A Multiagent Text Generator with Simple Rhetorical Habilities

Francisco C. Pereira[†] **Raquel Hervás**[⋆] **Pablo Gervás**[⋆] **Amilcar Cardoso**[†]

[†] Creative Systems Group AILab - CISUC
Departamento de Engenharia Informática
Universidade de Coimbra
Pinhal de Marrocos 3030 Coimbra
{camara,amilcar}@dei.uc.pt

[⋆] Natural Interaction based on Language
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, Spain
raquelhb@fdi.ucm.es,pgervas@sip.ucm.es

Abstract

The research presented in this paper is aimed at improving 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 comparison, metaphor, and analogy. The enhanced system follows the OAA multiagent 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.

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 & Lester 2001) present a wider range of decision points during the generation process than medical diagnosis (Cawsey, Binsted, & Jones 1995) or weather reports (Goldberg, Driedgar, & 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; 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 - very little has been done in terms studying actual realization of metaphorical uses of words in natural language generation.

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. Emotional content is added in the form of tags, to be realized as synthezised emotional voice (Francisco, Hervás, & Gervás 2005). PRINCE has been used as natural language generation front end of ProtoPropp (Gervás *et al.* 2004), a system for automatic story generation.

The research presented in this paper is aimed at improving the stylistic quality of the texts generated by the PRINCE system, by entending its capabilities to include the use of simpel 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.

PRINCE is implemented using the cFROGS architecture (García, Hervás, & Gervás 2004), a framework-like library of architectural classes intended to facilitate the development of NLG applications. It is designed to provide the necessary infrastructure for the development, minimising the implementation effort by means of schemas and generic architectural structures commonly used in this kind of systems. cFROGS identifies three basic design decisions when designing the architecture of any NLG system: (1) what set of modules or tasks compound the system, (2) how control should flow between them, deciding the way they are con-

Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

nected and how the data are transferred from one to another, and (3) what data structures are used to communicate between the modules.

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 - what to say -, Discourse Planning - in which order to present it -, Referring Expression Generation - how to describe each occurrence of a concept -, Lexicalization - which word to use for each concept - and Surface Realization - putting it all together in a grammatical sentence -, each one of them in an independent module.

Some resources for creative realization

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.

Synonymy and Hypernymy

Two expressions are synonymous if the substitution of one for the other never changes the truth value of a sentence in which the substitution is made. Words in different syntactic categories cannot be synonyms because they are not interchangeable, which makes it necessary to partition WordNet into nouns, verbs, adjectives and adverbs.

Hyponymy/hypernymy is a semantic relation between word meanings. A concept represented by the synset {x,x,...} is said to be a hyponym of the concept represented by the synset {y,y,...} if a native speaker accepts sentences constructed from such frames as *"An x is a (kind of) y"*. Since there is normally a single superordinate, it generates a hierarchical semantic structure, in wich a hyponym is said to be below its hypernym or superordinate. A hyponym inherits all the features of the more generic concept and adds at least one feature that distinguishes it from its superordinate and from other hyponyms of that superordinate. This convention provides the central organizing principle for the nouns in WordNet.

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 convert a noun from singular to plural and the opposite. 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.

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

[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)

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

For the Prince project, we are exploring the structure mappings with two particular realization templates in mind: "X is the Y of Z" sentences (Fauconnier & Turner 2002); "X is as Y as Z" sentences.

X is the Y of Z 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 & Turner 2002)). For example, "Freud is the father of Psychoanalysis" results from the the mappings `Freud` ↔ `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 sumarize in the discussion section.

For now, we are testing 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"). As we will see in the discussion section, this is on the one side a very powerful mechanism, but on the other a knowledge greedy algorithm, raising strong limitations.

X is as Y as Z Although less subtle than Analogy or Metaphor, the Simile is another figure of speech based on cross domain similarity that we believe can be explored computationally in a setup such as that of Prince. Again, we are looking for a cross-domain mapping, but now with less worries regarding structure alignment: we can focus on two individual concepts that share the same distinctive property, thus avoiding the look for surrounding consistency. For example, if *Adonis* is said to be handsome, one can straighforwardly map our *knight* (which is said to be handsome in the Knowledge Base) and generate the sentence "The knight was as handsome as Adonis". Again, this can only be made possible recurring to a rich knowledge base.

