THE UNIVERSITY OF WOLVERHAMPTON
SCHOOL OF LAW, SOCIAL SCIENCES AND COMMUNICATIONS

UNIVERSIDADE DO ALGARVE
FACULDADE DE CIÊNCIAS HUMANAS E SOCIAIS

José Guilherme Camargo de Souza

Coreference Resolution for Portuguese using Parallel Corpora Word Alignment

A Project submitted as part of a programme of study for the award of
MA Natural Language Processing & Human Language Technology

Dr. Constantin Orăsan
Dr. Jorge Baptista

May 18th, 2011

Abstract

In Information Extraction, the target data must be found in a set of texts. In a text, the target information (or objects of interest) are linked in different ways in different places. The problem of determining which references point to which objects is one of the several challenges of the process. This problem is known as coreference resolution.

Several natural language processing applications may benefit from a coreference resolution system. Some of them are: machine translation, automatic summarisation, cross-document entity coreference, question answering, and information extraction. However, for the Portuguese language, there are few systems that perform coreference resolution with satisfactory results.

This study presents a system that automatically extracts coreference chains from texts in Portuguese without having to resort to Portuguese corpora manually annotated with coreferential data. In order to achieve this goal, it was implemented a method for automatically obtaining data for training a supervised machine learning coreference resolver for Portuguese. The training data is acquired by using an English-Portuguese parallel corpus over which the coreference chains annotated in the English part of the corpus are projected to the Portuguese part of the corpus.

The methodology is developed using a parallel corpus for the English and Portuguese language pairs with 646 texts of a scientific brazilian magazine. The coreference resolution system for Portuguese is tested using a corpus composed of 50 texts from the science section of a brazilian newspaper. Each text presents coreference chains manually annotated by at least two annotators. The evalution is made using two different coreference resolution scores such as MUC and CEAF.
Keywords: Coreference Resolution; Parallel Corpus; Word Alignment; Machine Learning.

Resumo

A área de Extração da Informação tem como objetivo essencial investigar métodos e técnicas para transformar a informação não estruturada presente em textos de língua natural em dados estruturados. Um importante passo deste processo é a resolução de correferência, tarefa que identifica diferentes sintagmas nominais que se referem a mesma entidade no discurso. A área de estudos sobre resolução de correferência tem sido extensivamente pesquisada para a Língua Inglesa (Ng, 2010) lista uma série de estudos da área, entretanto tem recebido menos atenção em outras línguas. Isso se deve ao fato de que a grande maioria das abordagens utilizadas nessas pesquisas são baseadas em aprendizado de máquina e, portanto, requerem uma extensa quantidade de dados anotados.

Embora diversas tentativas de desenvolvimento de sistemas de resolução de correferência tenham sido feitas como parte das competições MUC (explicitar esta sigla por extenso...), grande parte dos sistemas existentes utiliza abordagens que têm base em aprendizado de máquina (Ng, 2010). Tais abordagens são exequíveis para a Língua Inglesa, que apresenta diversos corpora anotados extensos o bastante para serem usados no treinamento de sistemas de aprendizado. No entanto, no que concerne às línguas como a Portuguesa, a qual não apresenta os recursos de anotação necessários, abordagens que utilizam aprendizado de máquina não podem ser utilizadas de forma efetiva. Como resultado, a maioria dos trabalhos para a Língua Portuguesa focalizam determinados tipos de resolução pronominal anafórica ou concentram atenção em problemas relacionados à resolução de correferência e à resolução anafórica, tais como a classificação da anaforicidade de expressões do discurso. Tanto quanto se sabe, o único corpus disponível anotado com informações correferenciais é o corpus Summ-It (Collovini et al., 2007). Além disso, o único trabalho que usa este corpus para o desenvolvimento de uma abordagem para o Português e que utiliza aprendizado de máquina para resolver correferência é (Souza et al., 2008). Os resultados reportados por tal estudo foram inferiores aos obtidos por sistemas do estado da arte para o Inglês. Esses resultados, muito provavelmente, se devem ao tamanho reduzido do corpus utilizado para o treinamento do modelo.

Esta dissertação apresenta um sistema que extrai cadeias de correferência automaticamente de textos escritos em Português sem lançar mão de corpora em Português anotado com informações correferenciais. A fim de atingir este objetivo, um método para obter os dados necessários para treinar um sistema de resolução de correferência baseado em aprendizado de máquina supervisionado é implementado. Neste projeto, os dados de treinamento são obtidos mediante a utilização de um corpus paralelo para o par de línguas Inglês-Português. No lado Inglês deste corpus, são anotadas cadeias de correferência, as quais são projetadas para o lado Português do corpus, de uma forma similar à adotada por Postolache

et al. (2006), que projeta cadeias do Inglês para o Romeno. Em contraste com o método desenvolvido por Postolache et al. (2006), o objetivo desta dissertação não é criar um recurso com anotação correferencial, mas implementar um sistema funcional que seja capaz de extrair cadeias de correferência de textos escritos em Português.

O primeiro passo do processamento identifica as cadeias de correferência na parte em Inglês do corpus paralelo. Um sistema de resolução de correferência para o Inglês é utilizado para anotar automaticamente as cadeias de correferência. A partir dessa anotação, o próximo passo é gerar pares de expressões (antecedente e anáfora) a fim de que essas possam ser projetadas na parte em Português do corpus paralelo. Essas projeções são então utilizadas para treinar um modelo baseado em aprendizado de máquina supervisionado. Apesar de o método apresentado nesta dissertação constituir sua base num corpus paralelo, a maior parte dos corpora paralelos disponíveis não apresentam alinhamento lexical. Tal alinhamento é necessário para que as projeções dos pares de expressões sejam efetuadas. Por esse motivo, torna-se necessário utilizar um sistema que implemente um algoritmo de alinhamento lexical para o par de línguas Inglês-Português.

O alinhamento lexical é utilizado no processo de geração de exemplos de treinamento dos pares de expressões em Inglês para o Português. Tendo em vista que erros são introduzidos na identificação de sintagmas nominais em Inglês por parte das ferramentas de pré-processamento e pelo processo de alinhamento, os sintagmas nominais do Inglês não são diretamente mapeados para sintagmas nominais do Português. Primeiro, um algoritmo de *matching* é utilizado para identificar quais são os melhores sintagmas nominais do Português a serem alinhados com um determinado sintagma nominal do Inglês. Uma vez que um par é identificado no lado Português do corpus, *features*, são extraídas, a fim de produzir os exemplos de treinamento para o algoritmo de aprendizado de máquina.

A metodologia adotada é desenvolvida por meio da utilização de um corpus paralelo Inglês-Português formado por 646 textos de uma revista brasileira de divulgação científica. O sistema de resolução de correferência para o Português é testado em um corpus composto por 50 textos da seção de ciência de um jornal brasileiro. Cada texto apresenta cadeias de correferência anotadas manualmente por, pelo menos, dois anotadores. A avaliação é feita com duas métricas diferentes de avaliação de resolução de correferência: MUC e CEAF.

Keywords: Resolução de Correferência; Corpus Paralelo; Alinhamento Lexical; Aprendizado de Máquina.

Acknowledgements

First, I would like to thank the support and love my family has sent, even an ocean away, during the period of the development of this thesis.

I want to express all my gratitude to Dr. Constantin Orăsan, who supervised me during the development of this dissertation, providing, comments, ideas, suggestions and support.

I also would like to thank Dr. Jorge Baptista, who co-supervised this work, for all the guidance and suggestions during the period I have been in Portugal.

I would like to thank all the friends that helped me either proofreading or supporting me to develop this project: Sheila Castilho, Miguel Angel Ríos Gaona, Wayne Marriot, Maja Oreskovic, Renate Tilia Ellendorff, and Ruth Domínguez. I would like to make a special acknowledgement to Wilker Aziz for all the help with the parallel corpus alignment tools and for all the suggestions he has made.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

It is indisputable that, nowadays, humans are dependent of the computational and information systems they have developed for a myriad of different purposes. Some of these systems rely on data stored and managed by them. Virtually all data that are produced are stored in some digital format: from biological to geographical, chemical and mathematical data, among others. Unfortunately, not all of these data are stored in a structured format, easily accessible and suitable for automatic processing. This is the case of texts from, for instance, newspapers, magazines, books, scientific articles, and others.

The field of Information Extraction (IE), a subarea of the Natural Language Processing (NLP) area, studies methods and techniques for turning the unstructured information present in natural language texts into structured data. An important task when analysing natural language texts is to identify the mentions to the discourse entities used throughout the texts. In other words, to understand the text sentences, it is necessary to develop methods for mapping the relations established between each mention and its corresponding entity.

This task is called reference resolution and, according to Jurafsky and Martin (2009, page 729) it is "the task of linking or clustering the mentions into sets that correspond to the entities behind the mentions". Rephrasing this definition, reference resolution consists of linking together all the entity mentions that occur in a text. Since its inception, two instantiations of the more general problem of reference resolution have been studied: anaphora resolution and coreference resolution. Anaphora resolution is the process of finding the antecedent of an expression in the discourse. Coreference resolution is defined as the task of finding

all referring expressions in a text and clustering them into coreference chains. In this work, the focus is on coreference resolution for the Portuguese language.

1.1 Motivation

There are not many systems for performing coreference resolution for Portuguese mainly due to the lack of resources for building them. Most of the works for the Portuguese language focus on certain types of pronominal anaphora resolution (Paraboni, 1997; Paraboni and Lima, 1998; Aires et al., 2004; Coelho, 2005; Chaves, 2007; Chaves and Rino, 2008; Santos, 2008; Cuevas et al., 2008; Cuevas and Paraboni, 2008) or problems related to coreference and anaphora resolution such as anaphoricity classification (Collovini and Vieira, 2006a,b). To the best of our knowledge, the only available corpus annotated with coreferential data is the Summ-It corpus (Collovini et al., 2007) and at least one work focus on coreference resolution (Souza et al., 2008).

One problem with the Summ-It corpus is that it imposes limits on current supervised machine learning approaches (Ng and Cardie, 2002b; Ng, 2005; Denis and Baldridge, 2007; Bengtson and Roth, 2008; Yang et al., 2008; Haghighi and Klein, 2009) because these systems require a large quantity of data for training. Summ-It contains around 17,125 tokens and contains roughly 700 coreferent referring expressions distributed in 50 newswire texts. It is not as large as several corpora used to support current machine learning approaches for coreference resolution developed for other languages such as English (MUC[1] and ACE [2]), Spanish (AnCora (Recasens and Martí, 2009)), Dutch (KNACK-2002 (Hoste and Pauw, 2006)) and others, that contain more texts and tokens, and, consequently, more coreferential referring expressions. In the present research, one approach that does not rely on manually annotated data is proposed due to the lack of resources for developing coreference resolution systems for Portuguese. The idea follows the rationale described in Postolache et al. (2006) in which the authors transfer coreference chains from the English side to the Romanian side of a manually corrected parallel corpus through word alignment.

