A Text Generation System that Uses Simple Rhetorical Figures*

Raquel Hervás
Pablo Gervás
Natural Interaction based on Language
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, Spain
raquelhb@fdi.ucm.es,pgervas@sip.ucm.es

Francisco C. Pereira
Amilcar Cardoso
Creative Systems Group AILab
Dpto. de Engenharia Informática
Universidade de Coimbra
Pinhal de Marrocos 3030 Coimbra
{camara,amilcar}@dei.uc.pt

Resumen: Este artículo describe el enriquecimiento de la calidad estilística de los textos generados por el sistema PRINCE, una aplicación de generación de lenguaje natural diseñada para construir textos de cuentos fantásticos. Para ello se explota el potencial de un recurso léxico (WordNet) y algoritmos de mapeamiento estructural para enriquecer los textos de salida con figuras retóricas simples como metáfora y analogía. El sistema enriquecido sigue la arquitectura para sistemas multi-agente OAA, con varios agentes cooperando para conseguir los textos finales. Los resultados de la versión enriquecida son presentados y discutidos. Finalmente, se describen algunos problemas y posibles extensiones.
Palabras clave: Generación de Lenguaje Natural, WordNet, sinonimia, analogía

Abstract: This paper describes the improvement of the stylistic quality of the texts generated by the PRINCE system, a natural language generation application designed for building texts for simple fairy tales. This is done by exploiting the potential of a lexical resource - such as WordNet - and structure mapping algorithms to enhance the output texts with simple rhetorical tropes such as metaphor and analogy. The enhanced system follows the OAA multi-agent system architecture, with various agents cooperating to achieve the final texts. The results of the enhanced version are presented and discussed. Finally, some problems and possible extensions are described.
Keywords: Natural Language Generation, WordNet, synonymy, analogy

1 Introduction

The great challenge for natural language generation is known to be one of choice rather than ambiguity. Where a natural language understanding has to deal with ambiguity between different possible interpretations of an input, natural language generation has to decide between different possible ways of saying the same thing. Existing systems for natural language generation tend to focus on the generation of technical texts, where it is easier to identify 'the' correct way of saying something. But 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 (Manurung, 2003) or fairy tales (Callaway y Lester, 2001) present a wider range of decision points during the generation process than medical diagnosis (Cawsey, Binsted, y Jones, 1995) or weather reports (Goldberg, Driedgar, y Kittredge, 1994).

Rhetorical figures - such as simile, metaphor, and analogy - constitute one of the many ways in which human beings enrich the language they use. These figures have had little presence in natural language generation in the past. This is possibly due to the fact they have little use in the kind of technical document that was being addressed. Some efforts have been carried out to include comparisons in natural language generators operating in pedagogical settings (Milosavljevic, 1997a; Milosavljevic, 1997b), where a dynamic hypertext system (Peba-II) for description of animals is presented. There has been a lot of work on metaphor from a cognitive point of view, but - as far as we know

* Partially supported by the Spanish Ministry of Education and Science, Acción Integrada Hispano-Portuguesa (HP2003-0068).

- very little has been done in terms of studying actual realization of metaphorical uses of words in natural language generation.

The research presented in this paper is aimed at improving the stylistic quality of the texts generated by an existing natural language generation system, by extending its capabilities to include the use of simple rhetorical figures. This is done by exploiting the potential of a lexical resource - such as WordNet - and structure mapping algorithms to enhance the output texts with simple rhetorical tropes such as simile, metaphor, and analogy.

2 Applicable Techniques and Resources

The research presented in this paper relies on two concepts that need a little explanation: the WordNet lexical database and analogy as structure alignment.

2.1 WordNet

Most available linguistic resources are not suitable to use in generation directly due to their lack of mapping between concepts and words. WordNet (Miller, 1995) is by far the richest and largest database among all resources that are indexed by concepts. Other relatively large and concept-based resources such as PENMAN ontology (Bateman et al., 1990) usually include only hyponymy relations compared to the rich types of lexical relations presented in WordNet. For this reason, WordNet has been chosen as initial lexical resource for the development of the module presented in this paper.