Following ideas from Tony Veale (Veale 2006) and using the LightWeight WordNet (or LWWN) module developed by Nuno Seco (Seco 2005), we propose a simple mechanism that queries WordNet (more precisely, the Extended WordNet (Mihalcea & Moldovan) for property set intersections. These queries look both in the taxonomy as well as in the gloss contents. For example, if we ask for something that is "male" and has the word "handsome" in its gloss, we get the noun "Adonis". Naturally, this yields very interesting outcomes (e.g. "The princess was as pretty as a Rosebud", "The king was as stern as a Dutch Uncle"), but often ends in empty sets. We leave some considerations for the discussion section.

Our Agent Society

The design of Prince is widely modular. In the same way, the roles that different parts take correspond to different processes and techniques. 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 properly suit our needs. In this way, we can distribute different roles for different agents, each other being responsible for a special purpose task. This description coincides fairly with the Open Agent Architecture (Cheyer & 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 (for handling those queries to

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

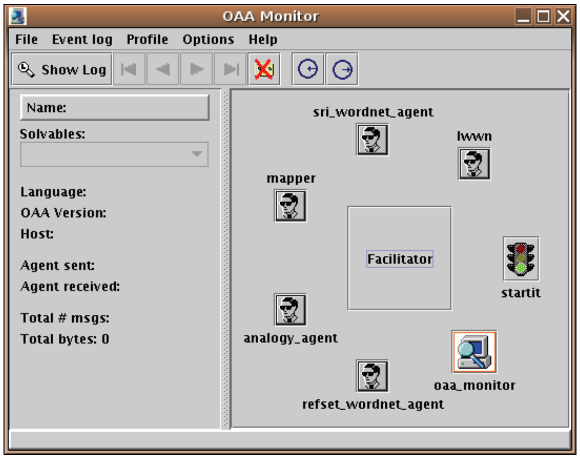

Figure 1: The agent society as seen in the OAA monitor

the database), a candidate reference 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. This agent society is shown in Fig. 1

WordNet agent

This agent was implemented in Prolog by Chris Culy (Culy 2002) and is essentially a server for all the queries associated with WordNet database. For example, to ask for the gloss (the informal description of a concept) of a synset, one can send the following message to this agent:

$$wn\_g(SynsetID, Gloss).$$

where $SynsetID$ is the number identifying the synset, while $Gloss$ will have the reply that this agent will provide.

RefSet agent

We developed this agent (also in Prolog) to provide the lexicalization module a set of candidate alternative lexicalizations. It receives the lexical item from the vocabulary (let us call it VBWord) for a concept, as well as its grammatical category (for example, for the concept *"dragon"*, VBWord would be the noun *dragon*). The VBWord will be the seed for searching for the corresponding WordNet synset (recall that, in WordNet, a concept is not a word, rather it is a set of synonyms). After choosing the correct synset, the RefSet agent gathers the entire synset as well as its hypernym list (ordered by hierarchy - first *father*, then *grandfather*, then *grand-grandfather*, etc.), asks for analogy alternatives from the analogy agents (*analogy_agent* and *lwwn_agent*), and delivers them back to the caller.

WordNet is not organized according to individual *words*, it is organized according to *concepts*. Due to linguistic phenomena such as polysemy and synonymy, there is potentially a many-to-many mapping in which relates to concepts and words. This raises the important problem of Word Sense Disambiguation (Ide & Veroni 1998) (WSD), which has by itself deserved the attention of many researchers. There have been many different approaches to this problem, from a knowledge-dependent perspective to plain statistical word counting of corpora. At this point, 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 & Kucera 1967)). Although this may not be the best corpus 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. Thus, 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.

