Referring Expressions and Rhetorical Figures for Entity Distinction and Description in Automatically Generated Discourses

PhD Dissertation

Submitted by
Raquel Hervás Ballesteros
in partial fullfilment of the requirements of the degree of Doctor of Computer Science

under the guidance of
Prof. Dr. D. Pablo Gervás Gómez-Navarro

Department of Software Engineering and Artificial Intelligence
Universidad Complutense de Madrid
Madrid, April 2009

Abstract

The field of human-computer interaction has evolved rapidly in recent years, becoming a key element of any computer system. If a system is capable of communicating with a human being through interactions that result natural and friendly for him or her (voice, images, etc.), the user will be much more perceptive to the transmitted information and will have more trust on the application and its results.

In this regard, a key area within the human-computer interaction field is Natural Language Generation (NLG), a subfield of Artificial Intelligence and Computational Linguistics. The field of Natural Language Generation is responsible for the design and implementation of systems that produce understandable texts in human languages from an initial non-linguistic representation of information. Within this field, one of the problems to be solved in order to generate satisfactory results is to decide how to refer to entities or elements that appear in the text.

The task of Referring Expression Generation deals with this specific problem. The different references to the same element in a text should be replaced by specific ways in which to refer to them or *references*. The process of referring expression generation should take into account two objectives. First, a reference to an element in the discourse should allow the reader or listener to distinguish it from any other element in the context with which it could be confused. In addition, sometimes the references may contain additional information intended to describe the corresponding entities beyond the function of distinguishing.

Of these two functions (distinctive and descriptive), only the former has been widely studied in the literature. Numerous works can be found dealing with the problem of distinguishing references, confronting issues such as minimality of an expression, similarity of a expression with the ones used by human beings, absence of ambiguity in the generated reference, etc.

However, although there is some work related to the generation of natural language descriptions, there are fewer works focused on enhancing a discourse with certain expressions that highlight descriptive information considered important, or on its relationship with the generation of distinguishing references.

This work addresses the complete problem of reference planning in two different ways. Firstly, several solutions and improvements to classical referring expression generation are proposed for references that attempt to distinguish the referents from other entities in context. The problem is addressed from three fronts: how to adjust the level of abstraction employed to name the reference according to the situation, which strategy to use for choosing the attributes that distinguish a concept, and what words or expressions are more appropriate to express a reference in natural language. For each of these points we present solutions based on classical techniques

and methodologies of Artificial Intelligence, such as evolutionary algorithms, case-based reasoning, or ontologies. The results obtained from the different solutions are also evaluated using classical metrics from this field.

Secondly, this work explores the enhancement of a given speech by providing descriptive information using figures of speech based on similarities between domains, such as comparison and analogy. In order to use such figures in a natural language generation system, it is necessary to address issues related to managing sources of knowledge, determining the appropriate figures, and defining an architecture to implement such systems. This work studies these issues and proposes a general framework to generate this kind of references.

The results obtained by the solutions proposed in this work lead to a discussion on the shortcomings of each approach, identifying aspects that could be improved in future work. The relationship between the generation of referring expressions (both distinctive and descriptive) and the complete process of natural language generation is also discussed.

Finally, the conclusions derived from these lines of research are presented, along with the identification of possible lines for future work and areas of application for the solutions and results presented in this work.

This document is a condensed translation from Spanish into English of the PhD dissertation "Expresiones de Referencia y Figuras Retóricas para la Distinción y Descripción de Entidades en Discursos Generados Automáticamente".

Contents

Chapter 1

Introduction

The field of human-computer interaction has evolved rapidly in recent years, becoming a key element of any computer system. If a system is capable of communicating with a human being through interactions that result natural and friendly for him or her (voice, images, etc.), the user will be much more perceptive to the transmitted information and will have more trust on the application and its results.

Natural Language Generation (NLG) is a subfield of Artificial Intelligence and Computational Linguistics that covers the design and construction of systems that produce text in human languages. The great challenge for natural language generation 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, 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 (Portet et al., 2007) or weather reports (Goldberg et al., 1994).

The general process of text generation (Reiter y Dale, 2000) 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 (CONTENT DETERMINATION), these messages are organised into a specific order and structure (DISCOURSE 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.

This work is focused in this field of Natural Language Generation and, more precisely, in the task of Referring Expression Generation and the challenges faced in this task. The following subsections present the motivation behind this work as a response to the difficulties involved in the selection of appropriate references for a text. We also introduce the specific objectives for this work and describe the general structure of the document.

1.1. Referring Expression Generation

The inputs for the Referring Expression Generation task (REG) are discourse plans for the final text, in which the preceding stages of Content Determination (CD) and Discourse Planning (DP) have already identified what information is going to be transmitted in the text and how it is going to be organized. These plans consist of messages that will be later transformed into sintactically correct sentences in the target language.

The symbolic identifiers that represent each entity on each message must be substituted by different kinds of references that allow the hearer or reader to identify the entities univocally. The REG task performs this substitution, deciding for each entity whether to use a pronoun (*he*, *she*, *they*, etc.), a proper noun (*John*, *The Caledonian Express*, etc.), or a nominal phrase (*the train*), which can be complemented with modifiers or relationships (*the Aberdeen train* or *the train on platform 12*). In each case, it will be necessary to take into account semantic information about the entities we want to refer to.

1.1.1. Referents and References

Jurafsky y Martin (2000) define the following terminology for the field of referring expressions, which we will follow in this work.

We denote the linguistic expressions that point out an element or an individual in a text as REFERRING EXPRESSIONS, REFERENCES or MENTIONS. The element or individual is in turn denoted as REFERENT or ENTITY. The other entities present in the context that might be mistaken as the referent are called DISTRACTORS and they form a CONTRAST SET. Two referring expressions that refer to the same entity are called CO-REFERENCES.

There is also terminology for a referring expression that enables the use of another one, just like an expression such as *John* enables a later use of *he*. In this case, it is said that *John* is the ANTECEDENT for *he*. A reference to an entity that has been previously introduced is called ANAPHORA and the referring expression is called ANAPHORIC.

1.1.2. Communicative Goal for References

Using appropriate referring expressions that can be compared to those that appear in human-generated texts represents a significant challenge. A referring expression should communicate enough information to identify the referent univocally within the context of the discourse, avoiding at the same time redundant or unnecessary modifiers.

Most referring expressions in a discourse are used with the single goal of identifying the referents among all the entities surrounding them. These references are said to perform a DISTINCTIVE FUNCTION. The sentence in example 1.1 would represent this case, where the speaker is using a relationship to command the hearer to put the box in the table that is close to the door, as opposed to other existing tables in the context.

(1.1) *Put the box on <u>the table next to the door</u>*

However, sometimes a reference can also be used to introduce additional information that is considered relevant by the speaker or writer, but that is not required to identify the referent within the context. These references are performing a DESCRIPTIVE FUNCTION or INFORMATIVE FUNCTION. In example (1.2), retrieved from the tale "The Princess and the Pea", there is only one character that can be identified as the king, but the reference gives the reader additional information about his age.

(1.2) *Suddenly a knocking was heard at the palace gate, and <u>the old king</u> went to open it*

In most cases, the descriptive function of references is used to emphasize specific properties from an entity that will be relevant later on the discourse. In example (1.3) it is explicitly mentioned that Cinderella's shoes are made of glass, even though no other shoes are mentioned in the tale. In this case, the intention is to remark that the shoes are made of glass, assuming that the reader is aware that this is not a common material to craft shoes.

(1.3) *Her godmother gave her a pair of <u>glass slippers</u>, the prettiest in the whole world*

The DESCRIPTION OF ENTITIES is a key task in the automatic generation of text for all forms of discourse. From dialog systems to narrative texts, any generated text needs references for the elements that appear in the context with different goals, and these references depend on the descriptions that have been previously introduced. Sometimes the referents need to be distinguished from other similar entities, and their characteristics must be exposed in such a way that they will be useful for this purpose. In other cases, the information provided about certain entities will be important for the complete understanding of the discourse.

Cheng (1998) suggests that a referring expression can be divided in two parts according to its communicative goals and how they must be generated. They are the REFERRING PART, with the goal of distinguishing an object among others, and the NON-REFERRING PART, with the goal of contributing additional information about the object. From this perspective, the references do not necessarily have a single goal (distinguishing or describing), but they may serve both purposes at once.

Cheng states that, even though the main function of referring expressions is to distinguish entities, the references may add redundant information that serves a descriptive purpose. In this cases, the reference must be based on the following principles to be correct:

1. The non-referring part of the reference must not lead to potential confusions about the intended referent on the referring part. That is, if the reference can identify univocally the entity, the addition of the non-referring part should not mislead the hearer or reader.

2. The non-referring part must not diminish the readability of the text. That is, the resulting referring expression must not be too difficult to read after adding the non-referring part.

For these reasons, even though it is common to find references with both referring and non-referring parts in human-generated texts, most existing works deal with both functions separately for the sake of simplicity.

From the perspective of how and when to select the information that will be included in the references, the introduction of information with a descriptive function depends on domain-specific criteria, while the introduction of information with a distinctive function depends on the context of the discourse.

1.2. Description and Goals of this Work

As we have outlined in this introduction, the generation of referring expressions is a complex task that must take into account different issues in order to generate appropriate references in different situations. In addition to deciding what information must be transmitted, it is necessary to take into account the different communication goals that may be involved and the potential implications of generating an inappropriate reference.

There are several works in the literature that approach the problem of Referring Expression Generation with a distinctive function and study aspects such as the minimality of an expression, the similarity of the generated references to those used by human beings, the non-ambiguity of the expressions, etc.

There is also some pre-existing work on Natural Language Generation focused on descriptions (Milosavljevic, 1999). However, there are fewer initiatives studying how to enrich the discourse with descriptive expressions that highlight specific information considered important or how these references are related to the generation of references with a distinctive function.

The general objective of this work is the study of Referring Expression Generation not only with the goal of distinguishing in mind, but also introducing techniques that allow the generation of references with the goal of creating richer and more descriptive texts, that try to lead the reader or hearer towards specific aspects. This general objective can thus be divided in two main blocks.

The first part will focus on classic Referring Expression Generation. We will propose new solutions and improvements over the existing approaches. We will also assess their performance using the most common metrics in this field.

The second part is related to the enrichment of the discourse through the addition of descriptive information as stated above. In particular, we will explore the use of rhetorical figures based on similarities between different domains, such as comparisons and analogies. We will also discuss their impact in the complete process of Referring Expression Generation and other stages of the Natural Language Generation pipeline.

These approaches can be divided in the following set of specific objectives:

1. To study the division of the generation of nominal phrase referring expressions in more specific sub-problems that may be treated separately. We will treat the problem of attribute selection as a search problem, the lexicalization as a task highly related to style considerations, and the abstraction level selection as a context-dependant problem. We will propose solutions for each sub-problem, based on different paradigms from the field of Artificial Intelligence.

2. To evaluate the implemented solutions using the most common metrics employed in the field of Referring Expression Generation, trying to assess their appropriateness (in terms of ambiguity and completeness) and their similarity to human-generated references. For this purpose, it will be necessary to use a corpus of referring expressions created by human evaluators.

3. To study the problem of Referring Expression Generation from the point of view of their descriptive function with the purpose of emphasizing specific information about the referents or describing unknown entities. We will employ rhetorical figures based on similarities between domains such as comparison and analogy.

4. To implement a general architecture that allows the integration of these referring expressions enhanced with rhetorical figures into a previously existing Natural Language Generation system. Given the nature of the problem, which requires analyzing varied knowledge bases, the architecture should allow the interaction of heterogeneous modules in a simple and efficient manner.

1.3. Outline of the Document

This document is structured as follows:

Chapter 1: Introduction. This chapter provides a global vision of the challenges faced in Referring Expression Generation. It includes a discussion about the different functions that a referring expression may serve (distinctive and descriptive) and about the challenges that emerge when both functions are used together causing interferences. It also includes an enumeration of the specific objectives for this particular work.

Chapter 2: Summary of Previous Work. This chapter summarizes the most important points of previous work that are required to follow the rest of the document. It includes a brief discussion of the algorithms and resources used as tools afterwards.

Chapter 3: Generation of Basic Referring Expressions. This chapter covers the generation of references with a distinctive function, studying separately the three steps proposed for the process: Abstraction Level Choice, Attribute Selection and Lexicalization. For each step, different solutions are proposed. These solutions are then assessed and compared in the context of different evaluation challenges in which they were presented.

Chapter 4: Descriptive Information using Rhetorical Figures Based on Similarities between Domains. This chapter explores the enrichment of a discourse through the use of comparisons and analogies to highlight relevant information or to aid the hearer or reader in the understanding of previously unknown entities. This study includes the proposal of a general architecture to integrate this process with previously existing generation systems.

Chapter 5: Discussion. This chapter contains a general discussion of the results, identifying the strengths and limitations of each approach, along with specific proposals for the treatment of the shortcomings identified. We also discuss the impact of these tasks and solutions in the complete process of Natural Language Generation, proposing possible approaches for the implementation of global systems that include these solutions.

Chapter 6: Conclusions and Future Work. In this chapter we summarize the main conclusions derived from this work and outline potential lines of future work.

Chapter 2

Summary of Previous Work

This chapter includes a brief summary of the most important previous work related to this thesis. It includes a description of the algorithm for referring expression generation that is the base of some of the algorithms presented (section 2.1). In section 2.2 a corpus of referring expressions is presented, as it has been used as knowledge resource for part of the work presented in this thesis. Section 2.3 is a brief summary of a series of evaluation tasks on referring expression generation where the implemented algorithms have been submitted. Finally, in section 2.4 the kind of rhetorical figures that have been used are presented, including some related algorithms and tools.