WordNet is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. The most ambitious feature of WordNet is its attempt to organize lexical information in terms of word meanings, rather than word forms. English nouns, verbs and adjectives are organized into synonyms sets, each of them representing one underlying lexical concept. These synonyms sets - or *synsets* - are linked by semantic relations like synonymy or hyponymy.

Its organization by concepts rather than word forms allows WordNet to be used also like a knowledge source. The hyponymy/hypernymy relation can be considered equivalent to the "isa" one, and the gloss of the concepts contains extra information that in particular cases can be extracted automatically.

A usual problem in text generation is the inherent ambiguity of language. To deal with this problem WordNet provides some help: the *tag count* field for synsets. This field allows us to order, within a synset, which of the nouns is more *usual* in a generic corpus (in this case, the Brown Corpus(Nelson Francis y Kucera, 1967)).

2.2 Analogy, Metaphor and Structure Alignment

Metaphor and analogy are two cognitive mechanisms that have been recognized as underlying the reasoning across different domains[1]. Because of this, they play an indomitable role in creativity, thus calling our attention as a potential resource for the PRINCE project. Although no consensus has been reached in the current literature regarding a clear distinction between metaphor and analogy, it is clear that their mechanics share many commonalities. It is widely accepted in analogy research that many of the problems of metaphor interpretation can be handled using established analogical models, such as the structure alignment approach (Gentner, 1983) [2]. The general idea behind this approach is that Metaphor (and Analogy) fundamentally result 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 (Veale, 1995)); if two nodes share the same connection to other two nodes that are forming a mapping, they form a potential mapping (squaring rule (Veale, 1995)). Since the domain mappings must be isomorphic (1-to-1), there may be

[1] This claim is nowadays widely agreed, as metaphor is seen as a cognitive rather than a linguistic device. For an extensive *figurative versus literalist* analysis, we redirect the reader to (Veale, 1995)

[2] As semminal works in this area, we can name SME (Falkenhainer, Forbus, y Gentner, 1989) and Sapper (Veale, 1995)

many possibilities. Our own approach follows a floodfill probabilistic algorithm (see (Pereira, 2005) for further details).

3 A Rethorical Text Generator: PRINCE

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 be able to tell a story received as input in a way that is as close as possible to the expressive way in which human storytellers present stories. 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.

PRINCE is implemented using the cFROGS architecture (García, Hervás, y Gervás, 2004), a framework-like library of architectural classes intended to facilitate the development of NLG applications. The flow of control information among the modules of PRINCE acts as a simple pipeline, with all the modules in a sequence in such a way that the output of each one is the input for the next. From a given plot plan provided as input to PRINCE, the text generator carries out the tasks of Content Determination, Discourse Planning, Referring Expression Generation, Lexicalization and Surface Realization, each one of them in an independent module.

3.1 Word Sense Disambiguation in PRINCE

The text generator operates on lexical items from the vocabulary that describe particular concepts. We refer to one such lexical item as VBWord. Each VBWord has a grammatical category which also plays a role in the process. For example, for the concept *"dragon"*, VBWord would be the lexical item *dragon*, of grammatical category *noun*. The VBWord will be the seed for searching for the corresponding WordNet synset.

For the current work, the disambiguation algorithm for VBWord is straightforward: find the synsets which contain VBWord; order them by their tag count; select the synset for which the tag count is higher. The intuition is simple: if VBWord is the commonly used word for a specific synset S (more than in any other synset), then its underlying concept is more likely to correspond to VBWord than any other synset. Although the Brown Corpus may not be the best reference for story tales, the results showed it as a good choice for our first approach to this problem, since we avoided the much larger complexity involved in other WSD methods. In further explorations to this issue, we expect to build our own statistics out of story tale corpora.

3.2 Synonyms in PRINCE

In PRINCE the stage of lexical realization is done in a very simple way. Each concept in the tale has an unique associated tag, and for each appearance of a concept the corresponding word is used in the final text. This produces repetitive and poor texts from the point of view of the vocabulary. As a result, the module gives as output the list of lexical items to convey each message.