After selecting the synset to explore, the RefSet agent gathers the set of synonym words (ordered by their individual tag counts) and the set of its hypernyms (ordered by their hierarchical distance). It then creates what we call a *context*: the set of relations from WordNet that are connected to the synonyms and hypernyms detected. In order to find a third list (the analogy and similes list), it sends the words, as well as the context, to the analogy agents. These in turn deliver back (to the RefSet) the analogy proposals. The RefSet agent thus sends the set of those sets to whoever has called it (in this case, the lexicalization module). The message that RefSet agent expects to receive has the following structure:

$$\begin{aligned} refset(\quad & Cat, Word, Context, \\ & [RefSetSyns, RefSetHyps, RefSetAnalogies]) \end{aligned}$$

Currently, the RefSet agent uses services from the WordNet, the Facilitator, the Analogy and the LWWN agents.

Analogy and Mapper agents

The analogy agent is essentially a proxy for the structure alignment agent (Mapper). It can optionally *enrich* the concept map of the VBWord (e.g. looking for relations in WordNet that are not considered in Prince KB), it loads the vehicle concept map and gives control to the Mapper agent. The latter looks for a structure mapping between the two domains and replies it back to the Analogy agents, which fills the slots according to the "X is the Y of Z" template.

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

LWWN agent

The LightWeight WordNet agent is a Java Wrapper for an engine developed by Nuno Seco (Seco 2005). This engine uses Lucene lightweight database server (Gospodnetic & Hatcher 2004) in order to provide a high speed query mechanism. The agent transmits the queries to Lucene and fills the slots according to the "X is as Y as Z" template.

Experiments

A set of five formalized fairy tales has 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 and analogies/comparisons have been performed separately.

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 kidnaped 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 VBWord 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 with the synonyms and hypernyms of Table 1, is the following:

> *A dragon lived in a cave. The mythical monster was fierce. The firedrake kidnaped 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.

Analogies and Metaphors

Tales have been enriched using the mapping capabilities of the agent society in the domain of the Greek deities. Only the concepts of the tale with more information associated obtain complete mappings, due to the fact that these concepts are the ones with more probability of matching relations in the domain used. An example is illustrated using one of the characters of the tale *"The Lioness"*: the princess.

When trying to obtain more information about the concept "princess" for its use during the text generation, PRINCE queries the RefSet agent using as context the available information about the princess:

```
gender(princess,female)
gender(king,male)
attribute(king,stern)
daughter_of(princess,king)
gender(castle,neuter)
live(king,castle)
attribute(princess, pretty)
```

The RefSet agent asks the Analogy mapper to enrich this context using information from WordNet. An extract of this context enrichment is the following:

```
isa(princess,aristocrat),
member_of(princess,royalty),
isa(female,animal),
isa(king,sovereign),
member_of(king,royalty),
isa(male,animal),
isa(stern,back),
part_of(escutcheon,stern),
part_of(skeg,stern),
part_of(stern,ship),
isa(castle,manse),
part_of(great_hall,castle),
isa(neuter,gender),
...
```

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

```
isa('Zeus', 'Greek_deity').
daughter_of('Aphrodite', 'Zeus').
gender('Aphrodite', female).
son_of('Apollo',' Zeus').
gender('Apollo', male).
son_of('Ares', 'Zeus').
gender('Ares', male).
daughter_of('Hebe', 'Zeus').
...
gender('Aeolus',gender,male).
rel('Aether',son_of,'Erebus').
gender('Aether',gender,male).
god_of('Apollo',god_of,light).
brother_of('Apollo', 'Artemis').
daughter_of('Aphrodite', 'Zeus').
gender('Aphrodite', female).
god_of('Ares', war).
...
```

In this example the obtained mapping is:

$female \leftrightarrow female$

$princess \leftrightarrow Aphrodite$

$aristocrat \leftrightarrow Greek\_deity$

$king \leftrightarrow Zeus$

$male \leftrightarrow male$

$sovereign \leftrightarrow fish\_genus$

$royalty \leftrightarrow Zeidae$

The reply of the RefSet agent is Aphrodite/royalty, meaning that the princess is the Aphrodite of the Royalty. The key of the mapping is that both the princess and Aphrodite are female, both are aristocrats, and while the princess is the daughter of the king, Aphrodite is the daughter of Zeus, who has himself been mapped with the king.