2.1. Incremental Algorithm for the Generation of Nominal Phrase References

Reiter y Dale (1992) describe a fast algorithm for generating referring expressions in the context of a natural language generation system. The algorithm they present is based on psycholinguistic evidence and the analysis of transcript data from human dialogue contributions. As such, it provides an acceptable baseline for the basic operations and the performance expected from such an algorithm.

Their algorithm relies on the following set of assumptions about the underlying knowledge base that must be used:

1. Every entity is characterized in terms of a collection of attributes and their values.

2. Every entity has as one of its attributes a type.

3. The knowledge base may organize some attribute values as a subsumption hierarchy.

Additionally, their algorithm relies for making its final decisions on the following information that must be provided to the system.

- Each object represented in the system should have an associated *basic level value*, which corresponds to the concept which is preferred when referring to that object. This is used to provide a departure point from which to start building references to the object: concepts closer to that basic value in the taxonomy will be preferred over concepts further away. Different basic-level classes may be assigned to the same object for different users.
- 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.
- A list of *preferred attributes* must be available, which describes the set of attributes which human beings prefer when describing objects (*type*, *size*, *shape* and *colour* were observed as preferred for the task the authors considered).

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, 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 realized. 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. The information about basic level value is used to give preference to some attribute over another when the other criteria give no clear choice.

In addition, the algorithm selects the most appropriate value for a specific attribute given an initial value for it. The idea is to use a value for that attribute that is subsumed by the initial value, accurately describes the intended referent, rules out as many distractors as possible, and, subject to these constraints, is as close as possible in the taxonomy to the initial value. This avoids the use of the reference *the chihuahua* in a context when there is only one dog and the most suitable reference would be simply *the dog*.

The complete algorithm can be seen in Figure 2.1.

2.2. TUNA Corpus

One of the most important projects in the field of referring expression generation is the TUNA project (van Deemter et al., 2006; Gatt et al.,

```
make-referring-expression(r, C,P)

L ← {}
D ← C
for each member A_i of list P do
   V = find-best-value(A_i, basic-level-value(r, A_i))
   if V ≠ nil ∧ rules-out(⟨A_i, V⟩) ≠ nil
   then L ← L ∪ {⟨A_i, V⟩}
            D ← D − rules-out(⟨A_i, V⟩)
   endif
   if D = {} then
     if ⟨type, X⟩ ∈ L for some X
       then return L
       else return L ∪ {⟨type, basic-level-value(r, type)⟩}
       endif
   endif
next
return failure
```

```
find-best-value(A, initial-value)

if user-knows(r, ⟨A, initial-value⟩) = true
then value ← initial-value
else value ← nil
endif

for v_i ∈ taxonomy-children(initial-value)
   if v_i subsumes value(r, A) ∧
     (new-value ← find-best-value(A, v_i)) ≠ nil ∧
     (value = nil ∨
      |rules-out(⟨A, new-value⟩)| > |rules-out(⟨A, value⟩)|)
   then value ← new-value
   endif
next
return value

rules-out(⟨A, V⟩)

return {x : x ∈ D ∧ user-knows(x, ⟨A, V⟩) = false}
```

Figure 2.1: Incremental Algorithm by Reiter and Dale

2007). It is a research project funded by the UK's Engineering and Physical Sciences Research Council (EPSRC) in collaboration with the University of Aberdeen, the UK Open University and the University of Tilburg. The main objective was to create a corpus of references for assessing algorithms addressing the generation of referring expressions. The project started in October 2003 and ended in February 2007.

Under the TUNA project a corpus of referring expressions was developed for visual entities in the domains of people and furniture. The corpus was obtained during an on-line experiment in which subjects wrote descriptions of various target entities in a domain where there were also six other entities called distractors. An example of the situations presented to evaluators is shown in Figure 2.2. Each referring expression from the corpus is accompanied by the representation of the domain in which it was generated, containing the target referent and the set of distractors, each with their attributes and values.

The TUNA corpus has been used in this work as reference for the evaluation of the implemented solutions. An exhaustive description of the corpus can be found in (Belz y Gatt, 2008).

2.3. Competitive Evaluation for the Generation of Referring Expressions

In recent years the need to produce results that could be evaluated and compared has emerged within the NLG community. The community took in 2007 the decision to hold a pilot competitive evaluation task focusing on the generation of referring expressions. This task was chosen because of the extensive research that has been done on it in recent decades with the resulting consensus on what problems must be solved and their scope.

Since then, three editions of this kind of competitions have been organized on the task of generating referring expressions, but each one being

Figure 2.2: Domain example presented to the TUNA corpus evaluators

directed to different aspects of the task, and performing different types of evaluation. As the basis for the definition, development and evaluation of all these tasks TUNA corpus data has been used.

The First NLG Challenge on Attribute Selection for Generating Referring Expressions (ASGRE'07)[1] only included the attribute selection task. The Generation of Referring Expresions Challenge 2008[2] expanded the scope of the ASGRE Challenge to include both attribute selection and realisation, separately and as an aggregated task. The third GRE Challenge was concieved as a TUNA Progress Test as part of Generation Challenges 2009[3]. It will take place on April 2009 and includes only the generation of complete references as an aggregation of attribute selection and realisation. More information about these challenges can be found in (Belz y Gatt, 2007) and (Gatt et al., 2008).

Different metrics were used to evaluate the participant systems for the attribute selection task of the challenges:

- **Identification**. System outputs were tested to determine whether the set of properties returned uniquely distinguished the intended referent, or not. As an aggregate measure, organizers took the proportion of outputs of the system which successfully identified their intended referents.

[1] http://www.csd.abdn.ac.uk/research/evaluation/
[2] http://www.nltg.brighton.ac.uk/research/reg08/
[3] http://www.nltg.brighton.ac.uk/research/genchal09/

- **Minimality**. A minimal description is defined as the smallest possible set of attributes, out of the available ones, which distinguishes the referent. As an aggregate measure, organizers took the proportion of minimal distinguishing outputs produced by the systems.

- **System-Human Match**. For every data set input there is a corresponding set of attributes derived from human-produced referring expressions for the target referent. Organizers measured the similarity between the two, which were expected to give different results from the metrics above, as humans choose to overspecify and underspecify for a variety of reasons.

  System outputs were compared to human outputs, estimating their degree of match by using the Dice coefficient. Given two sets of attributes, A1 (system) and A2 (human), Dice is calculated as (2 * the intersection of A1 and A2)/(the total number of attributes in A1 and A2). As an aggregate measure, organizers used the mean Dice score obtained by a system over the set of inputs. A measure called MASI (Measuring Agreement on Set-valued Items) was also used for set comparison. It is slightly biased in favour of similarity where one set is a subset of the other (Passonneau, 2006).

In the case of the realization task, different metrics for string comparison were used:

- **String-edit distance**. This is the classic Levenshtein distance measure, used to compare the difference between a peer output and a reference output in the corpus, as the minimal number of insertions, deletions and/or substitutions of words required to transform one string into another. The cost for insertions and deletions was set to 1, that for substitutions to 2. Edit distance is an integer bounded by the length of the longest description in the pair being compared.

- **BLEU-x.** This is an n-gram based string comparison measure, originally proposed by Papineni et al. (2002) for evaluation of Machine Translation systems. It evaluates a system based on the proportion of word n-grams (considering all n-grams of length $x < 4$ is standard) that it shares with several reference translations. BLEU ranges between 0 and 1.

- **NIST.** This is a version of BLEU, which gives more importance to less frequent (hence more informative) n-grams. The range of NIST scores depends on the size of the test set. It is an aggregate measure.

In addition, a small task-evaluation was perfomed over the final references generated by participant systems. The generated references were shown to readers who will be asked to identify the intended referent, given a visual representation of the input domain. The referring expression and the images of referents were shown separately in two steps, so that subjects were first shown the RE and could then bring up the images when they were ready. Reading speed was measured in the first step (as an estimation of ease of comprehension), and identification speed and identification accuracy (referential clarity) in the second step. Organizers also recorded whether or not the reader accurately identifies the target referent using the proportion of correctly identified referents as an aggregate score.

2.4. Rhetorical Figures based on Similarities between Domains

Metaphor and analogy are two cognitive mechanisms that have been recognized as underlying the reasoning across different domains. For an extensive *figurative versus literalist* analysis, (Veale, 1995) can be consulted. 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). As semminal works in this area, we can name SME (Falkenhainer et al., 1989) and Sapper (Veale, 1995).

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.

For this work we are exploring the structure mappings with a particular realization template in mind: *X is the Y of Z* (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. 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 text that is being generated), another for the *vehicle* domain (the one from which to draw the *analogical* perspective). This in itself raises challenges such as which domains to use or how to select a good mapping.

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. 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 a *knight* (which is said to be handsome) and generate the sentence *The knight was as handsome as Adonis*. Again, this can only be made possible recurring to a rich knowledge base.

For the work presented in this paper we have used a set of domains defined by Tony Veale for his system Sapper (Veale, 1995). In Sapper each domain is represented using a graph where concepts are represented by nodes and relations between concepts are represented by arcs connecting those nodes. Arcs are labelled with tags indicating the relation between two nodes (belong to, attribute, control,...). In addition to this tag, each arc/relation is marked with a real number between -1 and 1 representing the intensity of that relation between the two concepts. Negative intensities mean that the relation is the opposite of the one represented in the arc, and the higher the absolute value of the intensity the stronger the relation. For example, an arc of type *attribute* between the concepts *surgeon* and *educated* with intensity 0.85 implies that surgeons are generally very well educated. However, an attribute arc between *butcher* and *precise* with intensity -0.6 symbolizes not only that butchers are usually imprecise, but that they are quite imprecise.

Chapter 3

Generation of Basic Referring Expressions

Nominal phrases (usually in the form noun + adjectives) are one of the most common forms to express a reference to an entity. However, they are also one of the most complex approaches, as the potential range of choices is very broad. In these references, the noun will usually correspond to the type or class of the referent, and the adjectives will correspond to the modifiers applied to their properties. These modifiers may be either attributes or relationships, although in this work we will focus on the generation of referring expressions using the former. While a pronoun only needs to maintain number and gender concordance, a reference in the form of a nominal phrase with distinctive function must face other challenges. There are three important aspects to take into account in order to generate such referring expressions.

First, it must be observed that, in some cases, the type selected as the noun for the reference may not be the most appropriate choice given the context of the discourses. In those cases in which additional information about the entities is available (as a taxonomy or a hierarchy of concepts), it will be possible to identify the level in which the discourse is taking place and the level that should be used to create the reference. It is thus a problem of selecting an appropriate ABSTRACTION LEVEL TO REFER TO A CONCEPT.

Once we have established the type of the intended referent, it will be necessary to decide which set of modifiers applicable to the entity can distinguish it univocally from any other distracting entities. This process is called ATTRIBUTE SELECTION.

Finally, once we have selected the information that will be included in the reference (type + attributes), it is necessary to decide how that information will be expressed in the text. This will require selecting which words or expressions are more suitable in each reference for the type and the attributes. This is mostly the task for the LEXICALIZATION stage which happens later on the generation process, although it is closely tied to the generation of references.

This chapter covers objectives 1 and 2 from section 1.2. In section 3.1 the problem of choosing the level of abstraction for the reference is addressed, and in sections 3.2 and 3.3 various solutions are presented for attribute selection and lexicalization. Some of these solutions are also evaluated as part of a series of competitive tasks in the field (section 3.4).

The contents of this chapter correspond to the following publications: section 3.1 to (Hervás y Gervás, 2008), section 3.2 to parts of (Hervás y Gervás, 2007; Gervás et al., 2008; Hervás y Gervás, 2009), section 3.3 to parts of (Gervás et al., 2008; Hervás y Gervás, 2009), and section 3.4 to results in the mentioned publications and the evaluation in (Belz y Gatt, 2007; Gatt et al., 2008). Some extra information, not always presented in the papers, is also exposed in this chapter.

3.1. Choice of Abstraction Level for the References using Ontologies

Most of the solutions for the generation of references as a nominal phrase are based on the addition of modifiers to the type of the target referent so it can be distinguished from those that surround it. In general, the type is considered as given. But when a knowledge base organized as a taxonomy that contains the relationships between the types of entities is available, it is possible to improve these solutions if we choose the type of the entity depending on the context.

As an example, let us imagine a room with the following elements:

```
Element1 = {<type,sofa>,<size,big>}
Element2 = {<type,doberman>,<size,big>}
Element3 = {<type,chihuahua>,<size,small>}
```

The taxonomy in Figure 3.1 contains all the known information about the world that corresponds to this example.

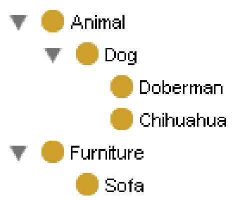

Figure 3.1: Taxonomy corresponding to the example

If we are referring to `Element2` a valid reference could be *the doberman*, because the type is enough to distinguish this element from the other ones. In this case the contrast set could have been only `Element2`, only the two dogs, or maybe the three elements. Now let us imagine that `Element3` disappears from the room. *The doberman* would still be a distinguishing reference for `Element2`, but it would contain additional information that could be irrelevant (the breed of the dog). Using the ontology it is possible to generate a more suitable reference taking into account the type of the other elements in the room. More appropriate references would be *the dog* or even *the animal*. In order to generate this kind of references, the contrast set must not be restricted only to the elements of the same type as the target, but also to elements of related types. Specifically, the contrast set must contain all the elements in the room. Then, the information in the ontology can be used to decide at which level we must name a given referent.