Another important motivation for this study is that coreference resolution is

[1] `http://www.itl.nist.gov/iaui/894.02/related_projects/muc/muc_data/muc_data_index.html`

[2] `http://projects.ldc.upenn.edu/ace/data/`

an interesting problem in itself and presents great complexity. The task plays an important role in NLP and it is a key task in several NLP applications as well as an important problem in natural language understanding. Several applications can benefit from a coreference resolution system. Some of them are: information extraction, machine translation, automatic summarisation, cross-document entity coreference, and question answering.

The importance of coreference resolution as a sub-task of other tasks is evidenced by the great number of works in the area for English and other languages (Soon et al., 2001; Ng and Cardie, 2002b; Yang et al., 2003; Luo et al., 2004; Luo, 2007; Bengtson and Roth, 2008; Denis and Baldridge, 2008; Versley et al., 2008; Yang et al., 2008; Ng, 2010; Recasens and Hovy, 2009).

1.2 Objectives

The aim of this research is to develop a system that automatically extracts coreference chains for texts in Portuguese without having to resort to Portuguese corpora manually annotated with coreferential data. In order to achieve this goal, it is necessary to implement a method for automatically obtaining data for training a supervised machine learning coreference resolver for Portuguese.

In this work, the training data is acquired by using an English-Portuguese parallel corpus over which the coreference chains annotated in the English part of the corpus are projected to the Portuguese part of the corpus in a similar way as the one proposed by Postolache et al. (2006) for projecting coreference chains from English to Romanian. One last requirement is that the system must be able to deal with coreference for noun phrases including definite and indefinite noun phrases and proper names. Pronoun resolution will not be addressed.

1.3 Thesis Outline

The present text is organised as follows. In chapter 2 it is defined the concept of coreference resolution and related concepts. Furthermore, in this chapter it is also delineated the relationship between coreference and anaphora with textual cohesion. Chapter 3 presents some of the state-of-the-art of coreference resolution for English and Portuguese and relevant related work. Chapter 4 introduces

the methodology developed for the present work. In chapter 5, the coreference resolution system for the English language adopted in the methodology is described. Chapter 6 explains the sentence alignment and the word alignment processes. Chapter 7 defines and explains the coreference resolution system for Portuguese implemented as part of the methodology developed in this study. In chapter 8 the results obtained with the methodology are presented and discussed. Chapter 9 presents the conclusions and lists the future work of this research.

Chapter 2

Linguistic Concepts

In this chapter, coreference resolution is explained in more detail as well as the correlated problem of anaphora resolution. Also, cohesion and other related concepts are introduced. Furthermore, it is explained how anaphora is related to cohesion, how it contributes to the cohesion of a text and the types of anaphora related to the present work are defined. Finally, a differentiation between anaphora and coreference is delineated.

2.1 Reference resolution instantiations

2.1.1 Anaphora resolution

For defining the concept of anaphora resolution it is important to first define what is anaphora and some related concepts such as anaphor and antecedent. Halliday and Hasan (1976, page 14) define *anaphora* as the linguistic phenomenon of cohesion "pointing back to some previous item". Mitkov (2002) says that the word or phrase is the linguistic item that points back and calls it *anaphor*. From now on, *anaphora* is regarded as a reference to an entity introduced previously in the discourse as defined by Jurafsky and Martin (2009). Another important concept is the concept of antecedent. *Antecedent* is the entity to which the anaphor points back, that is, the entity to which the anaphor refers. Next sentences present examples[1] of anaphora:

(2.1) Wash and core *six cooking apples*. Put *them* into a fireproof dish.

[1]Taken from Halliday and Hasan (1976, page 2)

Example (2.1) presents an anaphoric relation between two linguistic expressions: *six cooking apples* and *them*. In this sentence, the personal pronoun *them* points back to the noun phrase *six cooking apples*. Without the latter, the former cannot be interpreted.

Having defined anaphora it is possible to define what *anaphora resolution* is: the process of finding the antecedent of an anaphor (Mitkov, 2002). The process could be illustrated as follows. It takes a set of sentences containing nominal and pronominal expressions as input[2]:

(2.2) $[\text{Bob}]_1$ opened up [a new dealership]$_2$ last week. $[\text{John}]_3$ took a look at $[\text{the Fords}]_4$ in $[[\text{his}]_6\ [\text{lot}]_7]_5$. $[\text{He}]_8$ ended up buying $[\text{one}]_9$.

In sentence (2.2) there are nine expressions enclosed with brackets. Eight of which could be regarded as expressions that could be used to refer (all but expression 2) and one is an indefinite noun phrase (an expression that introduces an entity into the discourse and has an indefinite determiner). At the end of the anaphora resolution process, an anaphora resolver outputs pairs of expressions in which each member of the pair is the anaphor and its antecedent. The output of processing the sentences in example (2.2) is presented in figure 2.1 (where the first expression of the pair is the antecedent, and the second is the anaphor).

$$\Big\{(\text{Bob}, \text{his}), (\text{a new dealership}, \text{the Fords}), (\text{John}, \text{He}), (\text{the Fords}, \text{one})\Big\}$$

Figure 2.1: Example of anaphoric pairs.

2.1.2 Coreference resolution

Natural language expressions that are used to actually perform a reference are called referring expressions. A *referring expression* is either a definite noun phrase (a noun phrase whose determiner is a definite article, or a definite determinant such as 'this', 'these', 'those', and so on), or a proper noun, or a pronoun (Mitkov, 2002). Linguistic expressions refer to the extralinguistic entity either by evoking it

[2]Taken from Jurafsky and Martin (2009, page 742)

$$\Big\{\langle \text{Bob}, \text{his}\rangle, \langle \text{a new dealership}, \text{The Fords}, \text{lot}, \text{one}\rangle, \langle \text{John}, \text{he}\rangle\Big\}$$

Figure 2.2: Example of coreference chains.

(in the case of the proper names, indefinite noun phrases and others) or accessing it (definite noun phrases, pronouns and proper nouns).

Two referring expressions that refer to the same entity are said to be *coreferent* or *coreferential*. The set of all coreferent expressions of a given entity is the *coreference chain* of that entity. Thus, it is possible to define *coreference resolution* as the task of finding all referring expressions in a text and clustering them into coreference chains.

In figure 2.2 it is presented what should be the output of a coreference resolution system given the sentences in example (2.2) as input.

2.2 Anaphora Varieties

The type of anaphora varies according to the kind of expression that constitutes the anaphor. If the anaphor is a pronoun, then it is said the anaphora is a pronominal anaphora. It occurs with personal pronouns, possessive pronouns, reflexive pronouns, demonstrative pronouns and relative pronouns. This definition of anaphora follows the one described by Mitkov (2002).

Definite noun phrases and proper names can also be anaphoric. Mitkov (2002) calls this anaphora variety *lexical noun phrase anaphora*. Mitkov also says that the definite noun phrases do not just refer but also add information about the antecedent. In this kind of anaphora, usually the anaphor has some kind of semantic relation with the antecedent and, because of that, it is said to increase the cohesiveness of the text. The pointing back can be realized through:

- the repetition of the head of the antecedent, just like in example[3] (2.3);
- the use of a synonym of the antecedent's head as in example[4] (2.4);

[3]Taken from Koch (2002)
[4]Taken from Vieira et al. (2008)

- the use of a hypernym, generalization, or superclass, as seen in example[5] (2.5);
- the use of a hyponym, specification or subclass, presented in example[6] (2.6);
- the match of the whole or part of a proper name (example[7] (2.7)).

(2.3) *O presidente* viajou para o exterior. *O presidente* levou consigo uma grande comitiva.

The president travelled abroad. The president took a big entourage with him.

(2.4) Isso quer dizer que *os camundongos* transgênicos reduziram a gordura de seu corpo. *Os ratos* estudados...

This means that the transgenic mice had their body fat reduced. The rats studied...

(2.5) As mudanças nas populações de *pinguins* também serviram como indicativo do problema climático. *Os animais* usavam geleiras para se abrigar e procriar.

The penguins population changes also indicated a climatic problem. The animals used to shelter and procreate into the glaciers.

(2.6) Sem saber, *o aracnídeo* está providenciando o suporte perfeito para o casulo da parasita. Na noite em que a teia fica pronta, a larva irrompe do corpo d*a aranha*, matando-a.

Without knowing, the arachnid is providing the perfect suport for the parasite's cocoon. In the night the web is ready, the larva breaks out of the spiders' body, killing it.

(2.7) Roy Keane has warned *Manchester United* he may snub their pay deal. *United*'s skipper is even hinting...

As the linguistic expressions could be used to either evoke or access the extralinguistic entities, they assume different status in the discourse. This way,

[5]Taken from Vieira et al. (2008)
[6]Taken from the Summi-it corpus (Collovini et al., 2007)
[7]Taken from Mitkov (2002, page 10)

anaphora can be also classified according to their status. Vieira (1998) and then Collovini and Vieira (2006a) propose this classification. The expressions can be new or old in the discourse. When an expression is *new* in the discourse, its interpretation does not rely on any previous expression. It refers to the entity for the first time. They serve as antecedents to discourse-old expressions. When an expression is *discourse-old* it accesses an entity previously evoked in the discourse. The discourse-old anaphoras are of three types: direct anaphora, indirect anaphora and associative anaphora.

The direct anaphora establishes an identity of reference relationship with the expression to which it points back. Besides that, it has the same head as the antecedent. In spite of being based in a different perspective, this concept is equivalent to the concept of lexical noun phrase anaphora realized through repetition described by Mitkov (2002) and presented above. The same example can be used to illustrate this kind of anaphora (example (2.3)).

The indirect anaphora also presents an identity of reference with its antecedent. However, the heads of both the anaphor and the antecedent are not the same. In order to decode the meaning of an indirect anaphor, the reader of a text has to make use of semantic and pragmatic knowledge (Vieira et al., 2008). The indirect anaphora can be realized through the use of a synonym, a hypernym, a hyponym or through proper names. Here, the concept is equivalent to the remaining realizations of lexical noun phrase anaphora (synonym, hypernym, hyponym and proper nouns).