Using WordNet the lexical realization process has been enriched with lexical choice, where the decision between lexical alternatives that represent the same content is taken. When a word is needed for a concept during the lexical choice, the system looks for synonyms and hypernyms available for this word in WordNet. Only concepts with singular number are treated in this way to avoid the problem of automatically converting a noun from singular to plural or vice versa. If WordNet does not contain the concept searched, the word stored in the vocabulary is used.

The method for choosing between the alternatives provided by WordNet is as follows. In the first appearance of a concept in a paragraph the word from the system vocabulary is used. This is the word that has been chosen by the developer as the most descriptive one for the concept and explicitly written in the vocabulary. In the second appearance of a concept in a paragraph its first hypernym is used. This hypernym is just a generalization of the concept, but the most specific one. In the rest of appearances of the concept synonyms are used, always using all the synonyms in the list before repeating them.

3.3 Analogies in PRINCE

The task of generating texts where analogies are used graciously involves a number of challenges. On one hand, there is the basic task of identifying the additional domain, the struc-

tural mapping that licenses the analogy, and the corresponding concept in the other domain. On the other hand there is the task of inserting the appropriate linguistic structure for the analogy in the original text, including both the task of building its linguistic rendering and selecting the actual location in the original text in which it is to be inserted. 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.

Having established a particular target domain, the next challenge is to identify the relevant mapping. This involves an elementary operation of comparing two given domains to identify valuable mappings, searching for structural analogies between them. For the PRINCE project, we are exploring the structure mappings with one particular realization template in mind: "X is the Y of Z" sentences (Fauconnier y Turner, 2002).

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 is the rationale behind the "X is the Y of Z" expression, where Z is the domain in which X is integrated (from (Fauconnier y Turner, 2002)). For example, "Freud is the father of Psychoanalysis" results from the the mappings `Freud` $\leftrightarrow$ `father` applied to the domains *Psychoanalysis* and *family structure*, respectively. One can find this template present in many more examples (e.g. "Brugges is the Venice of Belgium", "the Lion is the king of the jungle", "the eyes are the mirror of the soul", etc.). Our goal is therefore to apply this template (using a structure alignment algorithm) in order to get potentially creative text realizations. Thus, we always need two domain concept maps, one for the context at hand (i.e. partially describing the story that is being generated), another for the *vehicle* domain (the one from which to draw the *analogical* perspective). This in itself raises challenges (which domains to use? when? how to select a good mapping?) for which we have some ideas that we will summarize in the discussion section.

As an initial motivating example, we tested with a single vehicle: the Greek deities domain, extracted from WordNet. It was obtained by isolating the subgraph representing the Greek deity taxonomy, enriched with a simple (algorithmical) extraction of relations from their glosses (to get knowledge such as "Aphrodite is the goddess of beauty", "Zeus is father of Aphrodite", "Aeolus is the god of wind", etc.). Whenever needed, our algorithm is able to map a part of the story under construction to this vehicle domain (thus providing expressions like "The princess was the Aphrodite of Royalty" or referencing *king* as the "The Zeus of Royalty"). This is on the one side a very powerful mechanism, but on the other a knowledge greedy algorithm, raising strong limitations.

The insertion of an analogy into a given text can be carried out in at least two different ways. One way is to respect the original text in its given form, and simply to build an additional sentence for conveying the analogy and inserting it at a chosen location. A more complex and richer solution is to add the corresponding message - represented in the same conceptual notation used for the original content of the text - to the input provided for the generator, and to let the generator convert the whole to a coherent text. This solution has the advantage of allowing the generator to reformulate the text surrounding the analogy so as to make the final resulting text linguistically and stylistically coherent. For our current purposes, we have opted for inserting a representation of the analogy at a conceptual level, and then generating a text for the resulting content.

3.4 Implementation Issues

In order to allow an open perspective to the future as well as integrate multiple modules coming from different contributors, it became clear that a multi-agent platform would suit our needs. In this way, we can distribute different roles for different agents, each one being responsible for a specific task. This description coincides fairly with the Open Agent Architecture (Cheyer y Martin, 2001). Such architecture is sufficiently open and modular to allow us implement and test the work presented in this paper as well as to make it easy to plug-in further functionalities.