In this work we propose the use of ontologies to deal with the referring expression generation task. We will pay special attention to the choice of the distractors that must be taken into account in the contrast set and to the use of ontology information to select the most appropriate type to be used for the referent. This work has been centered in the generation of definite noun phrases where the type of an element and a set of its properties are given to distinguish it from the other elements in focus. We are also supposing that the situations in which the reference is produced are static, that is, the hearer or reader perception of the world do not change during the process of reference generation.

3.1.1. Composing the Contrast Set

Information about type is generally used to determine which elements of the world must be considered as the contrast set. In referring expression generation algorithms that produce noun phrases formed by the type of the referent and a set of its attributes, the contrast set is made up of the elements of the world that have the same type than the intended referent.

In this work, all the information about the context is located in an ontology. Each instance of the context contained in it has a direct type (the most specific concept it belongs to) and a set of undirect types that are all the types between the direct type and the root of the ontology (that is usually the concept `Thing`).

With the knowledge represented in this way, it is not trivial to decide which type of the referent is appropriate to compose the contrast set. If the type chosen is too specific, the contrast set would be the intended referent and maybe a few more instances with the same direct type. Surely not enough for an appropriate reference generation in a wider context. If the type chosen is too general, the contrast set would be composed by all the instances of the ontology.

Even when it could be seen as a rather general approach, in the work we have developed we discovered that the use of the whole ontology as contrast set is the most suitable option for most of the situations. As we will see later, this choice avoids the use of references more specific than desired while at the same time allows the algorithm to choose the type that is more suitable in a given situation.

3.1.2. An Appropriate Type for the Referent

As many other algorithms for the generation of referring expressions, our approach takes as initial distinguishing attribute the type of the elements appearing in the world. This kind of solution is enough when the types defined for each of the entities of the world are fixed and there is not a close relation for different types. For example, a solution that takes as type the strict one defined in an element would not consider a *doberman* and a *chihuahua* as being both of them *dogs*.

Many reference generation algorithms tackle with this problem providing some kind of *basic level value* for each attribute, considering this value as the more general one known by the hearer or reader. For example, in the Incremental algorithm the values `Doberman` and `Chihuahua` would be subsumed by `Dog`, being this basic level value accessible when needed.

As stated before, in our case all the information about the world is located in an ontology. This ontology contains a taxonomy of concepts that have a common root called `Thing`. The most simplistic solution could be to consider the direct type of each instance as its type, but this would not take advantage of all the information provided by the ontology.

The algorithm we have implemented can be seen in Figure 3.2. Here, r is the intended referent, C is the contrast set, A is the list of attributes that the instances of the ontology hold, *typeValue* is the type that would be assigned to the referent by the algorithm, and L is the list of attribute-value pairs returned if the type is not enough to rule out all the distractors. The `rules-out` function works as the one used in the Incremental algorithm, and the function `incremental-algorithm` calls directly to the original algorithm by Reiter and Dale.

The function `find-best-value-type` is the one that delivers the most appropriate type for the intended referent r taking into account the information in the ontology. We have considered as basic level value for the type the most specific of the common types of the instances of the ontology. From this basic level type, the branch of concepts between it and the direct type of the intended referent r is visited. The type that will be used in the reference is the most general concept from this branch that discards a bigger number of distractors.

```
make-referring-expression ( r, C , A )

L ← {}
C ← instances-ontology()
typeValue ← find-best-value-type( r, C )
C ← C – rules-out( < type , typeValue > )
if C = {} then
    return < typeValue, {} >
else
    L ← incremental-algorithm( r , C , A )
    return < typeValue, L >
endif
```

```
find-best-value-type( r, C )

basic-level-type ← most-specific-common-type( r ∪ C )
value ← basic-level-type
for vi ∈ ontology-children(basic-level-type)
    if | rules-out( < type,vi >) |
        > | rules-out( < type,value >) | then
        value ← vi
    endif
next
return value
```

Figure 3.2: The Algorithm

3.1.3. Attribute Selection for Reference Completion

In some cases the type would not be enough to distinguish two or more referents of the world. This situation is produced when they belong to the same type. In this situation it will be necessary to use their properties to distinguish between them.

The attribute selection carried out in the Incremental algorithm from Reiter and Dale has been applied to these situations. The properties (expressed like attribute-vale pairs) of the instances that are of the same type are extracted from the ontology. The attribute selection is applied over them without any predefined order[1].

3.1.4. Some Examples

An ontology containing information about wines has been used in the work presented (Brachman et al., 1991). The aim was to generate references for different instances of wines which were together in a discourse. We have tested the implemented algorithm over different situations in which a set of wines is presented. For each of them, a distinguishing description is provided using the appropriate type found using the ontology and a set of attributes when they were required.

The instances of the world we have considered are shown in Table 3.1 (the properties of the wines that have not been used by the algorithm are not shown). The references generated for each of the referents are:

1. *The Riesling.* There is another white wine but not a `Dry_Riesling` one, so the most general type that discards more distractors is `Riesling`.

2. *The Chardonnay.* There is another white wine but the type `Chardonnay` is enough to eliminate the other distractors.

[1]As we pretend our solution to be domain-independent, it is not possible to give an order to these attributes

3. *The moderate Cabernet_Sauvignon*. In this case the type is not enough to distinguish this referent, so its attributes have been used. The property that differentiates it from the other `Cabernet_Sauvignon` is the moderate flavor.

4. *The strong Cabernet_Sauvignon*. As in the previous case the strong flavor is used to distinguish the wine from the other `Cabernet_Sauvignon`.

5. *The Rose_Wine*. In this case there is no more rose wines, so this generic type is enough to distinguish the referent.

	Name of the instance	Types	Properties			
			Body	Color	Flavor	Sugar
1	Corban_Dry_White_Riesling	Wine White_wine Riesling				
2	Bancroft_Chardonnay	Dry_Riesling Corbans_Dry_White_Riesling Chardonnay				
3	Marietta_Cabernet_Sauvignon	Bancroft_Chardonnay Rose_wine White_Merlot	Medium	Red	Moderate	Dry
4	Forman_Cabernet_Sauvignon	Forest_Glen_White_Merlot_Rose Red_wine Cabernet_Sauvignon	Medium	Red	Strong	Dry
5	Forest_Glen_White_Merlot_Rose	Marietta_Cabernet_Sauvignon Forman_Cabernet_Sauvignon				

Table 3.1: Examples in the wine ontology

3.1.5. Discussion

The main advantage of this approach is that the algorithm always finds the most suitable value for the type, taking into account the other entities of the world. As we have seen in the examples, if there is only a white wine the system would not provide redundant information about which kind of white wine it belongs to. Since this solution is completely generic and domain-independent, the algorithm would work in the same way with more general ontologies. For example, if the considered ontology contains information not only about wines, but also about other kinds of drinks, the values to be used as types of the referents would also be chosen in the same way. In this sort of situation the referent could be the unique wine in front of other drinks, and the reference generated for it would be the most appropriate one: *the wine*.

In the Incremental algorithm, Reiter and Dale do not address the question of how the contrast set is constructed or how its content can be updated during the generation process. They state that the contrast set is one of the inputs of their algorithm. In our work, we have chosen as contrast set all the instances that can be found in the ontology. This solution allows the algorithm to work with enough information to chose exactly at which level of the ontology the discourse is being displayed (more general or more specific). With this information the generated references are adapted to the level of specificity required in each case.

More details about this algorithm can be found in (Hervás y Gervás, 2008).

3.2. Attribute Selection as a Search Problem

Attribute selection tries to distinguish an entity from its distractors using a description consisting on its type and a set of its features. Let us imagine a room with three objects in it (two chairs and a sofa) with the following features:

```
Object1 = {<type,sofa>,<colour,red>,<size,big>}
Object2 = {<type,chair>,<colour,red>,<size,big>}
Object3 = {<type,chair>,<colour,red>,<size,small>}
```

A valid reference for Object1 would be only its type (*the sofa*), but that would not be enough for Objects 2 and 3. For these two it is necessary to carry out a selection of attributes which distinguishes them from the other objects in the context. In this example, the `size` and `type` attributes would be sufficient (*the big chair* or *the small chair*), while the colour and the type would be not (*the red chair* could be any of the two chairs).

The problem of attribute selection is therefore a complex problem, that can be studied from different perspectives, taking into account different sets of characteristics from the referring expressions.

The problem of attribute selection can be seen as a search problem, as it consists in finding what set of all possible subsets that can be formed with the attributes that describe a given entity is most appropriate when referring to it. Bohnet y Dale (2004) conducted a formal definition of the generation of nominal references nominal as a search problem, and using this definition formalized the most important reference generation algorithm as search algorithms. However, they do not address the question of what problem-solving paradigms based on search can be appropriate for the task of attribute selection.

This section presents three different algorithms for attribute selection as part of the task of referring expression generation. The first two are based on looking for various consideration orders for the attributes in the incremental algorithm. The last one is based on evolutionary algorithms to perform search of the appropriate attributes to use. This paradigm seems appropriate for this problem given that instead of encoding how the solution is found, it is based on how appropriate is a solution.

3.2.1. Attribute Selection based on Relative Groupings

Given the importance of the order of the attributes belonging to the target referent in the selection of attributes according to the algorithm by Reiter and Dale, we decided to consider what types of orders are used by humans when they refer to elements of their environment. The TUNA corpus was used for this purpose. The data was studied separately depending on the domain (furniture vs. people). Our idea was that not only the set of attributes in both domains was very different, but also that the psychological considerations taken into account for a person when referring to a piece of furniture or another person might be significantly different.

In the furniture domain we had to work with a set of six attributes. All the possible combinations of them in different orders gave us $6! = 720$ possibilities to explore. This is not much to be computed automatically, so we generated all the possible order combinations of the attributes and for each of them executed the whole process of generating the attribute selection corresponding to all the examples in the training corpus. The average of the Dice coefficient results was calculated in each case.

The study of these results revealed which combination of the attributes obtained the best results. But is also revealed a peculiarity of the way the quality of the results depended on the order of consideration of the attributes: it seemed to be dependant on the relative order in which certain subgroups of attributes were considered, rather than the order of attributes in general. In other words, the results were almost the same for certain orders of groups of attributes, independently of the internal order inside these groups.

The identified groups in this domain were:

- [*colour*, *type*, *size*]
- [*orientation*, *x-dimension*, *y-dimension*]

This distinction has some kind of psychological plausibility if we consider that one of the groups is more related with the spatial situation of the object, and the other with its own features. It seems possible that different people would feel more comfortable using one or another, depending on their general view of the world.

In the people domain we have to consider 11 attributes. This gave us $11! = 39.916.800$ possible orders for them, too many to be explored exhaustively. Following the intuition about groupings of attributes obtained from the furniture domain, we carried out several experiments creating different combinations of the given attributes. Our first approach was to aggregate the attributes into three sets:

- [*hasShirt*, *hasGlasses*, *hasSuit*, *hasTie*] (clothing related things)

- [*hasBeard*, *hairColour*, *hasHair*, *age*] (appearance related things)
- [*x-dimension*, *y-dimension*, *orientation*] (spatial situation)

each of them containing attributes semantically related. Many combinations of the groups and the elements inside them were tested, but the results obtained with these divisions were not very good.

So, we tried another approach aggregating the attributes into groups depending on the relevance its presence or absence have to distinguish one person from another. For example, to have beard or to wear glasses are usually more perceivable than to wear a tie (especially if the person is also wearing suit). Four new groups were used in the experiments:

- [*hasSuit*, *hasTie*, *hasShirt*]
- [*hasBeard*, *hasGlasses*, *hasHair*, *hairColour*]
- [*age*]
- [*x-dimension*, *y-dimension*, *orientation*]

More details about this algorithm and its evaluation can be found in (Hervás y Gervás, 2007).

3.2.2. Attribute Selection depending on the Most Frequent Value

After the obtained results with the previous algorithm that shown that the order of the attributes is important, we tried a new approach. In contrast to the algorithm by Reiter and Dale, in our algorithm the order in which the attributes must be considered is not the same for all the entities being referred. On the contrary, the order of the attributes for a given referent depends on the values of the attributes. The general idea is that more salient values for a given attribute can provoke its use when giving a description. For example, in the people domain we have observed that the 60 % of the target entities that have beard are referred using the attribute `hasBeard`, but when this attribute has value `0` it is never used. A similar case is the one of `hasHair`. It seems that people find more relevant the fact of not having hair than the fact of having it.

The attribute selection algorithm works as the one by Reiter and Dale, iterating over the list of preferred attributes until an univocal description of the referent is obtained. However, for each entity a different order of the attributes to consider is calculated. This requires some kind of knowledge base containing information on the relevance of certain values of the attributes over others. The ordered list of preferred attributes is calculated by adding first attributes that have more relevant values for the entity being described.

The TUNA corpus was used for determining the preference of the attributes depending on their values. The data was studied to obtain the probability of appearance of each of the attributes when the intended referent has a specific value for it. This probability was calculated using Formula 3.1:

$$prob_{val_i} = \frac{\sum appsValueInAttSet}{\sum appsValueInTarget} \tag{3.1}$$

For each possible value of each of the attributes of the domains, the sum of the appearances of this value in the `ATTRIBUTE-SET` elements (*appsValueInAttSet*) and the sum of the appearances of this value in the attributes of all targets (*appsValueInTarget*) are calculated. The division of these two values is the probability of mentioning an attribute when it has a specific value.