Collovini and Vieira (2006a) call *associative anaphora* any expression that is new in the discourse but that needs a discourse-old expression in order to be interpreted. The authors remark that even though it could be a new referent in the discourse, its meaning is strongly coupled to a previous expression. This new expression "anchors" its meaning in the old expression. The relationships between these kind of anaphors and their antecedents feature part-of, set membership and subset-set relations. It requires semantic and "world knowledge" in order to be interpreted. Mitkov (2002), calls the associative anaphora of indirect anaphora.

Having defined and seen these varieties of anaphoras one can conclude that the evocations and accesses to the extralinguistic entities in the discourse vary in a great extent. It shows how important is the anaphora phenomenon: it has an important role in the construction and understanding of the text. Consequently,

it shows that the anaphora resolution and, in a second moment, coreference resolution are crucial and play the very same roles in NLP. Besides that, anaphora is related to another important notion that is described in the next section: the notion of cohesion.

2.3 Cohesion and Anaphora

One of the problems in NLP systems is the problem of maintaing and understanding the cohesion of a discourse. The property of cohesion is what allows a text to be understood globally and as an unity. For Halliday and Hasan (1976), a text is not a grammatical unit as a sentence, for example. A text is a semantic unit of meaning. According to them, "a text does not consist of sentences; it is realized by, or encoded in sentences.". Thus, a text is a discourse realized by sentences. For the authors, what distinguishes a text from something that is not a text, are certain linguistic features that contribute to the unity of the text. They call these features the texture of the text.

Cohesion is defined by Halliday and Hasan (1976) as a semantic relation between an element in the text and some other element required for its interpretation. Hence, the cohesion occurs when the interpretation of an element in the discourse depends on the interpretation of another element. There are five types of cohesion: reference, substitution, ellipsis, conjunction and lexical. In this work the focus is on the reference and lexical classes.

The elements that belong to the class of reference cohesion are language items that cannot be semantically interpreted by themselves: they need other elements in the discourse for being decoded (Koch, 2002). There are two types of reference: situational (exophora) and textual (endophora).

The reference is situational or exophoric when the referent is linked to an element in the context, outside the text. The reference is textual or endophoric when the referent is linked to an item in the text. If the reference is endophoric it may appear before or after the cohesive item. If it comes before the cohesive item, it configures an anaphora. If it comes after, it configures a cataphora.

The reference class is subdivided into three subclasses: personal, demonstrative and comparative. The personal reference is established by

personal pronouns and possessive pronouns. In the following example[8], sentence 2.8, without the expression *Drogas baseadas no UCP-3* (drugs based on the UCP-3), *Elas* (they) could not be interpreted.

(2.8) *Drogas baseadas no UCP-3* teriam pouco em comum com os moderadores de apetite usados hoje. "*Elas* funcionariam do outro lado da equação", disse Clapham.

Drugs based on the UCP-3 would have little in common with the appetite suppressants used nowadays. "*They* would work in the other side of the equation", said Clapham.

The former presupposes the latter and cannot be interpreted without it. This semantic relation of the anaphoric expression *Elas* with its antecedent links the two sentences. When an expression refers back to a previous expression configuring an anaphoric relation, it gives cohesion to the two sentences allowing us to interpret both sentences (the text) as a whole (Halliday and Hasan, 1976). This is why it is said that the anaphor requires the antecedent in order to be correctly decoded. To a certain degree the pronominal anaphora concept is related to the personal reference. The difference is that the concept of pronominal anaphora encompasses all the pronouns whereas the definition of personal reference cohesion is restricted to personal and possessive pronouns.

Another interesting type of cohesion for the current work is the lexical cohesion. This kind of cohesion is subdivided in two subclasses: reiteration and collocation. The focus here is on the former. The reiteration occurs when there is repetition of the same lexical item or the repetition of synonyms, hiperonyms, hyponyms or generic names. They correspond to the examples (2.3), (2.4), (2.5) and (2.6), introduced in section 2.2.

(2.9) Todos ouviram um rumor de asas. Olharam para o alto e viram *a coisa* se aproximando.

Everyone has heard a sound of wings flapping. They looked above and saw the thing getting closer.

Sentence 2.3 presents an example of reiteration by repetition of the same lexical item, in this case *presidente* (president). Sentence 2.4 shows an example

[8]Taken from the Summ-it corpus (Collovini et al., 2007)

of reiteration by repetition of a synonym: *Os ratos* (the rats) reiterates *os camundongos* (the mice). By its turn, sentence 2.5 presents an example of reiteration by repetition of a more generic name: *Os animais* (animals) is a superclass or hyperonym of *pinguins* (penguins). The use of generic names in the reiteration is exemplified in example[9] 2.9. In this example, the definite noun phrase *a coisa* (the thing) refers back not to a single linguistic expression but to the something that is inferred from the situation introduced in the first sentence.

Koch (2002) presents studies of cohesion for Portuguese and divides cohesion into two subclasses: referential cohesion and sequential cohesion. The referential cohesion makes use of reiteration mechanisms like synonymy, meronymy, hypernym and generic names, the same way the lexical cohesion of Halliday and Hasan does. It is important to notice, that for Koch, the lexical cohesion, more specifically, the reiteration type, has the same cohesive function the pronouns have in the reference cohesion. They maintain the reference identity of the antecedent. Therefore, for Koch, the lexical cohesion is not an independent functional mechanism, as it is for Halliday and Hasan.

The sequential cohesion is related to the idea of textual progression. There are elements in the text that, when put together, give sequentiality and continuity to the main idea of the text. When the text is cohesive, the parts are interdependent and important to the general comprehension of the text. This phenomenon is called textual progression. This way, the sequential cohesion is used to perform the thematic maintenance and the chaining in the text. The chaining allows the establishment of semantic relations between the clauses, sentences or textual sequences. In this sense, the most frequently mechanisms used in coreference, for example, are the repetition and substitution, both of which realized in the lexical noun phrase anaphora described in section 2.2.

2.4 Anaphora and Coreference

Anaphora and coreference are related concepts that many times lead to wrong interpretations and assumptions. In this section the differences between anaphora and coreference are presented and described.

The only requisite for two linguistic expressions being coreferent is that they

[9]Taken from Koch (2002)

have the same referent in the real world or in the discourse. That is, given two expressions a and b, if $referent(a) = referent(b)$, then they are coreferent. In an anaphoric relation, when both the anaphor and the antecedent refer to the same entity it is said that both are coreferent. We can see that in example[10] 2.10. Besides presenting an anaphoric relation (*He* points back to *John*), the anaphor and the antecedent are also coreferent (since both expressions refer to the person Bill).

(2.10) John hid *Bill*'s car keys. *He* was drunk.

Anaphora is a reference to an entity introduced previously in the discourse. This way, it is important to notice that not every anaphoric relation is coreferent.

2.4.1 Computational Complexity

Anaphora and coreference resolution also have different computational complexities. The task of coreference resolution (and also the task of anaphora resolution) is considered one of the most difficult tasks in Artificial Intelligence (Ng, 2002; Denis, 2007). The computational complexity of the coreference resolution is exponential in the number of mentions whereas the complexity of anaphora resolution is quadratic in the number of mentions. For coreference resolution, the search space is the "the set of all mutually disjoint subsets that can be created over the set of mentions" (Denis, 2007, page 5). Luo et al. (2004) reports that the problem of coreference resolution is equivalent to the set partitioning problem (an NP-Complete problem) and that its search space can be modelled using a Bell-Tree in which the number of leaves is the number of possible coreference outcomes. Luo et al. (2004) illustrates the exponential complexity of coreference resolution showing that a text with only 20 mentions can have approximately 5.2×10^3 possible coreference outcomes (also the size of the search space).

In this chapter coreference resolution and anaphora resolution have been defined and explained. In the next chapter, related work in the area of reference resolution is commented and briefly explained.

[10] Taken from Jurafsky and Martin (2009, page 723)

Chapter 3

Related Work

This chapter introduces some of the approaches to the task of coreference resolution. At the end of the chapter the research done for the Portuguese language is presented.

3.1 General Algorithm for Reference Resolution

Most of the works in anaphora or coreference resolution follow similar steps in order to perform the reference resolution. In this section these steps are presented and briefly described. All approaches, taking into consideration their differences, roughly follow this general algorithm. The algorithm was adapted from Ng (2002) and Denis (2007). It receives a text as input and it returns a list of anaphoric pairs, in the case of anaphora resolution (as described in section 2.1.1), or it yields a list of coreference chains, in the case of coreference resolution (as described in section 2.1.2).

1. **Referring expressions identification.** The first step of any reference resolution system is to identify the discourse entities in the text. In this step, the nominal or pronominal expressions are extracted for being processed in the next steps.

2. **Characterization of the expressions.** In this step relevant information that might be useful for linking one referring expression to another

expression in the text are extracted. Which information and how it is extracted varies according to the different approaches. The extraction could be obtained automatically from preprocessing modules, or using corpora with gold standard information.

3. **Anaphoricity determination.** At this point, some systems determine which expressions are anaphoric and which are not. If the expression is not anaphoric it does not have any antecedent. Not all systems perform this step. Those systems that do not determine anaphoricity here, assume that all expressions selected in step 1 are potentially anaphoric.

4. **Generation of candidate antecedents.** Until here, all the steps had scope over all the document. From this point, the processing is over the anaphoric expressions computed until step 3. The goal of this step is to generate all the possible antecedents for each potential anaphor. Some systems assume that every expression previous to the expression currently being analyzed are possible antecedents.

5. **Filtering of antecedents.** The goal of this step is to filter unlikely antecedents from the list of candidate antecedents produced in step 4 according to different linguistic principles and constraints. As well as step 3, this step is not executed by all anaphora and coreference resolution approaches.

6. **Scoring/Ranking of antecedents.** In this step the candidates are ordered according to criteria established by each algorithm. This ordering can be seen as a ranking in which the most likely antecedent is always the first in the list. Each expression in the candidates' list is given a numeric value that reflects the likelihood of having an anaphoric or coreferential relation with a potential anaphor. This step is not performed by all the systems.

7. **Searching/Clustering.** In the final step, one expression is chosen as the antecedent of the anaphor being processed. If the list of candidates is empty, then no antecedent is selected. If step 6 is performed, then the selection consists of picking the expression ranked as the first in the candidate list. If not, the candidate list is searched for the "best" candidate

following some order defined by the approach. In this step, some coreference resolution systems, partition the set of referring expressions of the text through transitive closure of the anaphoric pairs.

3.2 Coreference Resolution

In the last decade, the approaches to coreference resolution moved from systems based on rules or heuristics to systems based on machine learning. Machine learning systems are more robust, portable and easier to implement than the first approaches and usually report better results than the rule-based ones. The construction of several corpora annotated with coreferential or anaphoric information allowed this change.