More precisely, we have a WordNet Agent

- implemented in Prolog by Chris Culy (Culy, 2002), essentially a server for all the queries associated with WordNet database -, a candidate reference agent - the RefSet Agent, which gives sets of candidate references for a concept to whoever asks for them -, a proxy agent - the OAA Facilitator agent that deals with requests/communications between different agents -, and two analogy related agents - Mapper and Analogy ones.

The TextGenerator agent deals with the NLG generation process, and it can be considered as a wrapper for the original PRINCE module. It sets off the flow of control information of the whole process. It initializes the Mapper agent with the whole context of the tale to be rendered into text, producing the mapping between the domains involved. From there, it follows the usual pipeline control flow of PRINCE, interacting with the RefSet agent to obtain information about concepts in the tale.

4 Experiments

A set of five formalized fairy tales and a short piece of Star Wars story have been used to test the generation capabilities of PRINCE. Each of them presents different features from the point of view of length, characters involved and available information. Experiments dealing with synonyms/hypernyms - using the fairy tales - and analogies - using Star Wars story - have been performed separately.

4.1 Synonyms and Hypernyms

"The Dragon" is one of the tales rendered into text by the generator module. In its initial version, without lexical enrichment, a piece of it was as follows:

> *A dragon lived in a cave. The dragon was fierce. The dragon kidnapped the three daughters.*
>
> *The three heroes were brave. The cave was dark. The three heroes went to the cave.*

The concepts *"dragon"* and *"cave"* have only one word associated in the vocabulary, provoking in the text a repetitive use of them.

In Table 1 examples of synonyms and hypernyms in WordNet for two concepts are shown. By definition, the synsets for the concepts always contain the corresponding VB-Word itself, and in the case of *"cave"* it is the only synonym.

The same piece of the tale *"The Dragon"* shown above, enriched after the lexical choice explained in Section 3.2 with the synonyms and hypernyms of Table 1, is the following:

> *A dragon lived in a cave. The mythical monster was fierce. The firedrake kidnapped the three daughters.*
>
> *The three heroes were brave. The cave was dark. The three heroes went to the enclosure.*

The text has become less repetitive, using an enriched vocabulary in a more natural way.

4.2 Analogies and Metaphors

In order to test the analogical capabilities of PRINCE we have resorted to the use of domain data generated in the past for previous research on Metaphor and Analogy (Veale, 1995). 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 to test the analogy capabilities of PRINCE: Star Wars and King Arthur saga. The former has been chosen to represent a very simple story to be rendered by our generation system, including the most typical relations of the characters and elements of the domain. The latter is the domain used by the Analogy agent to find analogies with the first one. A conversion to Mapper representation was necessary, and in that process some amount of knowledge (from Tony Veale's Sapper) was left behind, 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 caracters, objects and their first order relations within the story (e.g. have, friend_of, teach, loves, etc.).

The first step of the text generation in PRINCE is to obtain the possible analogies for the tale domain. The whole context of the story is sent to the RefSet agent so it can find out the analogies between the two domains used, and it is enriched by the Analogy agent

Table 1: Synonyms and hypernyms examples

Concept	VBWord	Synonyms	Hypernyms
dragon	dragon	dragon firedrake	mythical_monster monster imaginary_being imagination creativity power knowledge psychological_feature
cave	cave	cave	enclosure entity

using WordNet. For our simple Star Wars story part of the context is the following:

```
attr(obi_wan_kenobi, good)
have(luke_skywalker, light_saber)
teach(obi_wan_kenobi, luke_skywalker)
friend_of(luke_skywalker, han_solo)
loves(han_solo, princess_leia)
member_of(luke_skywalker, jedi_knights)
member_of(obi_wan_kenobi, jedi_knights)
gender(luke_skywalker,male)
gender(princess_leia,female)
...
```

The enriched graph of relations obtained from the initial context is mapped against the King Arthur saga relations. An extract of the domain information is the following:

```
isa(excalibur, weapon).
attr(excalibur, powerful).
have(king_arthur, excalibur).
gender(king_arthur, male).
gender(guinnevere, female).
friend_of(king_arthur, lancelot).
gender(lancelot, male).
loves(lancelot, guinnevere).
gender(merlin, male).
attr(merlin, good).
teach(merlin, king_arthur).
...
```

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 by the generator. In this example the obtained mapping shown in Table 2. For each association we can see the list of relations that have produced the mapping.