Obtained probabilities can be seen in Table 3.2. For example, the attribute `hasGlasses` is mentioned in the 60 % of the situations when its value is `1`, and in the 0 % of the situations when its value is `0`. On the contrary, the attribute `hasShirt` is almost never mentioned (0.8 % when its value is `1` and 0 % with value `0`). In the furniture domain it seems that values are not so important when determining the order of the attributes, although results show that attributes such as type or colour are used frequently whatever their value is.

More details about this algorithm and its evaluation can be found in (Gervás et al., 2008).

3.2.3. Search of Attributes using Evolutionary Algorithms

We propose the use of evolutionary algorithms (EAs) (Holland, 1992) to deal with the attribute selection task of referring expression generation. Evolutionary algorithms operate over a population of individuals (possible solutions for a problem) that evolve according to selection rules and genetic operators. The fitness function is a metric that evaluates each of the possible solutions, ensuring that the average adaptation of the population increases each generation. Repeating this process hundreds or thousands of times leads to very good solutions for the problem.

This paradigm results adequate for the attribute selection task because it is a process that cannot be formalized, whereas the definition of what is a good reference can be determined easily.

Each individual is represented by a set of genes that are the list of possible attributes in the reference. Each gene has an associated value of 0 (if the attribute is not included in the reference), or 1 (if the attribute is included in the reference). The initial population should have a low number of genes

Furniture			People		
X-DIMENSION	*1*	16.45 %	X-DIMENSION	*1*	35.71 %
	2	18.18 %		*2*	25.92 %
	3	24.59 %		*3*	37.50 %
	4	30.76 %		*4*	15.68 %
	5	18.64 %		*5*	26.00 %
Y-DIMENSION	*1*	33.65 %	Y-DIMENSION	*1*	39.24 %
	2	16.32 %		*2*	23.71 %
	3	29.56 %		*3*	40.25 %
ORIENTATION	*back*	49.01 %	ORIENTATION	*right*	2.43 %
	right	29.47 %		*front*	2.25 %
	front	35.29 %		*left*	1.26 %
	left	25.58 %	TYPE	*person*	83.79 %
SIZE	*small*	43.07 %	hasHair	*0*	15.78 %
	large	32.62 %		*1*	14.41 %
TYPE	*chair*	94.40 %	hasBeard	*0*	0.00 %
	desk	85.88 %		*1*	65.68 %
	fan	97.61 %	hasGlasses	*0*	0.00 %
	sofa	91.48 %		*1*	60.35 %
COLOUR	*blue*	87.20 %	hairColour	*light*	27.43 %
	green	87.50 %		*dark*	26.96 %
	grey	83.33 %	AGE	*young*	0.00 %
	red	86.31 %		*old*	0.00 %
			hasShirt	*0*	0.00 %
				1	0.80 %
			hasSuit	*0*	0.00 %
				1	4.59 %
			hasTie	*0*	0.00 %
				1	2.29 %

Table 3.2: Probabilities of appearance for the values of the attributes in the corpus

set to 1, because references tend to be short and the use of all the possible attributes should be avoided.

Two genetic operators are used: crossover and mutation. For the *crossover operator*, two individuals are selected randomly and crossed by a random point of their structure. Therefore, each of the descendants will have a part of each parent. For the *mutation operator*, some of the genes are chosen randomly to be mutated from 1 to 0, or vice versa.

The key to the evolutionary algorithm lies in the choice of a fitness function. In the case of references, it must find a balance between the univocal identification of a referent, and a natural use of attributes. The formula used as fitness function is defined in Equation 3.2:

$$fit_{ind_i} = f_{att_i} * weight_{att} + ident * weight_{id} \tag{3.2}$$

with $weight_{att} + weight_{id} = 1$.

Parameters $weight_{att}$ and $weight_{id}$ represent the relative weight given to the election of attributes and the identification function of the reference, respectively. The value of *ident* represents whether the reference is univocally identifying the target among the distractors, and f_{att_i} computes the role of attributes depending on the specific situation in which the algorithm is used. For example, if it is decided that the goodness of a references depends equally on its distinctive function and the chosen attributes, both $weight_{att}$ and $weight_{id}$ would have a value of 0,5. On the contrary, if the only thing considered is the distinctive function of the reference, the values $weight_{att} = 0$ and $weight_{id} = 1$ would be used.

In this work we have considered f_{att_i} as the normalised sum of the weight (depending on its absolute frecuency in `ATTRIBUTE-SET` elements in the TUNA corpus) of all attributes present (gene=1), as defined by Equation 3.3:

$$f_{att_i} = \frac{\sum gene_{att_i} * weight_{att_i}}{\#attsRef} \tag{3.3}$$

More details about this algorithm and its evaluation can be found in (Hervás y Gervás, 2009).

3.3. Reference Lexicalization depending on the Style

The Lexicalization task is usually defined as the task in the generation process that selects which words will be used to present a message in the text. In the context of Referring Expression Generation, the Lexicalization chooses an appropriate word to express each noun or adjective. For example, if the intended referent is a man, there are different choices such as (*man*, *guy*, *gentleman*, etc.) depending on factors such as context or style.

The proposal of a Referring Expression Generation algorithm with the capacity to perform lexical choices presents two different challenges. First, we need to have at our disposal enough knowledge to allow more than alternative for the lexicalization of the elements. Then, we need heuristics to decide which alternatives are more appropriate for each reference to an object in a given text. These heuristics take into account aspects such as terminological restrictions or common practices according to different styles or situations. Therefore, the heuristics must be tailored for each particular application domain.

This section presents two different solutions for lexicalization of nominal references. The first one offers a very simple lexical choice over the available data, and the second seeks to model the style or personal preferences using a solution based on case-based reasoning.

3.3.1. Lexicalization Chosing the Most Frequent Option

Although the input data considered for lexicalization is a set of attributes already selected for identifying the target, this is only part of the information required to generate the required expression. Additional decisions must be taken to generate a complete referring expression.

With respect to linguistic variation in the form of expression we have distinguished between choices that give rise to different syntactic structures (which we consider as syntactic choices) and choices which give rise to the same syntactic structures but with different lexical items (which we consider as lexical choices).

For each decision to be taken, the data in the TUNA corpus was studied and the most frequent option was chosen.

Referring Expression Generation

One decision that has been found to be particularly relevant concerns the use of determiners. The examples in the corpus include three possible alternatives: to use indefinite articles, to use definite articles, or to omit the determiners altogether.

Another decision that needs to be taken at this stage is the generation of expressions for describing the spatial location of referents. The corpus shows a wide range of options, many using different systems of reference (north-south vs. top-bottom).

Other decisions that should have been contemplated at this stage to be able to faithfully reproduce the samples in the corpus concern the use of particular features of the object in its description, as in *the desk with the drawers facing the viewer* or the chair with with seat facing away.

Additionally, many descriptions in the corpus rely on relative expressions based on comparison with all or some of the distractors. These can take the form of either adjuncts describing their position relative to other distractors, as in *the blue fan next to the green fan* and *the red chair above the blue one*, or comparative adjectives used for particular attributes, as in *the largest red couch* and *the larger grey chair* (and even combinations of the two as in *the smaller of the two blue fans*).

Finally, there are samples in the corpus of use of ellipsis and ungrammatical expressions that have not been contemplated either. This is important since it implies that there is an upper limit to the possible scores that the system may achieve over the corpus under the circumstances, totally unrelated with the correctness of the generated expressions.

Syntactic Choice

With respect to syntactic choice, some attributes show more than one possible option for syntactic realization. The number of alternatives varies from color (*grey chair - chair that is gray*), through beards (*with beard - with the beard - with whiskers - the bearded man - with a beard - with facial hair*) to orientation (12 different syntactic alternatives for expressing orientation: *back*).

Lexical Choice

There are slight variations of lexical choice over the corpus, as in *sofa - couch - settee - loveseat*, *ventilator - fan - windmill* or *man - guy - bloke* (for nouns) and *large - big* or *small - little* (for adjectives). Because it has a significant impact on the edit distance measure, it is also important to consider the existence of a large number of misspellings in the corpus.

Another important problem is the existence of conceptual mismatches between annotation for the attribute set and the given realization in some cases (*purple - blue*, *black and white - grey*, etc.).

More details about this algorithm and its evaluation can be found in (Gervás et al., 2008).

3.3.2. Case-Based Reasoning for Reproducing Reference Style

We have used the paradigm of case-based reasoning (Agnar y Enric, 1994) to provide a solution that is adapted to the style of a specific set of references. A corpus of references that have been considered appropriate is required in order to extract information for the cases. In our system we used the corpus TUNA for this purpose.

In our approach, a case consists of a description of the problem (`ATTRIBUTE-SET`) and a solution (`ANNOTATED-WORD-STRING`). Cases are stored in a Case Retrieval Net (CRN) (Lenz y Burkhard, 1996), a memory model developed to improve the efficiency of the retrieval tasks of the CBR cycle. Each attribute-value pair from the `ATTRIBUTE-SET` is a node in the net. Templates in `ANNOTATED-WORD-STRING` are considered as solutions to the cases. An example is shown in Figure 3.3.

Similarities between the nodes are established for the retrieval stage of the CBR process depending on the type of attribute. Similarity between equal values for the same attribute is always 1. Similarity between values belonging to different attributes is always 0.

```
<TRIAL CONDITION="-LOC" ID="s60t22">
  <DOMAIN>
    [. . .]
  </DOMAIN>

  <WORD-STRING>middle aged dark haired dark bearded man</WORD-STRING>

  <ATTRIBUTE-SET>
      <ATTRIBUTE ID="a6" NAME="type" VALUE="person"></ATTRIBUTE>
      <ATTRIBUTE ID="a5" NAME="hasBeard" VALUE="1"></ATTRIBUTE>
      <ATTRIBUTE ID="a4" NAME="hairColour" VALUE="dark"></ATTRIBUTE>
      <ATTRIBUTE ID="a3" NAME="hasHair" VALUE="1"></ATTRIBUTE>
      <ATTRIBUTE ID="a2" NAME="hairColour" VALUE="dark"></ATTRIBUTE>
      <ATTRIBUTE ID="a1" NAME="age" VALUE="young"></ATTRIBUTE>
  </ATTRIBUTE-SET>

  <ANNOTATED-WORD-STRING>
    <ATTRIBUTE ID="a1" NAME="age" VALUE="young">
        middle aged
    </ATTRIBUTE>
    <ATTRIBUTE ID="a3" NAME="hasHair" VALUE="1">
      <ATTRIBUTE ID="a2" NAME="hairColour" VALUE="dark">
          dark
      </ATTRIBUTE>
      haired
    </ATTRIBUTE>
    <ATTRIBUTE ID="a5" NAME="hasBeard" VALUE="1">
      <ATTRIBUTE ID="a4" NAME="hairColour" VALUE="dark">
          dark
      </ATTRIBUTE>
      bearded
    </ATTRIBUTE>
    <ATTRIBUTE ID="a6" NAME="type" VALUE="person">man</ATTRIBUTE>
  </ANNOTATED-WORD-STRING>

</TRIAL>
```

Text for the reference

Case Representation

Solution Template

Figure 3.3: Example of case from the corpus

The attribute-value pairs of `ATTRIBUTE-SET` that must be realized in a final string are used to query the net, which returns the more similar cases. Only one of them must be chosen to be adapted for the solution. We consider four different types of retrieved cases: *preferred* (cases with exactly the same attributes than the query), *more* (cases with the same attributes as the query and some more), *lessExtra* (cases that lack some attribute from the query but have some extra ones), and *lessNoExtra* (cases that lack some attribute from the query and have no extra ones). The order given is the preferred order to choose the most suitable case for the query.

Adaptation of the chosen case depends on its type. The idea is to keep all the parts of the template that correspond to attributes common to the query and the case. Extra attributes in the case that do not appear in the query are discarded. Attributes in the query not appearing in the case are lost.

An example is given to show how the algorithm works. Suppose that the algorithm is queried with a query like:

```
TYPE: chair, COLOUR: grey, Y-DIMENSION: 2
```

The reference corresponding to this example in the corpus would be *the grey chair in the middle row*.

The net retrieves a case of the ones considered as *preferred* because it contains exactly the same attributes than the query:

```
TYPE:    COLOUR:  Y-DIMENSION:
chair    grey     3
```

with *the gray chair in the bottom row* as associated reference, and corresponding to the following template:

```
"the"
<ATTRIBUTE name=colour value=grey string="gray"/>
<ATTRIBUTE name=type value=chair string="chair"/>
<META-ATTRIBUTE name=location string="in the bottom row">
   <ATTRIBUTE name=y-dimension value=3/>
</META-ATTRIBUTE>
```

For adapting the retrieved template to the query we must check if the values in the retrieved case coincide with the values in the query. If so, we maintain the lexical tag associated with that value, and otherwise we use a default tag that is assigned to each value. In this example, the values *gray* and *chair* coincide, but not the *y-dimension* one. Instead of using *in the bottom row* we use the default text for value 2: *in row two*. The final reference obtained would be: *the gray chair in row two*.

More details about this algorithm and its evaluation can be found in (Hervás y Gervás, 2009).

3.4. Evaluation of the Presented Algorithms

The presented algorithms for attribute selection and lexicalization were submitted to different editions of the referring generation challenges. Evaluation results for each system can be seen in (Hervás y Gervás, 2007; Gervás et al., 2008; Hervás y Gervás, 2009). Comparisons with other participant systems can be seen in (Belz y Gatt, 2007), (Gatt et al., 2008) and (Gatt et al., 2009).