For the English language, the Message Understanding Conference[1] (MUC) and the Automatic Content Extraction[2] (ACE), competitions promoted to develop resources for different Information Extraction tasks, led to the development of widely used corpora with coreference information. At the same time, under the scope of these competitions, the research on reference resolution was directed to the more general problem of coreference resolution.

As have been seen in the previous sections, reference resolution requires considerable knowledge to be successfully performed. Knowledge about different linguistic levels such as morphology, syntax, semantics, discourse, pragmatics and even general world-knowledge are useful when performing reference resolution. Machine learning algorithms allow the construction of robust systems that can automate the acquisition of the knowledge required from annotated corpora by learning patterns from it.

In these approaches, the coreference resolution is recast as a binary classification problem followed by a clustering step. The binary classification consists of deciding if pairs of mentions are coreferential or not. After the classification phase, a clustering algorithm merges the pairs into coreference chains. These approaches can be characterized in terms of (i) the machine learning algorithm used to induce the model; (ii) the knowledge sources employed to develop the features used to induce the model; (iii) the method used to create

[1] `http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html`
[2] `http://www.itl.nist.gov/iad/mig//tests/ace/`

the data for training the model; and (iv) the clustering algorithm employed to form the coreference chains.

3.2.1 Single-Mention Pairwise Machine Learning Approach

The single-mention pairwise approach is divided into two steps: one step in which pairs of expressions are classified into coreferent or not coreferent classes and one clustering step in which the coreferent expressions clustered into chains. Soon et al. (2001) is a representative example of this approach to coreference resolution and it is one of the baselines for systems being developed nowadays. In this section it is described the approach proposed by Soon et al. (2001) since most of the work done after this approach always refer to or is based on this work due to its good performance (62.6% for MUC-6 and 60.4% for MUC-7) and robustness.

Model

The system was designed to resolve general noun phrases including personal pronouns, reflexive pronouns and possessive pronouns in unrestricted texts (i.e. texts from any domain). Preprocessing modules provide tokenization, part-of-speech tagging, chunking and named entity recognition.

Following the outline of machine learning approaches described above, Soon et al. (2001) divided their approach in two steps: a pairwise classification phase and a clustering phase. A classifier is induced using the C4.5 algorithm (Quinlan, 1993) based on a sampling of training instances created from coreference annotated corpora. Having the pairs classified, the clustering step merges coreferent pairs into coreference chains.

Feature set

The authors propose 12 features to induce the classifier and determine if two mentions are coreferent or not in the classification phase. The features were designed having in mind their use in any domain. The feature set uses knowledge derived from morphology, syntax, semantics, and lexical comparison between mentions. The features are extracted based on two noun phrases, i (the potential antecedent) and j (the anaphor). The features are:

- **Distance (DIST):** captures the distance in sentences between noun phrase i and j. If i and j are in the same sentence the distance is 0.

- **i-Pronoun (I_PRONOUN):** if the i expression (the antecedent) is a pronoun, this feature is true. Otherwise it is false.

- **j-Pronoun (J_PRONOUN):** if the j expression (the anaphor) is a pronoun, this feature is true. Otherwise it is false.

- **String Match (STR_MATCH):** holds the result of the string comparison between i and j. Prior to the comparison, articles and demonstrative pronouns are removed. Possible values are true or false.

- **Definite Noun Phrase (DEF_NP):** if j is a definite noun phrase this feature is true, else it is false. In Soon et al. (2001) a definite noun phrase is a noun phrase that starts with the word *the*.

- **Demonstrative Noun Phrase (DEM_NP):** if j is a demonstrative noun phrase the feature holds true, else it is false. A demonstrative noun phrase for the authors is a noun phrase that starts with the words *this*, *that*, *these* or *those*.

- **Number Agreement (NUMBER):** true if both expressions agree in number (i.e. both singular or both plural). Otherwise false.

- **Semantic Class Agreement (SEMCLASS):** the authors defined ten semantic classes *female*, *male*, *person*, *organisation*, *location*, *date*, *time*, *money*, *percent*, and *object* arranged in a simple ISA hierarchy. Thus, *female* and *male* are a kind of *person*, and *organisation*, *location*, *date*, *time*, *percent*, *person*, and *money* are subclasses of *object*. Each semantic class is mapped to a synset in WordNet (Miller, 1995). The semantic classes of i and j are in agreement if the head of one is in a parent class of the other or if both heads are in the same class. If one of the preceding conditions holds, the feature value is true. If the head of the noun phrase does not map to any of the defined classes the value of the feature is unknown. Else, if the semantic classes do not match, the feature value is false.

- **Gender Agreement (GENDER):** this feature holds true if both expressions agree in gender, false if they do not agree and unknown if

the gender of at least one expression cannot be determined. The system uses the semantic class to determine the gender of the noun phrases when applicable.

- **Both-Proper-Names (PROPER_NAME):** if both expressions are proper names, this feature receives true, else it receives false.

- **Alias (ALIAS):** this feature holds true if i is an alias of j or vice-versa. Otherwise the feature is false. This feature captures the named entities that refer to the same entities. For example, an acronym (*IBM/International Business Machines*), the last name of a person's name (*Bent Simpson/Mr. Simpson*), and others.

- **Appositive (APPOSITIVE):** if j is in apposition with i, then the feature is true. Otherwise it is false.

Creation of training instances

Training instances are created based on pairs of mentions in which each instance is represented by the set of features described above. The positive training instances are formed between an noun phrase and its **closest** preceding noun phrase in the same coreference chain. That is, given a chain of coreferent expressions $C = \{a, b, c, d\}$ from the manually annotated corpus, positives instances are formed using adjacent expressions in the chain. Thus a list of positive pairs T derived from C would be $\{(a, b), (b, c), (c, d)\}$.

This pairing method is called *non-transitive*. Earlier studies such as McCarthy and Lehnert (1995); Aone and Bennett (1995) employ the *transitive* pairing method for positive instances in which a noun phrase is paired with all its coreferent antecedents. The *non-transitive* method is an attempt of Soon et al. (2001) for reducing training time and data noise since the *transitive* method generates a great number of instances. Given a noun phrase j and a potential antecedent i, the negative instances are generated forming a pair between j and all the expressions not coreferent with j between i and j.

Clustering mechanism

For generating the coreference chains, it is assumed that every noun phrase j in the text is a possible anaphor and every noun phrase preceding j is a potential antecedent. The resolution mechanism work as follows: starting from the second noun phrase of the text, each noun phrase j until the end of the document is paired with each of its preceding noun phrases. For each such pairs, a feature vector is generated and given to the induced classifier. The classifier, by its turn, returns whether the pair is coreferent or not and the closest antecedent i is assigned to the same cluster as the noun phrase j. This process goes on until a pair is classified as coreferent or the beginning of the text is reached. This clustering mechanism is known as *Closest-First* clustering.

3.2.2 Limitations and Enhancements to the Pairwise Approach

The pairwise classification model exhibit some inherent problems. One problem is that each antecedent candidate is treated as a separate, independent event and fails to capture the dependencies between the different candidates (Yang et al., 2003; Denis and Baldridge, 2007). A better approach would be to rank the best antecedent in function of some criteria to decide which one is the best candidate (as in step 6 of the general algorithm). Another problem is that different noun phrases require different approaches to reference resolution that a single monolithic classification model cannot handle adequately.

There is also the so-called decision locality problem. The single pairwise classification model does not take into account the dependencies between coreference decisions during the training and during the application of the model (Denis and Baldridge, 2007). During training, pairs of mentions are classified as coreferent or not and these classification decisions do not use information from the previous decisions. Likewise, during the application of the model the clustering decisions are also made without any information regarding previous decisions. The clustering scheme poses an important problem when a situation like the following holds. When mention $a =_a b$ and $b =_a c$, where $=_a$ means "anaphoric", the clustering algorithm would likely merge all three mentions into one cluster even though $a \neq_a c$. In the single pairwise model there is no synchrony

between the classification and clustering steps, they are optimized independent from each other (Denis and Baldridge, 2007). This way, a large improvement to the classifier may not reflect any improvement in the final coreference chains.

Different solutions were proposed to cope with these problems. Some of them are models that are new approaches to the coreference resolution problem. Others, are enhancements or modifications to the pairwise approach that aim to alleviate its problems. The different approaches are listed and briefly described in section 3.2.3. The modifications are related to:

- the machine learning algorithm used.
- the knowledge sources applied to the feature set as well as the features themselves used in the classification step;
- the sampling method for generating training instances for the learning algorithm;
- the clustering method used for merging the coreferent pairs into coreference chains;

In this section the enhancements are presented and briefly described.

Machine Learning algorithm

Several works (McCarthy and Lehnert, 1995; Soon et al., 2001; Strube et al., 2002; Ng and Cardie, 2002b,a; Yang et al., 2003; Ng, 2004, 2007b) make use of decision trees (Russell and Norvig, 2003) to induce a classifier. This is the most common supervised machine learning algorithm used in the single pairwise model for coreference resolution. One of the reasons for this is the fact that decision trees can be visualised, are easy to understand, and are one of the most well known supervised machine learning algorithms. Some other works (Kehler, 1997; Ponzetto and Strube, 2006b,a; Denis and Baldridge, 2007), exploit the use of Maximum Entropy Models (Berger et al., 1996) for learning the coreference decisions. Besides decision trees and Maximum Entropy Models, Support Vector Machines (SVM) also were employed to learn the classifier (Ng, 2007a; Stoyanov et al., 2009b).

It is unclear for the author of this work whether one machine learning algorithm is superior over the other. Stoyanov et al. (2009b) experiments with

both decision trees and SVM and reports that the results are comparable. The choice of the learning algorithm is closely related to the model applied for coreference resolution. Therefore, a different approach to the problem may require a different machine learning algorithm.

Knowledge sources and features set

Coreference resolution is a difficult task that depends on several knowledge resources. However, Soon et al. (2001) employs a small set of 12 features extracted from limited resources for determining whether a given pair of mentions is coreferent or not. More recent studies (Ng and Cardie, 2002b; Ponzetto and Strube, 2006a,b; Ng, 2007b,a; Yang and Su, 2007; Bengtson and Roth, 2008) explore the expansion of the feature set to promote an improvement in the performance of the resolution.

A rather simple and cheap feature was introduced by Strube et al. (2002): the use of minimum edit distance between two mentions to determine whether the two have lexical similarities (configuring a case of anaphora by lexical repetition).

