royal_personage,person,woman]	0.63

Table 2. Resulting mapping between StarWars and King Arthur domains

By following the algorithm explained in Section 3.2, and using the properties and mapping of Tables 1 and 2, the following metaphorical references are produced:

1. "The medieval storm trooper" instead of "knight". In this case both concepts share many properties, but the attribute 'medieval' belonging to "knight" lets us distinguish between the two concepts and facilitates the correct inferences required to understand the metaphor.
2. "The steely light saber" instead of "Excalibur". As in the previous example, the concepts are distinguished by an extra property of Excalibur, while it also facilitates the readiness of the reference. However, in this case the first concept that comes to mind when reading this reference is 'sword' instead of 'Excalibur'. We will address this issue in the discussion.
3. "Blonde Princess Leia" instead of "Guinnevere". Here Guinnevere is supposed to be blonde while Princess Leia is a brunette. However, this metaphorical reference is completely unintelligible. This case is also addressed in the discussion.

5 Discussion

The generation of metaphors is a delicate process as a metaphor will only be understood if it is possible for the reader to find enough commonalities between the metaphorical concept used and the real concept. For example, in the example we have found a mapping between Lancelot and Han Solo: both of them are skilful men, brave and young. However, if we generate the sentence "Han Solo arrived to Camelot", the metaphor will not be understood. This is exactly the case of the third metaphorical reference shown in section 4, concerning Guinnevere and Princess Leia. Also in the second example it seems that the same problem is partially found when the concept inferred from the metaphor is not "Excalibur" but the class to which it belongs: "sword".

The explanation for these results is the mixed use of concepts and instances of concepts. Words like "sword" or "knight" are general classes that are defined by a set of properties. We can refer to them as concepts in a general sense. And all the specific individuals that belong to these classes, such as "Excalibur" or "Lancelot" are in fact instances of these concepts. In the experiments performed both concepts and instances were treated in the same way. However, the results have shown that inferring properties from or to a specific instance of a concept is more difficult than doing the same between general concepts. This fact agrees with the theory of Glucksberg and his colleagues [12] who argued that metaphors are interpreted as category-inclusion assertions of the form *X is a Y*. According to this proposal interpreters infer from a metaphor a category (a) to which the topic concept can plausibly belong, and (b) that the vehicle concept exemplifies. When using an instance in the metaphor (as for example, is some of the cases above when a particular person is mentioned by his/her proper name) none of these assumptions is fulfilled.

6 Conclusions and Future work

In this work we have focused in a quite simple vision of what is a metaphor. A valid metaphor is not only supposed to have many features in common with the initial concept, but also to provide extra information belonging to both concepts and not mentioned explicitly for the initial one. In addition, these extra features must be salient in the metaphorical concept used, or the reader may not identify them as also attributed to the target concept [12]. For example, in the sentence "His lawyer is a shark" the metaphor `lawyer` $\leftrightarrow$ `shark` is providing implicit information about the lawyer: he is vicious and cunning. This view of metaphor will be explored with more detail in the future.

In our present approach towards the generation of metaphorical references we have only used the properties belonging to concepts and instances when referring to them. However, elements in general domains are also related with other ones by different kinds of relations. For example, in the domains used we can find relations such as "Han Solo is in love with Princess Leia" or "Excalibur is stuck in a stone". These relations could not only be used during the generation of the mapping between domains, but also when the metaphorical reference is created as in “the steely light saber stuck in a stone”. This kind of information may make it easier to understand the metaphors. In contrast with the PRINCE fairy tale generator, TAP is capable of realizing into text this kind of linguistic structures, so this issue will be studied in the future.

The requirement of distinguishing between concepts and instances when facing the generation of metaphors suggests that the use of ontologies, where this distinction is explicitly managed, could be a point to study in the future. The characteristic structure of ontologies would not only permit us to differentiate between concepts and instances, it would also allow us to play with their more or less specific properties depending on whether they are properties of a concept or of an instance, or whether they are inherited from more general concepts. with the use of such a taxonomy of properties, new ways of deciding if a metaphor is suitable for a concept may be addressed.

References

1. Manurung, H.: An evolutionary algorithm approach to poetry generation. PhD thesis, School of Informatics, University of Edinburgh (2003)
2. Callaway, C., Lester, J.: Narrative prose generation. In: Proceedings of the 17th IJCAI, Seattle, WA (2001) 1241–1248
3. Cawsey, A., Binsted, K., Jones, R.: Personalised explanations for patient education. In: Proceedings of the 5th European Workshop on Natural Language Generation. (1995) 59–74
4. Goldberg, E., Driedgar, N., Kittredge, R.: Using natural-language processing to produce weather forecasts. IEEE Expert **9** (1994) 45–53
5. Gentner, D.: Structure-mapping: A theoretical framework for analogy. Cognitive Science **7** (1983)
6. Veale, T.: Metaphor, Memory and Meaning: Symbolic and Connectionist Issues in Metaphor Interpretation. PhD Thesis, Dublin City University (1995)
7. Pereira, F., Hervás, R., Gervás, P., Cardoso, A.: A multiagent text generator with simple rhetorical habilities. In: Proceedings of the AAAI-06 Workshop on Computational Aesthetics: AI Approaches to Beauty and Happiness, AAAI Press (2006)
8. Pereira, F.C.: Creativity and AI: A Conceptual Blending Approach. Mouton de Gruyter (2007)
9. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Cambridge University Press (2000)
10. Reiter, E., Dale, R.: A fast algorithm for the generation of referring expressions. In: Proceedings of the 14th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (1992) 232–238
11. Gervas, P.: TAP: a text arranging pipeline. Technical report, Natural Interaction based on Language Group, Universidad Complutense de Madrid, Spain (June, 2007)
12. Glucksberg, S.: Beyond literal meaning: The psychology of allusion. Psychological Science **2** (1991) 146–152