Chapter 4

Descriptive Information using Rhetorical Figures Based on Similarities between Domains

Rhetorical figures based on similarities between domains (such as analogy, metaphor or comparison) are some of the tools that people use to enrich the language they use.

This kind of figurative language is commonly used to stress and emphasize certain aspects of the entities involved in a discourse. Whether the intention is to augment user knowledge about something, or just to stress the importance of some piece of information in the text, figures like analogy or comparison are commonly used. For example, the analogy *the butcher was clean and precise, a surgeon between butchers* emphasizes how clean and precise the butcher was, something uncommon between butchers, much more than the sentence *the butcher was clean and precise*. In the same way, a comparison like *she was pretty as a rosebud* focuses the attention to how important it is to know that she was pretty, or how important this property is for her, more than the simple use of *she was pretty*.

This chapter explores how a given discourse can be enriched using comparisons and analogies to highlight relevant information or to assist the user when unknown entities are presented, covering goals 3 and 4 of section 1.2. First, we will present a high level proposal about how and when can be appropriate to introduce a comparison or analogy in a discourse (section 4.1). Then, we will present as a case of study how a multiagent architecture can resolve the issues arising when implementing the proposal (sections 4.2 to 4.5), presenting three different algorithms that use two different sources of knowledge.

4.1. High Level Architecture for the Insertion of Comparisons and Analogies in Text

The task of generating texts where analogies are used properly involves a number of challenges that need to be tackled in separate modules. In this work we will assume that the task is addressed from the point of view of enriching a given text represented in some kind of conceptual form that a generator is capable of rendering as text.

The complete process consists of three distinct phases. First, a basic task is to decide in which situations it is appropriate to use one of these figures in the text. In order to take this decision, we should consider what role is expected for the information in the text, and in what points of the discourse it is possible to find structures indicating that a figure of this type can be useful. Then, we should identify a target domain that our analogies or comparisons will refer to, and make the correspondence between the two domains based on their concepts and relations. Finally, we face the task of inserting the linguistic structure for the analogy or comparison in the final discourse, which includes constructing a proper expression for the figure and placing it properly in the text.

In order to undertake this process, it is necessary to resort to some knowledge resource that is sufficiently rich to allow the creation of analogies and comparisons from its data. It must not only include a large amount of data, but also some depth and semantics. If a concept present in the knowledge base is not thoroughly associated with other concepts, it will be difficult to find a conceptual basis on which to build the correspondences that are the foundations of analogies and comparisons.

The choice of why and how to use a comparison or analogy in a discourse is very dependent on whether we are dealing with an analogy or a comparison and will be discussed later. However, the other two issues to be addressed (identifying the target domain and the linguistic realization of the figure) are more general for both approaches and will be discussed in the following subsections.

4.1.1. Identifying the Target Domain and the Mapping

The task of identifying an appropriate additional domain as target domain for the comparison or analogy is quite complex. Given that their purpose is to contribute to an act of communication, enriching it or making it more understandable, the target domain must be sufficiently well known to the intended readers or hearers without further explanations. This narrows down the set of possibilities. It also makes the solution to the problem depend on the particular reader or hearer for whom the text is intended. Since this requires some means of representing the intended reader as part of the process of generation, we will take for granted that the reader is familiar with the target domain. Further work should also take into consideration models of the user's knowledge for the choice of target domains.

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. This comparison will require heuristics to identify acceptable candidates for the analogy or comparison out of the complete set of partial mappings. One possible way to do this is to check how many actual relations determine the association - relations mirrored in both domains which provide the basis for the structural analogy. It would be possible to assume that concept associations based on a higher number of relations are better candidates for analogy or comparison. Other heuristics could be used depending on the goal of the figure.

4.1.2. Realizing the Figure in the Text

Comparisons and analogies can not be studied as an isolated phenomenon. In many cases, the context provided by a text is necessary to guide the reader or hearer through the assumptions that he or she must follow to grasp the meaning of the metaphor. Consider the example *his uncle was a well-turned out shark*, that 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.

The insertion of an analogy or comparison 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 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.

4.2. General Knowledge for Figurative Language

When people use any type of figurative language, a complex inference process is set into motion relying on previously learned knowledge. Due to this knowledge about the surrounding world, and the inferences that can be made over it, a human being is able to understand sentences like *he is a finance shark* or *she is more beautiful than a flower*.

The process carried out to generate or understand this kind of figurative language is complex. Knowledge about the features of the different concepts involved is required, specially about those attributes that are more salient for them. In addition, some kind of ability to find similarities and differences between the concepts is required in order to make correct inferences. For example, in the case of *his uncle was a finance shark*, it is understood that we are talking about a person that is merciless and cunning in his work in the finance world, but it is not assumed that he looks like a shark or that he lives underwater. In the case of *the girl is more beautiful than a flower* we are considering that flowers are usually beautiful, and this property from the girl is being highlighted.

As we commented in section 4.1, when dealing with the automatic generation of this type of figures, some kind of general knowledge is required to carry out inferences and deductions like the ones exposed, at least partially. For this work we have used two different knowledge resources. We use WordNet (Miller, 1995) for comparison generation using its hierarchical structure and the glosses that are defined for the concepts. In addition we will also use the domains from Sapper 2.4 as general knowledge for both comparison and analogy generation.

4.3. A Multiagent Arquitecture for the Creation and Insertion of Rhetorical Figures in Text

In order to explore the feasibility the ideas presented in this work, we have implemented a system that automatically generates comparisons and analogies following the theoretical framework previously exposed. Having identified as requirements an open perspective to the future, the variety of knowledge bases and the need to 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 (Martin et al., 1999). This architecture was sufficiently open and modular to allow us to implement and test the ideas presented in this work and to make it easy to plug-in further functionalities. More precisely, we have a WordNet agent (for handling those queries to 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 one analogy related agent: Mapper (Pereira, 2007).

More information about this multiagent system and the agents that compound it can be found in (Pereira et al., 2006)

4.4. Comparison

The comparisons treated in this work correspond to *illustrative comparisons* as defined by Milosavljevic (2003). An illustrative comparison is a familiarity-based comparison which does not contain any differences between the compared entities. In this type of comparisons a comparator entity is used as a reference in order to help the reader to more easily form a correct conceptual model of a focused entity (or at least of a particular set of attributes of the focused entity).

This work has been focused on the introduction of comparisons in a text for emphasizing properties belonging to discourse entities.

We have considered two situations in which we could desire to highlight some property:

Important Property. When a discourse entity has a property that is very important or significant for its class. The insertion of a comparison in this case not only emphasizes the property from the point of view of the description of the entity, but it also reinforces its membership to its class. For example, a comparison of this kind could be useful when mentioning how pretty a princess is or how brave a knight is, since in the fairy tales domain princesses are usually pretty and knights are usually brave.

Atypical Property. When a discourse entity has a property that is atypical for its class. Atypical properties for a class are not the ones that are not explicitly specified in its definition, but the ones that are opposite of the properties belonging to the class. So, a tall surgeon is not atypical for his class, because to be a surgeon does not have connotations about height. However, a dirty or careless surgeon can be considered highly atypical.

In order to know if a property is significant or atypical for a specific class, Sapper's domains have been used.

4.4.1. Discourse Elements where Comparisons can be Introduced

Once we have decided when comparisons would be used, it is necessary to study in which points of the discourse the information that leads to create a comparison could be found.

As comparisons are ways of highlighting concept properties, the points of the text that must be considered for the creation and insertion of comparisons are the ones where features and attributes of the entities are exposed. These situations can be found in two cases:

1. When a copulative message is being used to give information about the properties of some entity. For example, this would be the case of sentences like *the butcher was precise* or *the princess was pretty.*

2. When the properties are accompanying an entity in an introductory reference. Examples of this situation are *a precise butcher* or *a pretty princess.*

If we are considering that a comparison has as goal to highlight some property belonging to an entity, it must be only used in situations where that entity is being described. A text in which a comparison is inserted anywhere in the discourse just because a property has been mentioned may not make sense.

4.4.2. Reference-Property Pairs as Source for Comparison

When one of the two situations that we have considered as valid for generating a comparison appears, we will have two pieces of information to generate it: the entity that would be compared, and its property that has to be emphasised. The process will be different depending on whether the property belongs to the entity class or not. The available general knowledge would be consulted for this purpose.

4.4.3. Identifying Targets for Comparison

Once it has been decided that a comparison is going to be created taking into account everything exposed above, a concept that can be compared with the entity in the discourse must be found. We have tried two different approaches based on two different knowledge resources: WordNet and the same domains from Sapper we are using in the rest of the process.

Using WordNet Glosses for Comparison Generation

Following ideas from Veale (2006) and using the agent that queries WordNet, we propose a simple mechanism that queries WordNet for property set intersections. These queries look both in the taxonomy as well as in the gloss contents by searching for the hypernyms of the concept one by one, but retaining only the ones that contain the attribute in the gloss. 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.

Using Structure Alignment for Comparison Generation

A mapping from a concept X to a concept Y produced by 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 can be projected to its counterpart in the other domain. If two concepts can be considered analogous or very similar, and one of them has as a remarkable property the one that it is going to be highlighted for the other one, a comparison between both concepts could be useful.

For the current purpose, the agent Mapper has been used. Once a concept and one of its properties are going to be used to create a comparison, Mapper is used to look for possible alignments between this concept and all the concepts in Sappper domains. Each of the obtained concept-concept mappings is checked, and if the target concept found has as an important or remarkable property the one we are looking for, it is selected as target for the comparison. For example, if we are searching for a comparison for a butcher that is atypically clean, the mapping butcher-surgeon arises where the surgeon class is known to be very clean, and the comparison *the butcher is clean as a surgeon* is created. However, even when there are other very clean entities in the domains (operating theater or scalpel), they are not considered because no mapping between them and the butcher class is obtained.

4.4.4. Realising Comparisons

Depending on the situation, the message corresponding to the comparison would be introduced in the text in a different way.

- If we are in the case where the property to be highlighted appears in a copulative message describing the entity, the process is just a substitution of this message for a new one including the comparison.

- If we are in the situation where the property appears modifying the first mention of an entity, the comparison message would be added right after the sentence where this first mention is situated.

4.4.5. Examples of Generated Comparisons

Some small texts are shown where an entity and some of its properties are mentioned. Examples of both important and atypical properties are included, both in positive and negative, to check all the possible situations.

Example 1

We have the following initial text:

There was a surgeon. He was educated. He was influential. The surgeon was careless. He was handsome.

The resulting text after the enrichment using comparisons with WordNet is the following:

There was a surgeon. He was influential. He was educated as a hakham. The surgeon was careless as a scrawler. He was handsome.

And using structure alignment over Sapper domains:

There was a surgeon. He was influential. He was educated as a military general. The surgeon was careless. He was handsome.

In this example we have four candidate properties to be highlighted using a comparison: *educated*, *influential*, *careless* and *handsome*.

To be educated is an important property for a surgeon, whereas to be influential is also a typical property for the class, but not so important. Because of that the sentence *he was educated* is transformed into a comparison. Using WordNet we obtain *he was educated as a hakham*, being hakham *a Hebrew title of respect for a wise and highly educated man.* Using Sapper domains we obtain *he was educated as a military general*, because military general has as attribute *educated* with high intensity. In this case the surgeon holds a property that is considered very representative for its class.

On the contrary, surgeons are supposed to be careful, but this specific surgeon is not. This atypical property is highlighted and instead of *the surgeon was careless* we have *the surgeon was careless as a scrawler*, where scrawler is *a writer whose handwriting is careless and hard to read* in WordNet. However, using structure alignment no comparison has been created, because no concept mapped with surgeon has as important property to be careful.

Finally, the property *handsome* does not appear in the surgeon class, neither positively nor negatively. Therefore it is not highlighted and it is not considered to give rise to a comparison.

Example 2

The initial text is the following:

There was a clean and precise butcher.

The resulting text after the enrichment using comparisons with WordNet is the following:

There was a precise butcher. He was clean as a window cleaner.

And using structure alignment over Sapper domains:

There was a precise butcher. He was clean as a brain surgeon.

We have two candidate properties for comparisons: *clean* and *precise*.

In this case, it is supposed that butchers are neither clear nor precise. The property of not being clean has high intensity value, so from the sentence *he was clean* we obtain *he was clean as a window cleaner* with WordNet, and *he was clean as a brain surgeon* with Sapper domains. The butcher in the example has a property that is considered as highly incompatible with its class, and because of that it must be remarked.

Example 3

The initial text is the following:

There was a corpse. It was unhealthy. It was unpleasant.

The resulting text after the enrichment using comparisons with WordNet is the following:

There was a corpse. It was unhealthy. It was unpleasant as a pill.

Corpses are supposed to be quite unpleasant, and as the one in the discourse is unpleasant as well, the sentence *it was unpleasant as a pill* is generated. However, even when corpses are also unhealthy, the system can not find any comparison for this case using WordNet.

No comparisons have been found using Sapper domains, neither for unhealthy nor unpleasant.

4.5. Analogy

Another rhetorical figure based on similarity between domains that can also be useful to highlight information in a discourse is analogy. An analogy can be used instead of a set of properties if the target of the analogy is something for which these properties are representative, in addition to being known by the listener or reader. This work has found that an analogy may be useful for enriching a given text in one of the following cases:

Entity belongs to an unknown class. When the entity belongs to a class that is unknown to the reader or listener. In this case, instead of giving a detailed list of the properties of the entity in order to explain the reader what we are talking about, an analogy can be used. Thus the reader or listener will get an idea of what the entity is and what things can be inferred on the basis of information already known. Imagine for example that we are talking about the movie Star Wars, and it is unknown to the reader. An analogy like *a storm trooper is like a knight in Star Wars* is giving information about the *storm trooper* as being similar to knights in Middle Age.