Ng and Cardie (2002b) expand the set to 53 features containing different kinds of information: lexical, grammatical (including a variety of linguistic constraints and preferences), semantic and knowledge-based, positional and others. The author analyzed the performance of the system according to the features employed and concluded that not all 53 features contribute to the resolution process, and that, in fact, using all the 53 features degrades the system's performance (mainly on common nouns resolution). After testing combinations of the initially proposed feature set, Ng and Cardie (2002b) came up with a hand selected set of 22 to 26 features (the feature set vary according to the corpus used). The study reports better results than the best performing systems in 2002 for the MUC-6 and MUC-7 corpora (70.4 and 63.4 respectively).

Although the WordNet has been widely used for coreference resolution, it presents coverage limitation (the coverage for common nouns is limited) and other problems (refer to Markert et al. (2003); Ponzetto and Strube (2006a) for more details). Ponzetto and Strube (2006b,a); Ng (2007b,a); Yang and Su (2007); Bengtson and Roth (2008) explore the use of deeper semantic information in the task of coreference resolution. Ponzetto and Strube (2006b) employs semantic role labelling (Gildea and Jurafsky, 2002; Jurafsky and Martin, 2009) to add two

new features to the Soon et al. (2001)'s feature set regarding the possible semantic roles of the antecedent and the anaphor. Ponzetto and Strube (2006a) go further and employ semantic features extracted from two different sources, the WordNet and the Wikipedia with their semantic role labelling features. Combining all semantic features proposed, Ponzetto and Strube (2006a) reported improvement (69.5% of F-Measure) over the baseline (Soon et al., 2001) using the ACE 2003 corpus[3].

Ng (2007a) builds a supervised semantic class classifier of noun phrases for applying to the coreference resolution task. The author was intrigued with the fact that no semantic features were used in the final decision tree trained in Soon et al. (2001). The study proposes using semantic class agreement as a feature processed before the coreference resolution task. Results report that using a semantic class classifier obtained through supervised machine learning is better than following Soon et al. (2001)'s semantic class method (briefly described in section 3.2.1) on ACE corpus.

Yang and Su (2007) automatically extracts effective patterns for coreference resolution from Wikipedia. Examples of such patterns are "X such as Y" (*is-a* relation), or "X and other Y" (*other*-relation). The results show that when applied to noun phrases that contain proper names it is noticed an improvement on the performance of the resolution for the ACE-2 corpus [4]. However, for noun phrases whose head is a common noun no improvement is observed.

Bengtson and Roth (2008) observed that most of the works in coreference resolution propose new models rather than concentrate on useful features for determining coreference. In view of this, the authors propose a knowledge-rich feature set formed by eight categories: mention types (indicate whether the mention is a proper name, a common noun, or a pronoun); string relation (string comparison functions that indicate whether two strings share some property, such as one substring of the other); semantic (gender match, number match, WordNet match, and others); relative location (distance measures between two mentions including apposition relation); learned features (modifier names and anaphoricity); aligned modifiers (determine the relationship of any pair of modifiers that share a hypernym); memorization features (learn which pairs of nouns tend to be used to mention the same entity); and predicted entity type

[3] `http://www.itl.nist.gov/iad/mig//tests/ace/2003/`

[4] `http://www.itl.nist.gov/iad/mig//tests/ace/2002/`

(checks if the two mentions share the same entity type, for instance). The results report the best performance for coreference resolution in the ACE 2004 English training data[5] (B^3 F-Measure of 78.24).

These studies show that reliable features set is a key factor for good coreference resolution. Besides, relying on knowledge-rich features contribute for an increase in the performance of the system.

Sampling training method

Different sampling methods lead to different instance sets with more or less data noise, depending on the features and corpora available. McCarthy and Lehnert (1995) and Aone and Bennett (1995) used the *transitive* method for pairing positive samples. Soon et al. (2001) built its *non-transitive* to overcome the data noise and data sparsity produced by the transitive method. In the non-transitive method, the closest antecedent is chosen to form a pair with the anaphor.

Ng and Cardie (2002b) proposes yet another method for generating positive instances for training. Rather than forming a pair between the anaphor and its closest antecedent, a pair is formed pairing the anaphor and its **most confident** antecedent. For each **non-pronominal** noun phrase, it is assumed that the most confident antecedent is the closest **non-pronominal** preceding antecedent. For **pronouns**, the most confident antecedent is its closest preceding antecedent. Negative instances are created as in Soon et al. (2001) (described in section 3.2.1). The results report better sampling when using the method this method than non-transitive method prosed by Soon et al. (2001).

Ng and Cardie (2002a) cites two intrinsic coreference properties that pose a problem to the pairwise classification followed by clusterization approach (i.e. single-mention pairwise model) to coreference resolution. The first is that coreference is a rare relation, that is, many coreference corpora contain a minuscule number of positive instances when compared to the negative ones. The MUC-6 and MUC-7 corpora contain only 2% positive instances (Ng and Cardie, 2002a), for instance. The second is that different noun phrases require different approaches for their resolution. Pronouns may be dependent only on its closest antecedent, and proper names may rely only on string matching or aliasing techniques, for example. This way, creating positive instances generically, for all

[5] `http://www.itl.nist.gov/iad/mig//tests/ace/2005/`

types of noun phrases, may generate pairs "hard" to classify.

Building on top of these limitations of the model, Ng and Cardie (2002a) propose two instance selection methods. One for negative instance selection and one for positive instance selection (in the general algorithm this would be the step 5 of filtering of candidates). The negative instance selection algorithm, retains only the negative instances that are in between the mention j and its **farthest** potential antecedent i. Any negative instance before i are discarded (as opposed to Soon et al. (2001) method which considers **all** non-coreferent noun phrases preceding j). For positive instance selection, Ng and Cardie (2002a, page 56) presents "a corpus-based method for implicit selection of positive instances" which is a fully automated version of the selection algorithm described by Harabagiu et al. (2001). This positive instance selection tries to avoid the inclusion of "hard" training instances. When combining these two filtering algorithms, Ng and Cardie (2002a) reports an improvement on the system performance comparing to their baseline (about 17 F-Measure points in MUC score for MUC-6 dataset and 16 for MUC-7 dataset).

Following this rationale, Uryupina (2004) experiments with a sampling method in which each different type of noun phrase receives a different treatment. There are different sampling methods based on linguistic evidences for pronouns, proper names, definite noun phrases and the remaining noun phrases. The results indicate improvements both in the speed and in the performance of the resolver.

These studies show the methods used for sampling training instances do contribute in great part for the resolution performance. Therefore, considering different sampling methods than the ones developed by Soon et al. (2001) is advisable.

Clustering method

Besides a new feature set (see section 3.2.2), Ng and Cardie (2002b) also propose a new clustering mechanism. The idea behind this clustering algorithm is to do a right-to-left search for a *highly likely antecedent* (as opposed to the first coreferent noun phrase). The clustering algorithm is modified to select as the antecedent of the noun phrase j the closest noun phrase with the **highest** coreference likelihood value among all the preceding noun phrases. Additionally, all preceding noun phrases must have a confidence value above a certain threshold (usually 0.5).

Since a decision tree usually labels the pairs with a binary value, it is necessary to come up with a way of making the classifier able to return a value between 0 and 1. This value is calculated using the ratio defined in 3.1 where p is the number of positive instances and t the total number of instances in the decision tree's leaf node.

$$\frac{p+1}{t+2} \tag{3.1}$$

The method proposed by Soon et al. (2001) is known as *Closest-First* clustering and the method presented by Ng and Cardie (2002b) is known as *Best-First* clustering. McCarthy and Lehnert (1995) employs an *Aggressive-Merge* clustering in which each mention j is merged with all its preceding coreferent mentions. According to Denis and Baldridge (2007), Aggressive-Merge is likely to yield good recall while Closest-First and Best-First are likely to yield better precision. All these methods are local clustering methods.

3.2.3 Models

In this section, approaches different than the single-mention pairwise approach are listed and briefly described. They view the problem of coreference resolution from a different perspective than the pairwise model delineated in section 3.2.2.

The Competition Learning Approach

Yang et al. (2003) proposes a competition learning approach using a twin-candidate model based on the work of Connolly et al. (1997). In the twin-candidate model the training and testing instances are formed by an anaphor and two potential antecedents. A learning algorithm is then used to induce a classifier that, in its turn, is used to determine the preference between two antecedent candidates of an anaphor encountered in a new document. The candidates "compete" and the one with most wins in the comparisons is selected as the antecedent. In this approach, a great number of training and testing instances is generated and for reducing data noises and computation cost, an antecedent filter is employed (in training and testing). According to Stoyanov et al. (2009b), this is the best performing system on MUC-6 and MUC-7 datasets (71.3% and 60.2% of MUC score F-Measure respectively).

Multi-Candidate Ranking

Rather than using a single-candidate or a twin-candidate model, the multi-candidate goes further and ranks, through a log-linear model, all the antecedents of an anaphor. The antecedent with the best score is the one chosen. With multi-candidate ranking the decisions are made globally while with single-candidate and twin-candidate the decisions are made locally. As well as in the twin-candidate model, in each training instance (anaphor and respective antecedents) an antecedent must be chosen (as opposed to the single-candidate model in which an anaphor may not have an antecedent). Besides using a different model, Denis (2007) proposes different ranking models for each class of referential expressions (third person pronouns, speech pronouns, proper names, definite descriptions, and other types of phrases). This approach has been applied both to pronoun resolution and to coreference resolution.

Unsupervised Machine Learning approaches

One of the earliest unsupervised machine learning approaches to coreference resolution is the one proposed by Cardie and Wagstaff (1999). The coreference resolution is recast as a noun phrase clustering task represented by a set of eleven features very similar to the feature set used by other works by the time the study was released (McCarthy and Lehnert, 1995; Soon et al., 2001). The resolution process consists of a right-to-left single-link clustering algorithm (the same rationale of the closest-first method described in section 3.2.1) to partition the set of mentions into coreference chains. The results demonstrated to be superior to the ones obtained by McCarthy and Lehnert (1995) (a supervised machine learning approach similar to Soon et al. (2001)).

Another unsupervised machine learning approach is the one introduced by Bean and Riloff (2004) which makes use of thematic roles to improve the performance of the system (results show that pronominal anaphora resolution is improved by 15%). A more recent work (Haghighi and Klein, 2007) based on unsupervised learning presents a fully generative non-parametric Bayesian model of mentions that captures both within- and cross-document coreference with performance comparable to the state of the art (MUC score F-Measure of 70.3 on MUC-6 dataset).