The analogies delivered by the RefSet agent has been used in PRINCE in two different ways: as separated sentences or substituting a reference.

When used as a separated sentence in the text, they can be considered as analogies in a strict sense:

> *A princess lived in a castle. She loved a knight. She was the daughter of a king. The princess was the Aphrodite of Royalty.*

When used substituting a reference for a concepts, they can be considered as metaphors:

> *A princess lived in a castle. The Aphrodite of Royalty loved a knight. She was the daughter of a king.*

Comparisons

The LWWN agent has been used to obtain simile information from the gloss of the concepts. For each attribute of each concept, the agent has been used for searching for the hypernyms of the concept one by one, but retaining only the ones that contain the attribute in the gloss.

In the frame of *"The Lioness"* tale, some comparisons obtained are the following:

- The concept "princess" of the tale has as attribute "pretty". After the search using the hypernyms of the concept "princess", the result given by the LWWN agent is that for the hypernym "person" it has found the words "rosebud" and "pin up" having "pretty" in their gloss. So, we get the comparisons "The princess was as pretty as a rosebud" or "The princess was as pretty as a pin up".
- The concept "knight" of the tale has as attribute "handsome". Following the same process, the LWWN agent finds for the hypernym "person" the word "Adonis" having "handsome" in its gloss. We obtain the comparison "The knight was as handsome as Adonis".
- The concept "king" in the tale has as attribute "stern". The LWWN agent recovers for the hypernym "person" again the word "Dutch uncle" having "stern" in its gloss. We obtain the comparison "the king was as stern as a Dutch uncle".

Discussion

In its current state of development, PRINCE has a panoply of rhetorical figures that can be used to enrich generated text. The main challenge facing the development team is to desgine appropriate heuristics to control where and when the various resources are best introduced into the fabric of an input text. An important problem that needs solving is how the different rhetorical figures interact with one another and with the basic NLG tasks that PRINCE is capable of solving - mainly referring expression generation and aggregation. In this context, the design of the right heuristics becomes a question of redesigning the natural language generation architecture of PRINCE. The flexible nature of the cFROGS framework (García, Hervás, & Gervás 2004) that underlies PRINCE makes it possible to address this problem with a certain ease.

The introduction of synonynms and hypernyms may conflict with basic NLG tasks. If an occurrencer of a concept has been replaced by a pronoun during Referring Expresion Generation, the availability of synonyms at that point becomes irrelevant. This suggests that the introduction of synonyms and hypernyms should either be integrated as part of the task of referring expression generation, or be designed to operate in conjunction with it - the fact that it may involve introducing and eliminating concepts that were not in the original input implies that it may also affect Content Determination.

The task of generating texts where analogies are used graciously involves a number of challenges that need to be tackled in separate modules. On one hand, there is the basic task of identifying the additional domain, the structural 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.

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. The process of identifying the target domain and the mapping would be carried out here. Given that the analogy 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 analogy, a domain must be sufficiently well known to the intended readers of the text so as to require no additional explanation. This binds the solution of the problem to particular contexts of interaction, where information about the reader must be available. A simple solution that would allow progress is to consider as candidate domains only those that have been described by the system in the past. This is based on the assumption that system output has been read consistently by the same reader, and it would involve keeping a record of system output - and the concepts involved each time. A simple approach would be for the system to compute mappings between the current input and the most recent domains for which it has provided output, and to select the domain for which the best mapping is obtained. This may be computationally inefficient.

The alignment algorithm used (based on Divago's Mapper (Pereira 2005)) 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. While this may solve the problem of processing time, it still leaves the major issues uncovered: how to choose the vehicle domain? how to choose a good mapping? A possible solution may be the use of a collection of domains, use heuristics to filter the candidates and find mappings for each one; select the mappings that satisfy some simple constraint (e.g. the largest). We may build that collection with the LWWN agent.