In this case the analogy is being used to make unknown things understandable using something that belongs to previously known domains.

Entity with atypical properties. When an entity has some properties that are atypical in its class. As in the case of comparisons, properties are considered atypical for the class when they are opposite to its properties, not just when they are not specified. An example might be a butcher who is clean and accurate, two qualities that are atypical for butchers, but very common among surgeons. A possible analogy would be *he was a surgeon between butchers.*

Unlike the case of comparisons, which were intended to highlight a property of an entity, analogies intend to stress that the entity is strange for its class, and the analogy is giving implicitly why.

To determine whether a class or concept is unknown to the reader or listener it is required to have some kind of user model containing this information. This model should include any prior knowledge that can be considered as known. However, we have not treated this problem in the present work.

4.5.1. Discourse Elements where Analogy can be Introduced

Once he have decided the expected use for analogies, it is necessary to consider in which points of the discourse you can find the required information to decide if an analogy is appropriate, and to create it if it is considered useful. This will depend on whether we are in the case where the analogy is used to present a concept in terms of another, or to highlight an unusual entity with respect to its class.

- *Entity from an unknown class.* If the analogy is intended to give information about an unknown concept, using another that it is known, it must be placed immediately after the presentation of the unknown concept. This presentation may not only refer to the first occurrence of the concept, but depending on the situation could include the entire description that was made for it.

- *Entity with atypical properties.* If the analogy is intended to highlight an atypical entity, the analogy must be included in the point where not only the concept has been described, but where the properties that make it so atypical have also been mentioned. For example, an analogy such as *the butcher was like a surgeon between butchers* can be a nonsense if it has not been commented that this butcher was particularly precise and careful in his job.

Although in the first case it seems that the analogy could be understood without difficulty both before and after the complete description of a concept, we have considered that analogies are always inserted after the entity has been fully described in the speech.

Thus, we need to identify at what point of the discourse we can consider that a description is complete. As in the case of comparisons, we assume that descriptions are expressed using copulative messages like *the man was brave*, so the description of a discourse entity is considered complete after the last copulative message mentioning any of their properties.

4.5.2. Set of Properties as Source for Analogy

When a point in the discourse is identified as the end of the description of a concept, it is necessary to consider if we are in the case where the concept is unknown and an analogy can be used to clarify his meaning, in the case where the concept is atypical for its class, or neither one.

In order to know in which case we are, it is required to study the properties that form the description of the entity in the discourse. However, this properties are not only those explicitly mentioned in the discourse. If the concept was previously known, many of its known properties are assumed by the reader or listener, and can be considered for subsequent reasoning.

We therefore consider that the properties to be taken into account for an entity mentioned in the discourse are both the explicitily mentioned properties, and the implicitly mentioned when giving the class name. For example, when a cat is mentioned in any discourse, the reader or listener performs a series of inferences about what to be a cat means: it is an animal with four legs, it is surely sneaky, etc.

However, sometimes inferred properties may contradict the mentioned ones, and therefore are not applicable to the specific instance of the class that appears in the discourse. An example would be a cat that only has three legs because he lost one in an accident, for example. In order to obtain the set of implicit properties that do not contradict the explicitly mentioned it will be necessary to resort to general knowledge in a similar way as was done in comparison.

4.5.3. Identifying Targets for Analogy

In order to establish an analogy, it is necessary to construct a domain from the original text and look for a target domain that could provide valid analogies. To find the initial domain, the information corresponding to the entity for which the analogy is being sought must be enriched, both with specific information from the context and information extracted from the general knowledge used by the system. It would include related concepts, being a rich conceptual network that represents all the information related to the concept.

When looking for the target domain of the analogy, it is not enough to find a possible match between domains to consider that it is valid to give rise to an analogy. Depending on the situation, an analogy is only valid if it meets certain criteria.

When the purpose of the analogy is to emphasize an atypical entity, the properties that make it atypical must be present in the target concept of the analogy. In fact, in order to understand properly the relationship between the atypical properties and the analogy, these properties have to be considered important or relevant in the target class.

If the purpose of the analogy is to facilitate the understanding of an unknown concept using a known one, it will be necessary to use some kind of model about the user to verify that the target concept is properly known by the user.

The analog capabilities of the architecture has been tested using the domains from Sapper. When looking for an analogy for a concept from the discourse, we try to find correspondences between the initial domain and each of the possible target domains using Mapper agent.

4.5.4. Realising Analogies

The point of the discourse where the insertion of an analogy must be considered is when the description of an entity is finished. If we consider that descriptions are composed of copulative messages, the analogy must be placed after the last message that mentions any of the properties of the entity.

4.5.5. Examples of Generated Analogies

We present examples of analogies generated from discourses that had the two situations we have considered appropriate for the insertion of an analogy.

Examples of Analogies for Unknown Concepts

Out of the complete data set used in Sapper, two well known domains have been used to test the analogy capabilities of our system: Star Wars and King Arthur saga. The former has been chosen to represent a very simple story to be rendered by our generation system, supposing that it is an unknown domain for the reader or listener. The latter has been then used to find analogies with the first one and facilitate comprehension. A fragment of both domains can be seen in Table 4.1.

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

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

Table 4.1: Examples of relations for some of the concepts from both domains

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 is shown in Table 4.2. For each association we can see the list of relations that have produced the mapping.

By following the algorithm explained before, and using the properties and mapping of Tables 4.1 and 4.2, the resulting text in the Star Wars domain is obtained:

> *Luke Skywalker was a young jedi knight. 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.*

Examples of Analogies for Enhancing Atypical Information

We present examples of analogies used to emphasize atypical properties of an element of the discourse. For each original text we show the final result after establishing analogies.

Imagine the following initial text:

> *He proposed her a contract. It was illegal. It was dangerous.*

After the enrichment using analogies, the generated text is the following:

> *He proposed her a contract. It was illegal. It was dangerous.* <u>*The contract was like a burglary.*</u>

Mapping	**Relations**		**Intensity**
obi_wan_kenobi ↔ merlin	good wise person man	powerful old magician	0.52
storm_trooper ↔ knight	warrior person	man	0.66
light_saber ↔ excalibur	hand_held long	narrow weapon	0.73
han_solo ↔ lancelot	skilful handsome man	brave young person	0.43
princess_leia ↔ guinnevere	beautiful royal_personage woman	young person	0.63

Table 4.2: Obtained mapping for Star Wars and King Arthur Saga

In the domains we are working with, a contract is considered as something legal, so this specific contract is atypical for its class. Using the Mapper agent we obtain the analogy that the contract is like a burglary: illegal and dangerous.

Another input text would be:

The policeman was unethical and dishonest.

After the enrichment using analogies, we obtain the following text:

The policeman was unethical and dishonest. <u>He was like a criminal.</u>

In this case, policemen are supposed to be ethical and honest in the general knowledge. When looking for an analogy for the atypical properties of this policeman, the Mapper agent finds the concept criminal.

Chapter 5

Discussion

The work presented in this thesis goes beyond the task of referring expression generation. The process of reference generation with descriptive or distinguishing function is dependent of tasks such as lexicalization (when realising references), sentence planning (when the sentences have been modified to include a comparison or analogy), content determination (which determine the domains used to find analogies), etc. Although most of these issues were not within the scope of this work, it is appropriate to mention and discussed them for future work.

We discuss below the solutions provided for the generation and evaluation of basic referring expressions (section 5.1), and the descriptive information about entities by using rhetorical figures (section 5.2). Finally we will also provide a brief discussion about the impact of the inclusion of such figures in the process of automatic generation of text (section 5.3).

5.1. Generating and Evaluating Basic Referring Expressions

In chapter 3 we implemented and evaluated various algorithms to solve the complete problem of reference generation by separating it in three different tasks: choice the level of abstraction, attribute selection and lexicalization. In this section we discuss this work, and issues that might be considered in the future.

The work presented for the choice of abstraction level focused on static situations where there is no context such as which entities have been mentioned previously. But the generation of referring expressions is usually part of a more complex process of language generation as in the generation of dialogue or discourse. In these cases the contrast set could be restricted using different kinds of heuristics. One of them could be the assignment of salience weights used by Krahmer y Theune (2002), where each entity in the domain is assigned a certain degree of salience. They use a dynamic contrast set called *context set* that is composed of the entities in the world that have equal or greater salience that the intended referent.

With respect to the evolutionary algorithm for attribute selection, it would be also interesting to compare our solution with different approaches found in the literature, as for example (Reiter y Dale, 1992) or (Krahmer y Theune, 2002) for referring expression generation. From the comparisons with these algorithms may emerge information for new genetic operators and fitness functions.

Another possibility for evaluating the population at the end of each generation is the use of neural networks to reproduce human evaluation of the quality of the resulting reference. Neural networks have been previously used in the evaluation of individuals for evolutionary algorithms. Some examples in the field of automatic generation of poetry can be found in (Levy, 2001) and (Manurung, 2003). It would be possible to use a corpus to train a neural network to measure how successful the generated reference is in comparison with the rest of references from the corpus.

Although we have outlined the idea of using a case-based solution in order to take into account different styles of reference generation, it has not been completely carried out within this work. One possible way of doing this might be to use a model of the speaker perception to guide the process of generating the referring expression. For instance, a professional of the fashion world might describe people in terms of the clothes they are wearing whereas a doctor might rely more on their physical complexion. We will then train the module using a set of referring expressions or another depending on the style we want to use.

If we examine the evaluation results obtained by the participating systems in the competitive evaluation tasks for the generation of referring expressions, it could be inferred that the systems with better intrinsic evaluation values are not always the more appropriate from the point of view of the extrinsic evaluation using human evaluators. This is probably due to the difficulty for creating a corpus that is representative of how humans generate references in general. Despite it has been made using a high number of evaluators, it is not possible to cover all possibilities. Moreover, the fact that references in the corpus come from many different solutions can cause not uniformed evaluation results.

5.2. Generating and Evaluating References using Rhetorical Figures

In chapter 4 we studied how to deal with the generation of referring expressions using information from other domains to generate comparisons and analogies. In this section we discuss topics related with this chapter and pending issues to be considered in the future.

The key to a possible improvement lies in the multiagent architecture. The OAA architecture (Martin et al., 1999) implementation described 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 general knowledge required for the generation of rhetorical figures was obtained from the domains created by Tony Veale for his system Sapper. However, we could use other sources of knowledge for this function. Available common sense ontologies such as Cyc (Lenat, 1995), Open Mind Common Sense (Singh, 2002) and Thought Treasure (Mueller, 1998) could be used. However, in an ontology it would not be possible to find intensities for relations as in Sapper. Therefore, some kind of mechanism to decide if a property is important for the definition of a concept would be required. Without this information, comparisons and analogies could only be generated without taking into account if a property is remarkable or not.

The generation of comparisons using concept glosses of WordNet presents some problems. The responsible agent searches for concepts containing the adjective in its gloss to make the comparison. For this purpose it checks that these concepts have as hypernyms any of the hypernyms of the concept we want to compare. This ensures that the two concepts for the comparison are similar, or at least related. In addition, it is assumed that if a concept has an adjective in its gloss, this word must be important for defining the concept. However, sometimes strange comparisons are obtained as for example *it was unpleasant as a pill*. The gloss corresponding to *pill* in WordNet is *something unpleasant or offensive that must be tolerated or endured; "his competitor's success was a bitter pill to take"*, but the final realization of the comparison makes difficult to realize we are referring to this definition of *pill*.

Avoiding these departures from meaning is somehow achieved by using structure alignment for searching the target of the comparisons. Concepts that are used as targets in the comparisons are always similar or analogous to the entity they are being compared with, so it is more difficult to find confusing comparisons. But it also prevents the appearance of more creative comparisons as *the butcher is clean as a operating theater* or *the surgeon is precise as a scalpel* could be.

In the description of the architecture presented in section 4.1, we considered a high-level approach to the identification and introduction of an analogy between a single concept from a given domain and a concept in a target domain. In the examples presented in section 4.5.5, it became apparent that, given two domains which are structurally analogous, the process described returns a number of associations between concepts in both domain. From a stylistic point of view, it is clear that a text in which all possible associations between the domains are included is overloaded with analogy. This suggests that the heuristics being considered for the selection and location of analogies must be revised in search for more natural results.

In the case of references generated by using rhetorical figures based on similarities between domains, no appropriate metrics for evaluation have been found in the literature. As future work we could address the problem of defining both intrinsic and extrinsic metrics to measure these constructions. One possibility would be to create a corpus of comparisons and analogies applied to the generation of references where it was possible to compare the automatically generated ones. However, it might be quite difficult. In addition, we have observed for the basic references that a corpus is not always a good reflection of how humans perform these tasks. Another possibility might be to make some kind of extrinsic evaluation on humans to measure how appropriate are references depending on the generated knowledge about different domains.

5.3. Interactions between Comparisons and Analogies and the Automatic Generation of Text

Considering the multiagent architecture presented for the generation of rhetorical figures, any system that needs to use the implemented functionality must have presence in the multiagent system. In this case it is possible to implement a TextGenerator agent which manages the process of generating text. It could simply be a wrapper for the original module of generation. This would be the agent that triggers the control flow of the process. The TextGenerator follows the control flow of the NLG module, interacting with agents from the architecture when it needs information about the different concepts in the discourse.