Other approaches

Several other approaches different than the previous ones were proposed and evaluated. Ng (2005) presents a study in which different learning-based approaches to coreference resolution are employed to produce candidate partitions (coreference chains) of the noun phrases. After, an "automatically acquired ranking model" (Ng, 2005, page 157) (SVM-based) ranks the candidate coreference chains and chooses the best to be the final response. Results show improvement over the baseline (Soon et al., 2001) however, the methodology is rather difficult to implement (it requires the implementation of different systems). Denis and Baldridge (2007) recast the coreference resolution problem as an optimization problem, namely, an Integer Linear Programming (ILP) problem. Good results (comparable to the state of the art at the time) are reported over the ACE dataset. Some other approaches employ Conditional Random Fields (McCallum and Wellner, 2004), and graph algorithms Luo et al. (2004); Nicolae and Nicolae (2006) but all of them have inferior performance than Denis and Baldridge (2007).

3.2.4 Evaluation

There are two main types of evaluation for coreference resolution: intrinsic evaluation and extrinsic evaluation. Intrinsic evaluation consists of measuring the performance of the system against a gold standard annotated corpus. Extrinsic evaluation is the evaluation of a system by using it embedded into another system. The focus of the evaluation of the coreference resolution task has been in intrinsic evaluations rather than in extrinsic evaluations.

In intrinsic coreference resolution evaluation, the evaluation metric must consider the coreference chains produced by the systems and provide a value for measuring how well they match to the chains manually annotated in the gold standard corpus. Three metrics were developed for evaluating the performance of coreference resolution systems (among others): the MUC (Vilain et al., 1995) metric, the B^3 (Bagga and Baldwin, 1998) metric, and the CEAF (Luo, 2005) metric. All three metrics report performance in terms of *precision* and *recall* but each metric computes them in a different way. The description and notation of the metrics are based on Denis and Baldridge (2007).

MUC metric

The MUC metric is a link-based evaluation. It counts the number of links present in the response set $\mathcal{R}$ and in the "true" or "key" chains set $\mathcal{K}$ and intersect them. Recall is the ratio between the number of links that are common to $\mathcal{R}$ and $\mathcal{K}$ and the total number of links in $\mathcal{K}$. Precision is then the ratio between the number of links that are common to $\mathcal{R}$ and $\mathcal{K}$ and the total number of links in $\mathcal{R}$. Therefore, recall penalises the missing links (the links present in $\mathcal{K}$ but not in $\mathcal{R}$) whereas precision penalises the spurious links (the links present in $\mathcal{R}$ but not in $\mathcal{K}$). The definitions of precision and recall are given respectively by 3.2 and 3.3, where R is one of the chains belonging to $\mathcal{R}$ and T is one of the chains belonging to $\mathcal{K}$.

$$Precision_{MUC} = \frac{\sum_{R \in \mathcal{R} \cap T \in \mathcal{K}} |R \cap T| - 1}{\sum_{T \in \mathcal{R}} |R| - 1} \tag{3.2}$$

$$Recall_{MUC} = \frac{\sum_{R \in \mathcal{R} \cap T \in \mathcal{K}} |R \cap T| - 1}{\sum_{T \in \mathcal{K}} |T| - 1} \tag{3.3}$$

The MUC metric is the oldest of the three metrics (introduced in the MUC-6 competition) and has been being used by several studies since then. However, several studies report problems (Bagga and Baldwin, 1998; Luo, 2005; Popescu-Belis and Robba, 1998) in the MUC metric. One of the shortcomings is that the metric favors systems that produce large chains. If all mentions in a document are put in the same chain (i.e. refer to the same entity), the results would be 100% of recall, 78,9% of precision and 88,2% of F-Measure. This behaviour is explained due to the fact that the metric counts the minimum number of links required to map a chain R to a chain T. For example, given two set of chains, $R = \{\{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}\}$ and $T = \{\{e_1, e_2, e_3, e_6\}\{e_4, e_5, e_7\}\}$, one would be mapped into the other by adding only one link. One related shortcoming is the limitation of MUC metric handling singleton chains (chains composed by only one reference to an entity). Singleton chains do not present any link to be computed and MUC metric is a link-based evaluation. Because of these two problems, a worst system could obtain better results than a system considered to perform better.

B^3 metric

The B^3 metric is a mention-based evaluation proposed by Bagga and Baldwin (1998) to overcome the shortcomings of the MUC metric. Instead of computing over the links, this metric computes at the level of each mention. Let R_m be the coreference chain containing mention m and T_m be the key chain containing m. The precision is the ratio between the number of mentions common to R_m and T_m and the total number of mentions in S_m. Similarly, the recall for m is the ratio between the number of mentions common to R_m and T_m and the total number of mentions in T_m. Thus, both are defined as 3.4 and 3.5 respectively.

$$Precision_{B^3} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \frac{|R_m \cap T_m|}{|S_m|} \tag{3.4}$$

$$Recall_{B^3} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \frac{|R_m \cap T_m|}{|T_m|} \tag{3.5}$$

Following 3.4 and 3.5, first, all the mentions are calculated individually. Next, the individual recall and precision scores are averaged over all the mentions. Both in 3.4 and 3.5, $\mathcal{M}$ is the set of all mentions. This is the version of the B^3 metric in which all the mentions have the same weight. There is another version in which is possible to assign a different weight to each mention (refer to Bagga and Baldwin (1998) for more details).

By this formulation it is possible to see that singleton chains are not ignored by B^3 since the metric computes each individual mention. Likewise, large chains are not favoured by the same reason: the errors in the chains formation affect each individual mention's score. However, Luo (2005) reports shortcomings in the B^3 which lead to counterintuitive results. For instance, given a response set $R_1 = \{\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10},$
$e_{11}, e_{12}\}\}$ and a key set $T = \{\{e_1, e_2, e_3, e_4, e_5\}, \{e_6, e_7\}, \{e_8, e_9, e_{10}, e_{11}, e_{12}\}\}$, the B^3 recall is 100% (the precision is 37.5%). According to Luo (2005), this result is counterintuitive because the set of "true" reference entities is not a subset of the response entities. The same can be observed regarding the precision. Given the response set $R_2 = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\},$
$\{e_6\}, \{e_7\}, \{e_8\}, \{e_9\}, \{e_{10}\}, \{e_{11}\}, \{e_{12}\}\}$, the precision is 100% (the recall is 25%). Luo (2005) claims intersecting the response and key entities allows an entity to be used more than once, leading to counterintuitive results.

Furthermore, Stoyanov et al. (2009b) point out that the B^3 assumes that both the response set and the key set deal with the same set of expressions or mentions, i.e., both are a clustering over the same set of mentions. This is clearly not the case when the mentions are automatically identified by the system (in contrast to systems that use gold standard annotation for identifying referring expressions in a document – step 1 of the general algorithm presented in section 3.1). For using B^3 in such systems, a mapping is required so that a given mention in the response set correspond to its mention in the key set. Stoyanov et al. (2009b) calls $twin(m)$ the unique annotated/extracted mention to which the extracted/annotated mention is matched. A *twinless* mention is that which does not have any corresponding mention. Thus, extracted twinless mentions indicate the system extracted spurious mentions whereas annotated twinless mentions indicate the system failed to identify the mentions.

Stoyanov et al. (2009b) proposes two solutions to extend B^3 and make it capable of dealing with twinless mentions. The first is to keep all twinless extracted mentions. Keeping them, the definition of precision and recall remains the same when the mention has a twin and are defined as 3.6 and 3.7 for the mention m, respectively, when the mention is twinless.

$$Precision_{B^3_{all}}(m) = \frac{1}{|S_m|} \tag{3.6}$$

$$Recall_{B^3_{all}}(m) = \frac{1}{|T_m|} \tag{3.7}$$

The second possible way of dealing with twinless mentions in B^3 is to discard all extracted twinless mentions and penalising the recall by setting it to 0 for all twinless mentions. This solution (B^3_0) assumes that all extracted twinless mentions are spurious (Stoyanov et al., 2009b).

The CEAF metric

The Constrained Entity Aligned F-Measure (CEAF) is an entity-based metric as opposed to link-based MUC and mention-based B^3. The authors (Luo, 2005) proposed this metric for solving the problems pointed out by them regarding the intersection procedure used by both previous metrics which allow a mention to be used more than once in the evaluation of the entire partition. In CEAF,

a response chain R is mapped to at most one key chain K. This is done by computing the best of all possible one-to-one mappings $G(\mathcal{R}, \mathcal{K})$ where $\mathcal{R}$ is the response set of chains and $\mathcal{K}$ is the set of key chains. The best mapping g^* is the one that maximises the similarity $\Phi(g)$ for a mapping g, which is the sum over the pairwise similarity $\phi(R_i, K_i)$ over pairs of aligned R_i and T_i chains. Here, the pairwise similarity corresponds to the ϕ_3 similarity function presented by Luo (2005) which is defined as $\phi(R_i, K_i) = |R_i \cap K_i|$. Thus, the similarity between two chains R and K is the number of common elements in both chains. The CEAF precision and recall are defined in 3.8 and 3.9.

$$Precision_{CEAF} = \frac{\Phi(g^*)}{\sum_i \phi(K_i, K_i)} \tag{3.8}$$

$$Recall_{CEAF} = \frac{\Phi(g^*)}{\sum_i \phi(R_i, R_i)} \tag{3.9}$$

The precision is thus the ratio between total similarity of the best mapping g^* and the number of mentions in $\mathcal{K}$. Recall, by its turn, is the ratio between total similarity of the best mapping g^* and the number of mentions in $\mathcal{R}$. Stoyanov et al. (2009b) point out two problems with CEAF: (i) it assigns a zero score to each twinless extracted mention and (ii) weights all coreference chains with the same weight, no matter what their size is. Mainly because of the first problem, systems that rely on automatically extracted mentions have bad and unreliable results when evaluated with CEAF precision.

3.3 Reference Resolution in Portuguese

Research on reference resolution for Portuguese does not present as many works as the research for reference resolution for English language. However, in the last years, some researchers have put some effort in the development of anaphora resolution systems and, in a smaller extent, coreference resolution systems. This section presents some of these studies.

3.3.1 Anaphora Resolution

The bulk of work on anaphora resolution for Portuguese is not large. In this section, some of this these works are presented.

Paraboni (1997); Paraboni and Lima (1998) propose a Portuguese possessive pronominal anaphor resolution algorithm for third person intra-sentential pronouns. The algorithm rely on three different knowledge sources: surface patterns (which take into consideration syntactic parallelism); possessive relationship rules and sentence centering. The study reports difficulties in the interpretation of this kind of anaphora for Portuguese. For instance, the absence of gender and number agreement between the anaphor and the antecedent, the different syntactic functions this kind of pronouns can establish with the other constituents, and others. For tackling this problem a rule-based multi-agent architecture is proposed. The study reports an accuracy of approximately 92.97% for possessive pronoun resolution on a juridic corpus.