When using a concept during the generation process, PRINCE has the possibility of asking the RefSet agent for the analogy information of this concept.

For the time being, the complete set of analogies found is used, and each one is inserted in the original text after the first appearance of the corresponding concept. An extract of the resulting text in the Star Wars domain, using the analogies achieved, is the following:

> *Luke Skywalker was the King Arthur of the Jedi Knights. He had a light saber. The light saber was powerful. The light saber was the Excalibur of Luke Skywalker.*
>
> *Han Solo loved Princess Leia. He was the Lancelot of Luke Skywalker. She was the Guinnevere of Han Solo.*
>
> *Obi Wan Kenobi taught Luke Skywalker. Obi Wan Kenobi was the Merlin of the Jedi Knights.*

5 Discussion

With respect to the classic pipeline structure of a simple natural language generator the introduction of a conceptual analogy would take place at the Content Determination stage.

Several aspects must be taken into account for this task. It may be necessary to provide more strict criteria for filtering the set of associations returned by the analogy agent. The heuristics currently in use for this purpose were designed to select only one analogy for insertion in the text, and they are oriented to ensuring that no analogy is introduced unless it is supported by a minimal number of relations mirrored between the two domains. When several associations are possible, this restriction is not enough to reduce significantly the number of candidates. The target domain must be presumed to be well known to the intended recipients. The introduction of the analogy may involve the

Cross domain association	Supporting Relations
$good \leftrightarrow good$	[attr]
$obi\_wan\_kenobi \leftrightarrow merlin$	[attr,teach,gender]
$luke\_skywalker \leftrightarrow king\_arthur$	[have,friend_of,teach,gender]
$light\_saber \leftrightarrow excalibur$	[attr,have]
$powerful \leftrightarrow hand\_held$	[attr]
$han\_solo \leftrightarrow lancelot$	[loves,friend_of,gender]
$princess\_leia \leftrightarrow guinnevere$	[loves,gender]

Table 2: Resulting mapping between StarWars and King Arthur domains

suppression from the final text of the explicit mention of at least some of the most obvious of the relations that support it. The decision of how many of the supporting relations to suppress must also be based on some kind of model of the knowledge that the reader has. The minimum amount of relations required to make the analogy understandable must be retained.

The problem of deciding where to place the linguistic realization of the analogy with respect to the rest of the text must be addressed at the stage of Discourse Planning. The set of all analogies found as result of a mapping between two structurally analogous domains is itself so rich to constitute almost a parallel text in itself. In such cases, instead of inserting individual sentences describing each of the possible associations, it may be worthwhile to introduce a full subtext describing the view of the original domain that corresponds to the target domain. Or to insert the analogy-related messages as groups of associations rather than as individual messages. This would correspond to considering the set of associations returned in a mapping as an addition to the conceptual content to be converted into text, susceptible of undergoing - as described above - stages of content determination - where some of the less interesting associations may be discarded - and discourse planning - where the remaining associations are regrouped, possible taking into account the target domain relations that may bind together concepts that appear in different associations. For instance, in the example presented above, it is clear that the messages describing King Arthur and Excalibur as analogous to Luke Skywalker and his light saber might be better presented if they are grouped together:

Luke Skywalker had a light saber. The light saber was powerful. He was the King Arthur of the Jedi Knights and the light saber was his Excalibur. ...