The whole process of generating text for the entities in a discourse starts after the stage of Discourse Planning. At this point decisions have taken about what information is going to appear in the final text, and how that information will be organized using messages. Here it has been decided what entities will be mentioned in the text and where, as well as the properties to be used for their descriptions. From this initial discourse organized as a set of messages, the entities and their descriptions can be considered to decide what kind of enrichment can be carried out. The insertion of comparisons or analogies will therefore interfere with the information selection and organization carried out by the Discourse Planning stage of the process.

Once the discourse has been enriched with analogies and comparisons, the treatment of the entities of the discourse passes through the task of Referring Expression Generation. This task is performed by relying on existing conceptual information about entities and the context of what has been said in the text. However, after the enrichment the discourse would contain entities that are not in the original conceptual information for the text. This may be a problem as there is not enough information to perform the process of reference generation. In fact, these new entities in the discourse, which will correspond to the target concepts of comparisons and analogies, should not enter into the process of generating references.

Chapter 6

Conclusions and Future Work

One of the key aspects analyzed in this work is that references in a text can have two different functions or objectives. The distinctive function of references has been defined as the intention of univocally identifying a referent in contrast to others. The descriptive function of references has been defined as the insertion of information that is not necessarily distinctive in a reference, but that is providing descriptive or extra information about the entity that is considered important.

The distinctive function of references has been widely studied in literature while the descriptive one has not. This is due to the fact that the goal of identifying referents is a basic one in natural language generation, and more specifically in referring expression generation. The definition of this task is quite simple, it is only dependent on the discourse context, and it can be automatically computed with deterministic algorithms that produce a result that is either valid or not. On the contrary, the descriptive function of references is harder to define, as it is a vague function that depends on the context in which the discourse is presented, but not always on the discourse context. The decision of adding descriptive information is not only dependent of the information presented in the discourse, but it is also related with the intention of the writer or speaker when creating the discourse.

Every reference in a text has a distinctive and a descriptive part, each of them corresponding to one of the goals it can address. The relation between these two parts in the same reference is a problem that has not been studied in the literature either. To deal with this problem it would be necessary to start studying the two goals separately to find accurate definitions, solutions to achieving them, and metrics to evaluate their correctness. This separate study is precisely the main contribution of this work.

In this work we have studied the distinctive function of references on the generation of basic references in the form of nominal phrases, and the descriptive function of references enhanced using rhetorical figures based on similarities between domains. This work is intended as a first step towards the study of the complete process of referring expression generation involving both the descriptive and the distinctive function of references. The following sections discuss in more detail the main conclusions for each of these two functions.

6.1. Assessment of Basic Reference Generation

In the case of the distinctive function of basic references, a lot of work can be found about its definition, possible solutions and evaluation metrics. We have studied the most common solutions found in literature, and we have proposed improvements that have resulted in good evaluation values. We have also found that this process results in interactions beyond the task of Referring Expression Generation. For example, when deciding the abstraction level to be used in a reference, we are addressing issues more related with the conceptual information available during Content Determination than with reference generation.

In addition, when choosing the words to express the resulting references, decisions about its syntactic form or which word can be used is more related with the whole lexicalization process addressed in the NLG pipeline. In this work we have considered that for the generation of nominal referring expressions with the only goal of distinguishing entities, it is common to consider a division of the problem in three different parts: choice of abstraction level, attribute selection and lexicalization. Some of them are beyond a strict definition of the REG stage in the NLG pipeline, as they interfere in conceptual and syntactic choices, for example.

In section 3.1 we discussed the importance of using different abstraction levels for the references depending on the context. In situations where the information required for Referring Expression Generation (REG) is stored in a taxonomy or hierarchy, it is possible to rely on this information to determine if it is better to refer to a referent using a very specific type (e.g. *doberman*), or to use a more generic value (e.g. *dog*). This decision depends directly on the situation. If there are more dogs of different breeds in the discourse context, the specific reference would be more appropriate. If there are only other types of animals, the generic reference would be enough and would not give more information than required.

In this work we have studied the use of ontologies to deal with this problem as a first step in the complete reference generation process. As stated in the incremental algorithm, the abstraction level for the type or *basic level value* is obtained from the knowledge base or the user model. We have implemented a dynamic method to obtain this value that only depends on the knowledge available about the world.

As we studied in section 3.2, the selection of attributes to be mentioned in a reference can be considered as a search problem. In this work three possible solutions for this task were proposed: using relative groupings of the attributes when choosing which ones are mentioned (section 3.2.1), taking into account the value of an attribute when deciding whether to add it to a reference (section 3.2.2), and the use of evolutionary algorithms to identify which is the most appropriate attribute set (section 3.2.3). The first two were based on the incremental algorithm by Reiter y Dale (1992). Some conclusions can be extracted from the results obtained for these solutions.

For the solutions based on the incremental algorithm, it is clear that the way in which the possible attributes are considered has a strong influence in the results obtained. In the case of the solution based on relative groupings, different clusters of attributes were identified. The results were very different depending on the order of consideration chosen for the clusters in the algorithm, whereas the order of attributes inside the clusters had little repercussion in the results. When the influence of attribute values was also considered, the results showed that, depending on their value, some attributes are more likely to be used than others.

When considering the use of evolutionary algorithms to solve the problem of attribute selection, the main problem we faced was the choice of the fitness function. This kind of algorithm resulted to be a good approach for a problem where the goodness of a solution can be computed, but the process of obtaining it cannot be formalised. The approach presented in this work dealt adequately with the task and obtained reasonable results.

In section 3.3 we discussed different aspects about lexicalization of references. In section 3.3.1, a lexicalization based on the most used options was presented, showing that the lexicalization goes beyond the simple choice of words. Finally, the importance of style when deciding which words are appropriate for expressing something was introduced in section 3.3.2.

The lexicalization of references using the most used option shows some dependencies that make almost impossible to separate it from other NLG tasks such as Referring Expression Generation or Syntactic Choice. When generating the lexicalization for a reference, it is necessary to choose which determiners must be used (Referring Expression Generation), which specific words are chosen (Lexical Choice) and what syntactic structure is the most appropriate (Syntactic Choice). Although these decisions are quite independent, all of them are taken into account when lexicalizing the conceptual representation of a reference.

Another decision that has been addressed during the lexicalization of referring expressions is the style of the reference that will be used in each specific situation. Each person has a different way of expression, and this results in several possible lexicalizations that are equally good. The only difference between them would be the style of discourse in which they are included. Some way of generating references using different styles can thus be contemplated. In this work we have proposed the use of a case-based reasoning approach to deal with this issue. The underlying idea is to train the system using references generated according to a specific style, so that the generation can be based in the style that the system understands. There are so many correct ways of creating a reference, that it is not possible to provide a general solution that deals with all of them in the same appropriate way.

6.2. Assessment of Reference Generation using Rhetorical Figures

In chapter 4 we addressed the use of rhetorical figures based on similarities between domains in order to enhance the references in a text with descriptive information. More specifically, we considered the use of comparisons and analogies to deal with this task.

The purpose of comparison and analogy as studied in this work is to sharpen the hearer's existing knowledge of a concept by relying on similarities with other concepts. Comparisons and analogies are, apparently, a good approach for this purpose as they remark a property by searching for an entity that also holds it. The structural alignment between entities has also been studied to avoid incorrect inference issues that arise when there is a shallow consideration of types and similarities.

The process of generating a rhetorical utterance for referring to a specific concept involves several tasks that are treated 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 comparisons or analogies. Once it is found, and the mapping between the two domains is established, it is necessary to generate a set of possible references for each concept susceptible of being referred through a rhetorical figure. This set of rhetorical references must be studied and evaluated in terms of clearness and suitability, so that inappropriate ones 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 available comparisons or analogies to refer to the concept at that stage or not, always avoiding loss of meaning or unnecessary ambiguity.

We have chosen to implement our ideas in the form of a multiagent system. This kind of architecture has allowed us to use easily knowledge resources and previously implemented systems that were required for the process of analogy and comparison generation. As knowledge resources we have used WordNet and Sapper's domain to obtain the information required for the generation of the comparisons and analogies.

6.3. Evaluation of the Generated References

In chapter 3 we evaluated and discussed the proposed solutions for basic reference generation using common metrics in the area. Most of them were based on the comparison of the generated references with a corpus of human-authored references in different domains. In addition, task-based evaluations were also carried out to test how people react to the generated references to distinguish an entity in a given situation. Results showed that systems that performed well in comparison with the corpus, were not always the best suited to be used for real people. Moreover, the corpus was no homogeneus enough, in such a way that solutions that performed well for some kind of references in the corpus, obtained bad results for other ones in the same corpus. The conclusion obtained from these results was that a corpus created from references that were authored by different people can hardly be seen as a general example of how things should be done, but when used taking into account personal styles and viewpoints, it could be used more fruitfully.

In the case of descriptive information added to references by using rhetorical figures, there was no available knowledge (such as a corpus) that could be considered as an exemplification of how people use this kind of figures to enhance information or describe unknown entities. As a result, it was not possible to carry out a formal evaluation of the kind done for basic references. Even a knowledge resource like a corpus could not be enough to evaluate the generated rhetorical structures. In the case of basic references, where the task is quite delimited and specific, evaluation using corpus has not resulted to be the best option as expected. In the case of the rhetorical figures, that are by themselves quite subjective, a corpus would not be enough to evaluate the solutions proposed and some other metrics must be searched.

6.4. Interactions with the Complete Text Generation Process

In the generation and subsequent use of the referring expressions, both for distinction and description, the task of Referring Expression Generation (REG) is not the only one to be considered, but Content Determination (CD) should also be addressed.

CD is responsible for deciding what information is conveyed in a text on the basis of all available information. From the viewpoint of the descriptive function, the decision about whether it is appropriate or necessary to make a description and what information it is going to include may be taken in the final stages of CD. At that point it has already been decided which entities will appear in the text and how to talk about them.

Thus, when generating a reference to a concept, the REG stage should take into account what has been described previously about this concept and therefore what information known to the reader can be used to refer to it. In most NLG systems the relationship between these two tasks is unidirectional.

The most extended architecture for NLG systems is a sequential architecture or *pipeline*, where the choice of content for the discourse and its organization using messages is done early in the process. The subsequent steps in the generation process use this information, including the REG stage, without the possibility of reviewing the outcome of previous stages of the pipeline. From the point of view of the relationship between the CD and the REG stages, the flow of information provided by this architecture may not be appropriate on certain occasions. For example, we could consider a case in which during the REG stage the system realizes that there is not enough information to distinguish one entity from the rest that are in the same context. With a sequential architecture the problem have no easy solution and the generated expression would be ambiguous.

There are two possible solutions for this problem. In a first approach, CD would be responsible of checking whether the selected information is enough for generating distinctive references, in addition to deciding what information is important for the discourse as a whole. Alternatively, CD could select little or no information for the descriptive part of the references, being REG responsible for requesting the inclusion of new information when necessary to generate references.

6.5. Future Work

The first issue that should be addressed in future work is the combination of the solutions for basic references with distinctive function and rhetorical figures providing descriptive information. This requires to complete the study of basic references addressing their use for descriptive functions, and the interactions of descriptive rhetorical figures in the distinctive part of references. Only when the repercussions of both functions in both types of references were completely studied, it would be possible to provide solutions for their combination.

Some lines of future research involving only basic references are the use of some kind of user model representing the expertise level of the reader or hearer in a specific domain when determining the abstraction level, and a more accurate definition for the fitness function in the evolutionary algorithm. It is also important to note that, in this work, we have only considered the generation of nominal references to refer to entities in a discourse. There are another type of references that could also be used as they provide fluidity and naturalness to the texts: pronominal references, relations as modifiers for nominal phrases, proper nouns, etc.

In our approach towards the generation of analogical 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 elements by different kinds of relations. For example, in the domains we used it is possible to 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 analogical reference is created. This kind of information makes it easier to understand the generated references.

A future research line can be the development and implementation of a corpus of rhetorical figures, but it falls beyond the scope of this work. Even with such a knowledge source as a corpus of rhetorical figures used to refer to entities in a discourse, it would be difficult to consider it a measurable reference of how people use comparisons and analogies. Another way of evaluating the generated figures would be to perform some kind of experiment with human evaluators that will decide if the generated comparisons are analogies are fulfilling their goal in each of the cases considered in this work.

The work addressed in this thesis has not been tied to any specific domain. We provide generic solutions for the stated problems in such a way that it will be possible to apply them to text generation for any domain. However, every different type of text has different problems and requirements to be taken into account.

A special type of texts that are interesting to explore are narrative texts. Most of the existing solutions for REG based in attribute selection assume that the features of the involved objects do not change through time. Therefore, these methods are valid in the generation of descriptive texts or dialogs where the situation in which they appear does not vary.

However, in the case of narrative texts, the attributes do not remain the same over the time. Narrations are based in sequences of events that occur, changing the state of the world. Consequently, characters and objects involved in the stories are born, grow up, die, change their form, appear, disappear, etc.

Thus, the generation of referring expressions in narrative texts cannot apply a simple solution based on attribute selection because the attributes can change during the discourse either explicitly (for example, after *the princess heard the scream and she got scared*, the princess changes her state to *scared*) or implicitly (for example, after *the prince killed the dragon* the state of the dragon changes to *dead* even when it has not been stated directly). Aspects such as timelines (story time vs. discourse time) or existing and mentioned features are only examples of issues that must be addressed in future work about references in narrative texts.