Aires et al. (2004) describe a study in which the Centering theory is evaluated for its use in pronoun resolution for Portuguese. The study was carried out using a corpus in order to check if the rules and constraints prescribed by the Centering theory hold for Portuguese and can be applied to pronominal anaphora resolution systems. The work reported results in the order of 51% of accuracy when using Centering theory for pronoun resolution for Portuguese.

Coelho (2005) adapted the Lappin and Leass (1994) approach to pronoun resolution to Portuguese. The scope of the implemented system was the resolution of third person and reflexive pronouns. As well as Lappin and Leass (1994)'s algorithm, it depends on full syntactic parse trees to perform the resolution. The adaptation to Portuguese makes use of the PALAVRAS syntactic analyser (Bick, 2000) for obtaining the full parse trees. The system was evaluated on three different corpora, one journalistic, one literary, and one juridic corpus and obtained 43.56, 31.32 and 35.15% of accuracy, respectively.

Chaves (2007); Chaves and Rino (2008) adapt to Portuguese the original Mitkov's algorithm for pronoun resolution. As well as Mitkov's approach, Chaves (2007) resolves only third person personal pronouns and makes use of shallow syntactic information. The system requires a preprocess step in which the shallow syntactic parsing is performed by the PALAVRAS syntactic parser. The algorithm was evaluated in the same three corpora thar Coelho (2005) evaluated his system. It achieved an accuracy of 67.01, 38 and 54%, respectively.

Santos (2008) describes an adaptation to Portuguese of the Hobbs pronominal resolution algorithm which was extended to include reflexive pronouns, not

considered in Hobbs' original algorithm. The system was evaluated on the same three corpora used by Coelho (2005) and obtained the following accuracy scores: 61.9, 44.24, and 43.21% on the journalistic, literary and juridic corpus respectively. Another evaluation was executed on a corpora merged with the pronouns of these three corpora plus the pronouns in the Summ-it corpus (Collovini et al., 2007). The total accuracy was of 45.84%.

Cuevas et al. (2008) investigate multilingual resolution of Portuguese personal pronouns to improve the accuracy of their translations to both Spanish and English in an underlying Machine Translation project. To carry out this investigation a corpus tagged using the PALAVRAS syntactic analyser was annotated with third person personal pronouns anaphoric relations. The pronoun resolution methodology follows the approach of Soon et al. (2001). The features used are: NUMBER_AGREEMENT, GENDER_AGREEMENT, FUNCTION_AGREEMENT (true if both noun phrases are subjects or objects), DISTANCE (the number of sentences between the two expressions), and PREPOSITION_TYPE (no preposition *eles*, 'they/them'; *deles*, 'of them'; or *neles*, 'in them'). The system was evaluated on the annotated corpus and obtained 70.3% of F-Measure for the coreferential class using a classifier induced by a decision tree algorithm. In this experiment, the FUNCTION_AGREEMENT feature was discarded since it was concluded in previous experiments with the same dataset that this feature degrades the overall performance of the system. Cuevas and Paraboni (2008) extend the feature set of their previous work with syntactic and semantic features and obtained an improvement in the performance (86.6% of F-Measure for coreferent expressions).

3.3.2 Coreference Resolution

Although Collovini and Vieira (2006a,b) are not concerned with anaphora or coreference resolution themselves, these studies present relevant work for both tasks based on previous studies for the English language (Vieira, 1998). Both (Collovini and Vieira, 2006b) and (Collovini and Vieira, 2006a) present an anaphoricity classifier for definite descriptions for Portuguese. The idea of the study is similar to the idea of Ng (2004). Based on relevant features for determining whether a definite description is classified either as anaphoric or as non-anaphoric, a classifier is induced using a decision tree algorithm. Both

studies report good results on anaphoricity determination that could be used for supporting anaphora resolution and coreference resolution systems.

Until the present date, the author of this work is not aware of any other study regarding coreference resolution for Portuguese except the one proposed by Souza et al. (2008). This work presents a noun phrase coreference resolution approach for Portuguese based on the approach introduced by Soon et al. (2001). The work relied on morphological, syntactical and limited semantic information provided by the PALAVRAS syntactic analyzer. The corpus used was the Summit corpus, already mentioned in previous sections and used in works on pronominal anaphora resolution for Portuguese. The learning algorithm employed was the J48 implementation of decision trees available in the WEKA (Hall et al., 2009) machine learning framework. The authors report a MUC score F-Measure of 51.3% and B^3 F-Measure of 69.66%.

In this chapter several works related to the task of coreference resolution were described. The principal approaches to the problem, its criticisms and proposed improvements were also discussed. In the next chapter it is presented the methodology of the present study that makes use of some of the previous research done in the are of coreference resolution.

Chapter 4

Methodology

In this chapter, an overview of the methodology developed in this study is described. In section 4.1, an overview of the methodology is presented. Specifics on each part of the methodology can be found in Chapter 5 for the English coreference resolution, in Chapter 6 for the alignment step, and in Chapter 7 for the Portuguese coreference resolution step.

4.1 Overview

The ultimate goal of this research is to extract coreference chains automatically from Portuguese texts. The basic idea to achieve this goal is to use a parallel corpus to project coreference relations from the English part of the corpus to the Portuguese part of the corpus. The relations projected are then used for training a supervised machine learning model that can be applied to Portuguese texts. Figure 4.1 shows an overview of the whole system.

The methodology is composed of several steps that can be roughly grouped into three main parts: annotation of resources for the corpus (English coreference resolution and Portuguese parsing and noun phrases extraction), alignment (sentence and word alignment of the English and Portuguese part of the corpus), and coreference resolution for Portuguese (instances generation, features generation and coreference resolution model). In the next sections, the architecture and the whole process as well as the parallel corpus are briefly described.

4.2 Corpus

The use of a parallel corpus is key to the method developed in this work. In the case of this study, an English-Portuguese parallel corpus was required. The parallel English and Portuguese corpus used for this work was extracted from the electronic version of the *Revista Pesquisa FAPESP* brazilian magazine[1]. The magazine is a monthly publication of the FAPESP foundation[2] and publishes news about domestic and international scientific policy, and about research carried out in Brazil and other countries.

This corpus has already been used in experiments for studies in Portuguese-Spanish and Portuguese-English statistical machine translation such as Aziz et al. (2008) and Aziz et al. (2009) and in research related to generating linguistic knowledge for machine translation using multilingual resources, such as Caseli (2007). It is formed by Portuguese, English and Spanish parallel texts extracted from the Environment, Science, Humanities and Technology supplements of the electronic magazine magazine. The number of tokens and sentences of the corpus are summarized in table 4.1. The corpus contains 646 texts with a total of 17,426 sentences for the English part and 18,159 sentences for the Portuguese part. The English part contains around 464,240 tokens and the Portuguese part contains about 433,212 tokens.

FAPESP Corpus		
Language	Tokens	Sentences
Portuguese	433,212	18,159
English	464,240	17,426

Table 4.1: The number of tokens and sentences in each part of the FAPESP corpus.

Additionally, two other corpora are used in this work. The NP4E corpus, described in Chapter 5, is used for training the English coreference resolution model, and the Summ-It corpus, described in Chapter 7 is used for testing the Portuguese coreference resolution model.

[1] `http://revistapesquisa.fapesp.br/`
[2] `http://www.fapesp.br/en/`

4.3 Automatic Corpus Annotation

The first step of the process is to annotate the corpus with the required data. At this point, the English part of the parallel corpus should be annotated with coreferential data. The Portuguese part, by its turn, must have its noun phrases identified. These two layers of linguistic information will enable, in a further step, the projection of the coreferential links present among the noun phrases contained in the English part to the noun phrases contained in the Portuguese part.

4.3.1 Coreference Resolution for English

The methodology assumes that no manual annotation of coreferential links is performed. Therefore, the idea is to obtain the coreference chains for the English part automatically. For that, one coreference resolution system for the English language should be employed. In this work, the system used is the Reconcile system (Stoyanov et al., 2010). The complete description of this step as well as the description of Reconcile are in Chapter 5.

4.3.2 Parsing and Noun Phrase extraction for Portuguese

The identification of noun phrases in the Portuguese part of the corpus should also be performed in an automatic fashion, without resorting to manual annotation. Besides that, for each noun phrase, syntactical, morphological and semantic data are required to generate feature vectors for the supervised machine learning based coreference resolution model for Portuguese.

This step needs to be performed explicitly only for the Portuguese part of the corpus, as the noun phrases in the English part of the corpus are identified during the coreference resolution process. The Portuguese noun phrases are identified using the Constraint Grammar based parser PALAVRAS (Bick, 2000). The authors report 99% of accuracy for part-of-speech tagging and about 97% of accuracy for syntactic function detection (Bick, 2000).

4.4 Alignment

The alignment step enables the projection of the coreference chains in the English part of the corpus to the noun phrases in the Portuguese part of the corpus. Having the noun phrases that form the chains in English and the noun phrases in Portuguese, it is possible to establish a mapping between the two phrases by mapping their heads. This mapping is enabled through the word (or lexical) alignment.

Even though the method proposed in this work relies on a parallel corpus, most of the parallel corpora available do not have a word-by-word alignment as it is required by this step. As the input of most word alignment algorithms require that the corpus is sentence aligned, it is also necessary to run a sentence alignment algorithm before the word-by-word alignment in case the corpus is not sentence aligned.

The alignment step receives the corpus preprocessed by the coreference resolution system for English and the Portuguese parser, using the sentence splitting provided by the two tools, for the English and Portuguese parts of the corpus, respectively. For this study both the sentence alignment and the word alignment are required. The sentence alignment algorithm employed is an implementation of the Translation Corpus Aligner (Hofland, 1996) called TCAlign (Caseli, 2003). The word alignment algorithm used was the one implemented in GIZA++, described by Och and Ney (2003), part of the Moses statistical machine translation toolkit[3]. The alignment processes are explained in detail in chapter 6.

4.5 Coreference Resolution for Portuguese

The final step is to perform the actual coreference resolution for Portuguese. In this step, the projected coreference chains are used for training a supervised machine learning coreference resolution model. The resolution system is a single-mention pairwise model as the one described in Chapter 3. It is formed by four modules: an instance generation module, an instance projection module, a feature vector generation module and a supervised machine learning based classifier along

[3] `http://www.statmt.org/moses/`

with a clustering algorithm.