However, this sort of arrangement may not be so good if it results in two completely different parallel texts. Some sort of intermediate clustering should take place during discourse planning, where elements bound together by some relation - like Luke and his light saber and/or Arthur and Excalibur - are grouped prior to establishing the analogy. This has happened by coincidence in the example text for Han Solo/Lancelot and Princess Leia/Guinnevere - and also for Merlin/Obi Wan Kenobi- due to the fact that they share a reciprocal relation. The heuristic should be refined to ensure that such effects are the result of explicit decisions rather than chance.

6 Conclusions and Further Work

The results so far have been positive. The quality of texts produced by the enhanced version of the system has improved noticeably. The range of vocabulary in use has expanded significantly with the use of WordNet.

One of the most rewarding moments during the development of the system occurred when the system started producing sentences that indicated a broader command of English than some of the researchers developing it. This is directly due to the broad coverage provided by WordNet. Nevertheless, it is striking when a program that one has developed - which implies that one knows very well its inner workings - manages to produce surprising results. Even more striking is the fact that the program seems to know English better than us.

In general terms, results can be improved using WordNet to perform the lexical choice not only for singular nouns, but also for plural ones. Implementation of a new module

to deal with the morphological derivations would be required. WordNet contains a morphological processor called Morphy (Beckwith, Miller, y Tengi, 1993) that is applied to the search string to generate a form that is present in WordNet, using during the process a set of morphology functions already defined. The use of Morphy could be a solution for the problem stated.

The extension of the PRINCE system to include use of analogy shows acceptable results for instances where analogies are sought for a single concept. The multi-agent architecture has proved to be a good solution for interconnecting the various resources and techniques that are required to solve the problem.

Further work is necessary to explore the extension of the functionality to instances where analogical equivalents are identified for more than one concept. The classic pipeline architecture of simple natural language generators provides a promising organization for the successive processes that would be involved in this task.

The system shows a lot of promise on the sort of simple examples that it has been tested on. We do intend to carry out more serious testing with a wider range of supporting domains, input domains, and contexts of occurrence.

Bibliografía

Bateman, J. A., R. T. Kasper, J. D. Moore, y R. A. Whitney. 1990. A General Organization of Knowledge for Natural Language Processing: the PENMAN upper model.

Beckwith, R., G. A. Miller, y R. Tengi. 1993. Design and Implementation of the WordNet Lexical Database and Searching Software.

Callaway, C. y J. Lester. 2001. Narrative Prose Generation. En *Proceedings of the 17th IJCAI*, páginas 1241–1248, Seattle, WA.

Cawsey, A., K Binsted, y R. Jones. 1995. Personalised explanations for patient education. En *Proceedings of the 5th European Workshop on Natural Language Generatio*, páginas 59–74.

Cheyer, A. y D. Martin. 2001. The Open Agent Architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1):143–148.

Culy, C. 2002. WordNet Agent. http://www.ai.sri.com/oaa/contributions/wordnet.

Falkenhainer, B., K.D. Forbus, y D. Gentner. 1989. The structure mapping engine: Algorithm and examples. *Artificial Intelligence*, 41:1–63.

Fauconnier, G. y M. Turner. 2002. *The Way We Think*. Basic Books.

García, C., R. Hervás, y P. Gervás. 2004. Una arquitectura software para el desarrollo de aplicaciones de generación de lenguaje natural. *Procesamiento de Lenguaje Natural*, 33:111–118.

Gentner, D. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2).

Goldberg, E., N. Driedgar, y R. Kittredge. 1994. Using natural-language processing to produce weather forecasts. *IEEE Expert*, 9:45–53.

Manurung, H.M. 2003. *An evolutionary algorithm approach to poetry generation*. Ph.D. tesis, School of Informatics, University of Edinburgh.

Miller, G.A. 1995. WordNet: a lexical database for English. *Commun. ACM*, 38(11):39–41.

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

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

Nelson Francis, W. y H. Kucera. 1967. *Computing Analysis of Present-day American English*. Brown University Press, Providence, RI.

Pereira, F.C. 2005. *A Computational Model of Creativity*. Ph.D. tesis, University of Coimbra.

Veale, T. 1995. *Metaphor, Memory and Meaning: Symbolic and Connectionist Issues in Metaphor Interpretation*. PhD Thesis, Dublin City University.