Another rhetorical figure based on similarity between domains that can be addressed following the work done on comparisons and analogies is metaphor. Metaphors are quite similar to analogies but they are usually used in a different way. Instead of having messages that compare the two concepts involved as in analogies, metaphors are used instead of the original concept.

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 instance, in the examples presented in this work 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. It has to be studied how this kind of misconceptions can be avoided. For example, the metaphor *the steely light saber* instead of *Excalibur* is more understandable because a difference between the concepts (to be made of steel) is being mentioned.

Bibliografía

Agnar, A., y Enric, P. (1994). Case-based Reasoning : Foundational issues, methodological variations, and system approaches. *AI Communications*, *7*(1).

Belz, A., y Gatt, A. (2007). The attribute selection for GRE challenge: Overview and evaluation results. En *Proc. 2nd UCNLG Workshop: Language Generation and Machine Translation (UCNLG+MT)*, (pp. 75–83). Copenhaguen, Denmark.

Belz, A., y Gatt, A. (2008). REG Challenge 2008: Participants pack. `http://www.nltg.brighton.ac.uk/research/reg08/` [24/2/2009].

Bohnet, B., y Dale, R. (2004). Referring expression generation as a search problem. En *Proceedings of the Australasian Language Technology Workshop 2004*.

Brachman, R. J., Mcguinness, D. L., Patel-schneider, P. F., Resnick, L. A., y Borgida, A. (1991). Living with classic: When and how to use a KL-ONE-like language. *Principles of Semantic Networks*, (pp. 401–456).

Callaway, C., y Lester, J. (2001). Narrative prose generation. En *Proceedings of the 17th IJCAI*, (pp. 1241–1248). Seattle, WA.

Cheng, H. (1998). Embedding new information into referring expressions. En *ACL-36: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, (pp. 1478–1480). Morristown, NJ, USA: Association for Computational Linguistics.

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

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

Gatt, A., Belz, A., y Kow, E. (2008). The TUNA challenge 2008: Overview and evaluation results. En *Proceedings of the 5th International Conference on Natural Language Generation, INLG-08*, (pp. 198–206). Ohio, USA.

Gatt, A., Belz, A., y Kow, E. (2009). The TUNA-REG Challenge 2009: Overview and evaluation results. En *Proceedings of the 12th European Workshop on Natural Language Generation, ENLG-09*. Athens, Greece.

Gatt, A., van der Sluis, I., y van Deemter, K. (2007). Evaluating algorithms for the generation of referring expressions using a balanced corpus. En *Proc. of the 11th European Workshop on Natural Language Generation (ENLG 07)*.

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

Gervás, P., Hervás, R., y León, C. (2008). NIL-UCM: Most-frequent-value-first attribute selection and best-scoring-choice realization. En *Referring Expression Generation Challenge 2008, Proc. of the 5th International Natural Language Generation Conference (INLG'08)*.

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

Hervás, R., y Gervás, P. (2007). NIL: Attribute selection for matching the task corpus using relative attribute groupings obtained from the test data. En *First NLG Challenge on Attribute Selection for Generating Referring Expressions (ASGRE'07), Proc. of the MT Summit XI Workshop: Using Corpora for Natural Language Generation*. Copenhagen,Denmark.

Hervás, R., y Gervás, P. (2008). Degree of abstraction in referring expression generation and its relation with the construction of the contrast set. En *Proc. of the 5th International Natural Language Generation Conference (INLG'08)*.

Hervás, R., y Gervás, P. (2009). Evolutionary and case-based approaches to REG: NIL-UCM-EvoTAP, NIL-UCM-ValuesCBR and NIL-UCM-EvoCBR. En *Generation Challenges 2009, Proc. of the European Natural Language Generation Conference 2009 (ENLG'09)*.

Holland, J. (1992). *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, Massachusetts, Second Edition: MIT Press.

Jurafsky, D., y Martin, J. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

Krahmer, E., y Theune, M. (2002). Efficient context-sensitive generation of referring expressions. En *Information Sharing: Givenness and Newness in Language Processing*, (pp. 223–264).

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

Lenz, M., y Burkhard, H.-D. (1996). Case Retrieval Nets: Basic ideas and extensions. En *KI - Kunstliche Intelligenz*, (pp. 227–239).

Levy, R. (2001). A computational model of poetic creativity with neural network as measure of adaptive fitness. En *Proceedings of the ICCBR-01 Workshop on Creative Systems*.

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

Martin, D. L., Cheyer, A. J., y Moran, D. B. (1999). The Open Agent Architecture: A framework for building distributed software systems. *Applied Artificial Intelligence*, *13*(1-2), 91–128. OAA.

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

Milosavljevic, M. (1999). *The Automatic Generation of Comparisons in Descriptions of Entities*. Ph.D. thesis, Department of Computing, Macquarie University, Sydney, Australia.

Milosavljevic, M. (2003). Defining comparison. En P. Slezak (Ed.) *Proceedings of the Joint International Conference on Cognitive Science with the Australasian Society for Cognitive Science*. University of New South Wales.

Mueller, E. (1998). ThoughtTreasure: A natural language/commonsense platform.

Papineni, K., Roukos, S., Ward, T., y Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. En *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, (pp. 311–318). Morristown, NJ, USA: Association for Computational Linguistics.

Passonneau, R. (2006). Measuring agreement on setvalued items (MASI) for semantic and pragmatic annotation. En *Proceedings of the 5th Language Resources and Evaluation Converence (LREC'06)*.

Pereira, F., Hervás, R., Gervás, P., y Cardoso, A. (2006). A multiagent text generator with simple rhetorical habilities. En *AAAI-06 Workshop on Computational Aesthetics: AI Approaches to Beauty & Happiness*.

Pereira, F. C. (2007). *Creativity and AI: A Conceptual Blending Approach*. Mouton de Gruyter.

Portet, F., Reiter, E., Hunter, J., y Sripada, S. (2007). Automatic generation of textual summaries from neonatal intensive care data. En *AIME '07: Proceedings of the 11th conference on Artificial Intelligence in Medicine*, (pp. 227–236). Berlin, Heidelberg: Springer-Verlag.

Reiter, E., y Dale, R. (1992). A fast algorithm for the generation of referring expressions. En *Proceedings of the 14th conference on Computational linguistics*. Nantes, France.

Reiter, E., y Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press.

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

van Deemter, K., van der Sluis, I., y Gatt, A. (2006). Building a semantically transparent corpus for the generation of referring expressions. En *Proceedings of the 4th International Conference on Natural Language Generation (Special Session on Data Sharing and Evaluation), INLG-06*.

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. En *GWC'2006, the 3rd Global WordNet Conference*.

Appendix A

Publications

The publications in which this work is based are shown in this appendix.

A.1. Natural Language Generation

1. García, C., Hervás, R., Gervás, P.: **"Una arquitectura software para el desarrollo de aplicaciones de generación de lenguaje natural"**. Sociedad Española para el Procesamiento del Lenguaje Natural, Procesamiento de Lenguaje Natural, n^{o} 33, Septiembre de 2004, pp. 111-118. ISSN: 1135-5948.

2. Gervás, P., Díaz-Agudo, B., Peinado, F., Hervás, R.: **"Story Plot Generation based on CBR"**. Journal of Knowledge-Based Systems 18, 4-5: Special issue AI-2004, pp. 235-242. Elsevier Science, 2005. ISSN: 0950-7051.

3. Hassan, S., León, C., Gervás, P., Hervás, R.: **"A Computer Model that Generates Biography-Like Narratives"**. International Joint Workshop on Computational Creativity (IJWCC'07), London, 2007.

4. Hervás, R., Gervás, P.: **"Descripción de Entidades y Generación de Expresiones de Referencia en la Generación Automática de Discurso"**. Sociedad Española para el Procesamiento del Lenguaje Natural, Procesamiento de Lenguaje Natural, n^{o} 41, Septiembre de 2008, pp. 217-224. ISSN: 1135-5948.

5. Dionne, D., de la Puente, S., León, C., Hervás, R., Gervás, P.: **"A Model for Human Readable Instruction Generation Using Level-Based Discourse Planning and Dynamic Inference of Attributes Disambiguation"**. 12th European Workshop on Natural Language Generation (EWNL'09), Marzo de 2009.

A.2. Referring Expression Generation

1. Hervás, R., Gervás, P.: **“Uso flexible de soluciones evolutivas para tareas de Generación de Lenguaje Natural”**. Sociedad Española para el Procesamiento del Lenguaje Natural, Procesamiento de Lenguaje Natural, nº 35, Septiembre de 2005, pp. 187-194. ISSN: 1135-5948.

2. Hervás, R., Gervás, P.: **“Agent-based Solutions for Natural Language Generation Tasks”**. XI Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 05), Santiago de Compostela, España, Springer LNAI Series, 2005.

3. Hervás, R., Gervás, P.: **“Case Retrieval Nets for Heuristic Lexicalization in Natural Language Generation”**. En Bento, C., Cardoso, A., Dias, G., eds: Progress in Artificial Intelligence (Proc. of the 12th Portuguese Conference on Artificial Intelligence (EPIA 05)), Covilha, Portugal, Springer-Verlag, LNAI 3808, pp. 55-66, 2005.

4. Hervás, R., Gervás, P.: **“An Evolutionary Approach to Referring Expression Generation and Aggregation”**. (Póster) En Wilcock, G., Jokinen, K., Mellish, C., Reiter, E., eds.: Proceedings of the 10th European Workshop on Natural Language Generation, pp. 168-173, Aberdeen, Scotland, 8-10 August, 2005.

5. Hervás, R., Gervás, P.: **“Case-Based Reasoning for Knowledge-Intensive Template Selection During Text Generation”**. 8th European Conference on Case-Based Reasoning (ECCBR 06), Fethiye, Turkey, Springer-Verlag, LNAI 4106, September 2006.

6. Francisco, V., Gervás, P., Hervás, R.: **“Automatic Knowledge Acquisition in Case-Based Text Generation”**. Proceedings of the ECCBR 06 Workshop on Textual Case-Based Reasoning, Turquía, pp. 68-77, Septiembre de 2006.

7. Francisco, V., Gervás, P., Hervás, R.: **“Dependency Analysis and CBR to Bridge the Generation Gap in Template-Based NLG”**. Computational Linguistics and Intelligent Text Processing (CICLing 2007), Méjico, Springer-Verlag, LNCS 4394, pp. 432-443, 2007.

8. Hervás, R., Gervás, P.: **“Degree of Abstraction in Referring Expression Generation and its Relation with the Construction of the Contrast Set”**. (Póster) En Proc. of the Fifth International Natural Language Generation Conference (INLG’08), pp. 161-164, Ohio, USA, Junio 2008.

A.3. Rhetorical Figures

1. Pereira, F.C., Hervás, R., Gervás, P., Cardoso, A.: **"A Multiagent Text Generator with Simple Rhetorical Habilities"**. In the AAAI-06 Workshop on Computational Aesthetics: AI Approaches to Beauty & Happiness, July 2006.

2. Hervás, R., Pereira, F.C., Gervás, P., Cardoso, A.: **"Cross-Domain Analogy in Automated Text Generation"**. Proceedings of the 3rd Joint Workshop on Computational Creativity, Italia, Agosto 2006.

3. Hervás, R., Pereira, F.C., Gervás, P., Cardoso, A.: **"A Text Generation System that Uses Simple Rhetorical Figures"**. Sociedad Española para el Procesamiento del Lenguaje Natural, Procesamiento de Lenguaje Natural, nº 37, Septiembre de 2006, pp. 199-206. ISSN: 1135-5948.

4. Hervás, R., Robinson, J., Gervás, P.: **"Evolutionary Assistance in Alliteration and Allelic Drivel"**. En Giacobini, M. et al., eds: Applications of Evolutionary Computing (Proc. of EvoMusArt 2007, EvoWorkshops 2007, dentro de Evo* 2007 (conferencia múltiple en el campo de la Programación Evolutiva)), Valencia, España, Springer-Verlag, LNCS 4448, pp. 537-546, 2007. ISSN: 0302-9743.

5. Hervás, R., Costa, R., Costa, H., Gervás, P., Pereira, F.C.: **"Enrichment of Automatically Generated Texts using Metaphor"**. En A. Gelbukh, A.F. Kuri Morales, eds: MICAI-07: Advances in Artifical Intelligence (Proc. of 6th Mexican International Conference on Artificial Intelligence (MICAI-07)), Méjico, Springer-Verlag, LNAI 4827, pp. 944-954, 2007. ISSN: 0302-9743.

A.4. Competitive Evaluation for the Generation of Referring Expressions

1. Hervás, R., Gervás, P.: **"NIL: Attribute Selection for Matching the Task Corpus Using Relative Attribute Groupings Obtained from the Test Data"**. First NLG Challenge on Attribute Selection for Generating Referring Expressions (ASGRE), UCNLG+MT Workshop, Machine Translation Summit XI, Copenhague, 2007.

2. Gervás, P., Hervás, R., León, C.: **"NIL-UCM: Most-Frequent-Value-First Attribute Selection and Best-Scoring-Choice Realization"**. Referring Expression Generation Challenge 2008, Proc. of the 5th International Natural Language Generation Conference (INLG '08), Ohio, USA, Junio de 2008.

3. Hervás R, Gervás P.: **"Evolutionary and Case-Based Approaches to REG: NIL-UCM-EvoTAP, NIL-UCM-ValuesCBR and NIL-UCM-EvoCBR"**. Generation Challenges 2009, Proc. of the 12th European Workshop on Natural Language Generation (ENLG '09), Atenas, Grecia, Abril de 2009.