The key to another possible improvement lies with the multiagent architecture. The OAA architecture (Cheyer & Martin 2001) underlying the implementation described in the paper allows each agent to run on a separate computer, handling any communications between agents with no additional effort. This would allow a configuration where several mappers agents - one for each candidate domain - can be started off in client machines to compute the mappings. The multiagent architecture also provides the means for establishing a strategy for processing the results. Options may range from selecting the domain processed by the mapper agent who replies first, to actually establishing negotiations between different mapper agents to select the best available mapping.

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 introduction of comparisons presents a dilemma closely related to that posed by the introduction of synonyms/hypernyms. On one hand, it has effects that should take place at the Content Determination stage - introduction of additional concepts not mentioned in the original input - , and on the other hand it could be understood simply as a possible enrichment of the way of referring to - or describing -an object, which would palce it in the Referring Expression Generation stage.

With respect to the actual selection of comparisons, the LWWN agent raises specific issues. The fact that it mostly relies on the WordNet informal glosses makes it as unpredictable and unbalaced as WordNet can be in certain aspects. We believe that solutions for this problem may be found by resorting to available common sense ontologies such as Cyc (Lenat 1995), Open Mind Common Sense (OMCS) (Singh 2002) and Thought Treasure (Mueller 1998). However the integration of such resources with the current method is clearly a subject in need of further research.

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, & 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 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. In this sense, our immediate plans are to work on the implementation of a meta-level of operation of PRINCE, which would be in charge of ascertaining when it is acceptable or interesting to use each of the rhetorical figures that it can produce, given the context of appearance of the particular concept to which the figure is applicable.

Acknowledgements

This work has been partially supported by the Spanish Committee of Science & Technology, Acción Integrada Hispano-Portuguesa (HP2003-0068), and by the Spanish Ministerio de Educación y Ciencia, project TIN2005-09382-C02-01.

References

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

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

Callaway, C., and Lester, J. 2001. Narrative prose generation. In *Proceedings of the 17th IJCAI*, 1241–1248.

Cawsey, A.; Binsted, K.; and Jones, R. 1995. Personalised explanations for patient education. In *Proceedings of the 5th European Workshop on Natural Language Generation*, 59–74.

Cheyer, A., and Martin, D. 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.; Forbus, K. D.; and Gentner, D. 1989. The Structure Mapping Engine: Algorithm and Examples. *Artificial Intelligence* 41:1–63.

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

Francisco, V.; Hervás, R.; and Gervás, P. 2005. Análisis y síntesis de expresión emocional en cuentos leídos en voz alta. *Procesamiento del Lenguaje Natural* (35).

García, C.; Hervás, R.; and Gervás, P. 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).

Gervás, P.; Díaz-Agudo, B.; Peinado, F.; and Hervás, R. 2004. Story Plot Generation based on CBR. In Macintosh, A.; Ellis, R.; and Allen, T., eds., *12th Conference on Applications and Innovations in Intelligent Systems*. Cambridge, UK: Springer, WICS series.

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

Gospodnetic, O., and Hatcher, E. 2004. *Lucene in Action*. Manning Publications. GOS o 05:1 1.Ex.

Ide, N., and Veroni, J. 1998. Word Sense Disambiguation: The State of the Art. *Computational Linguistics* (24):1–40.

Lenat, D. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38(11):33–38.

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

Mihalcea, R., and Moldovan, D. eXtended WordNet: progress report. In *Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*.

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. In *Proceedings of the 6th International Conference on User Modelling*.

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

Mueller, E. 1998. Thoughttreasure: A natural language/commonsense platform.

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

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

Seco, N. 2005. LeightWeight WordNet. http://eden.dei.uc.pt/ nseco/lwwn.tar.gz.

Singh, P. 2002. The public acquisition of commonsense knowledge. In *Proceedings of AAAI Spring Symposium*. Palo Alto, CA: AAAI.

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

Veale, T. 2006. A Typology of Lexical Analogy in Wordnet. In *GWC' 2006, the 3rd Global WordNet Conference*.