The instance generation model forms pairs of noun phrases (antecedent and anaphor) using the coreference chains generated by the coreference resolution system for English employed in the first step (section 4.3.1). Given the errors introduced by the identification of English NPs and by the alignment process, the English noun phrases are not directly mapped to Portuguese noun phrases. Instead, a matching algorithm is used to identify which is the best Portuguese noun phrase to be aligned to the English noun phrase. The matching algorithm is implemented in the instances projection module.

Once a pair is identified in the Portuguese data, features are extracted in order to produce training examples. The task of identifying the matching pairs is performed by the instance projection module and the task of generating the feature vectors is performed by the features vector generation module.

The feature vectors are used to train a machine learning classifier. For each pair, the classifier decides whether the pair is coreferent or not. Having the class of each pair, the model clusters the pairs into chains. The whole coreference resolution model is described in more detail in Chapter 7.

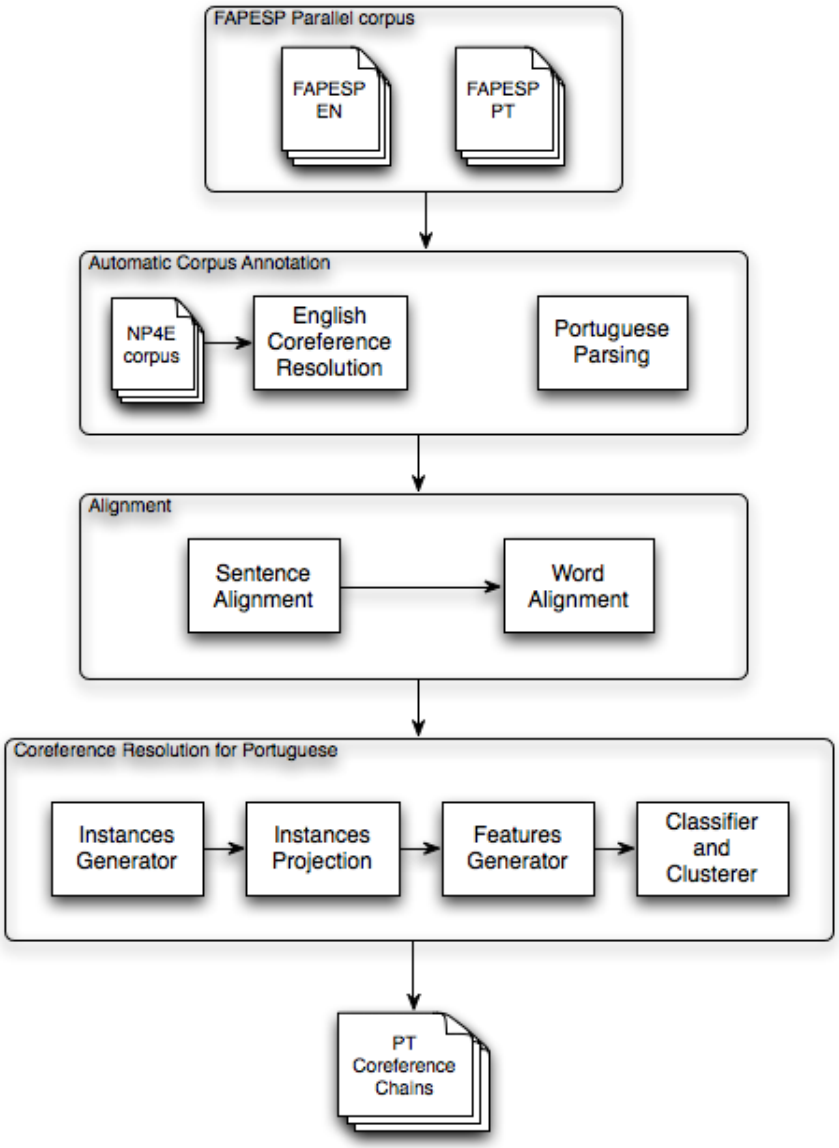

Figure 4.1: The proposed architecture for the coreference resolution system.

Chapter 5

English Coreference Resolution

This chapter describes the coreference resolution system used in this study for the English language. It is used for automatically obtaining coreferential annotation for the English part of the FAPESP parallel corpus. The following sections present the system, as well as the resources it uses.

5.1 Reconcile

The system adopted is called Reconcile and was proposed by Stoyanov et al. (2009a, 2010). Reconcile is an end-to-end coreference resolution system that can be used in an off-the-shelf manner. The term end-to-end is used in this chapter for describing a system that does not rely on manual annotation that could help the preprocessing steps of the coreference resolution task. All the required annotation is provided by tools. The kind of annotation preprocessed before the coreference resolution includes identification of noun phrases, classification of anaphoric noun phrases and non-anaphoric noun phrases, identification of named entities, and identification of semantic types of noun phrases.

Reconcile was designed as a modular Java architecture that incorporates basic design features of the single-mention pairwise model to coreference resolution. The architecture is similar to some supervised learning-based coreference resolution systems, such as Soon et al. (2001); Ng and Cardie (2002b) and Bengtson and Roth (2008). This model is described in more depth in section 3.2.1. According to the authors, the system is flexible enough to accommodate other approaches to coreference resolution (like the ones proposed by Yang

et al. (2003); Luo et al. (2004); Haghighi and Klein (2007) – briefly described in section 3.2.3).

The architecture of Reconcile is shown in figure 5.1. The figure was designed based on the architecture presented in Stoyanov et al. (2010). The architecture is composed of a preprocessing step, a feature generation step, a classification step and a clustering step. The input is a set of texts and the output is the set of texts with coreference annotation added to the texts. For training the model, the system also requires that the input texts are annotated with coreferential data. For that, the authors have bundled the NP4E corpus with Reconcile and the system's default model is trained over the NP4E. In the following sections, the steps depicted in figure 5.1 are described. The NP4E corpus is described on section 5.1.5.

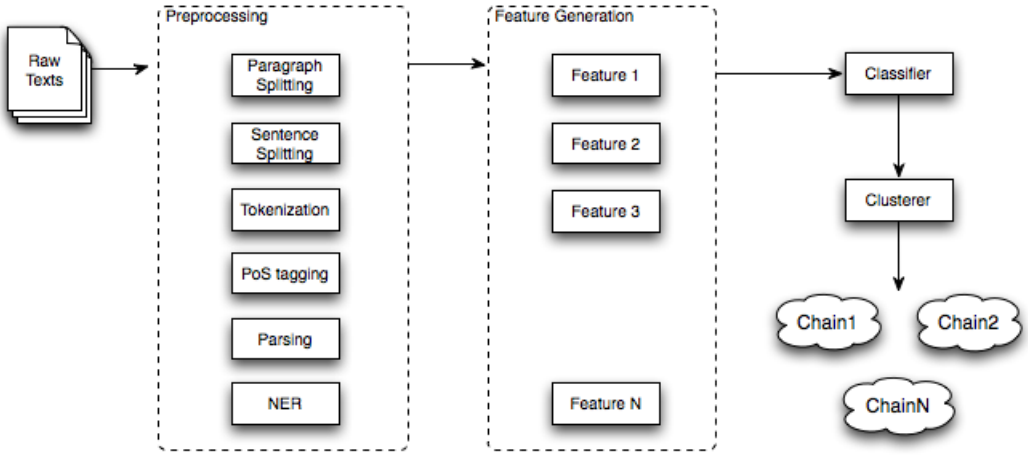

Figure 5.1: The Reconcile coreference resolution system architecture.

5.1.1 Preprocessing

The resolution process begins in the preprocessing step where the noun phrases and the named entities are identified. The preprocessing step is composed of the following modules: paragraph splitting, sentence splitting, tokenization, part-of-speech tagging, parsing, and named entity recognition.

All the modules are implemented using external open source tools freely available on the internet. There are usually two options of external tools for each

Module	Options
Sentence splitting	OpenNLP[1] and UIUC[2]
Tokenization	OpenNLP
PoS tagging	OpenNLP or the output of one of the parsers below.
Parsing	Stanford (Klein and Manning, 2003) and Berkeley (Petrov et al., 2006; Petrov and Klein, 2007)
NER	OpenNLP and Stanford (Finkel et al., 2005)

Table 5.1: Preprocessing tools available in Reconcile.

module. The options available in the Reconcile package for each preprocessing module are summarized in table 5.1.

There are two ways of using Reconcile: either from the source code, or using the executable JAR file. The tool for each module is specified through a configuration file when Reconcile is built from the source. As well as the modules in the preprocessing step, Reconcile allows the user to choose which features to use when running the system from the source code. All the data produced by the external tools is stored and used in further steps of the resolution process.

5.1.2 Features Generation

A great part of the data generated in the preprocessing step is used in the features generation step. This step generates feature vectors for pairs of noun phrases that may help the classifier decide whether a given pair is anaphoric.

The authors report that more than 80 features are included in Reconcile. These features were inspired by the works of Soon et al. (2001) and Ng and Cardie (2002b). They can be divided into different categories (as defined by Ng and Cardie (2002b)), namely:

- **Lexical features:** features that compare both noun phrases using string matching algorithms. Examples of this class of features, among others, are:
  - The Soon string matching (SoonStr), described in chapter 3, which compares the two expressions after discarding determiners;
  - PNStr, which checks whether both expressions are proper names and matches the same string;

  - WordsStr, which checks whether two non-pronominal expressions match;
  - HeadMatch, which checks whether the heads of the noun phrases match;
  - PNSubstr, which checks if one expression is the substring of the other in case both expressions are proper names.

- **Grammatical features:** features that compare the antecedent and the anaphor using grammatical information, either morphological, syntactical, or using heuristics. Examples of such features, among others, are:
  - BothPronouns, which checks if both expressions are pronouns;
  - BothDefinites, which checks if the expressions begin with the article "the";
  - BothProperNames, which checks if the expressions are proper names;
  - BothSubjects, which checks if both expressions have the role of subject in the sentence they appear in;
  - Agreement, which checks whether both expressions agree in gender and number.
  - Embedded1 and Embedded2, check if the first noun phrase is embedded in the second (Embedded1) or the opposite (Embedded2);

- **Semantic features:** features that use semantic resources for asserting whether the mentions in the pair corefer. Examples of such features are:
  - WordNetSense, uses WordNet (Miller, 1995; Fellbaum, 1998) to fetch the first sense that both expressions share;
  - WordNetDistance, uses WordNet to measure the distance of the two expressions in a Synset tree;
  - Subclass, checks whether one expression is present in a subclass of the other.

- **Other features:** features that do not fall in any of the previous categories, such as:

– SameParagraph, that checks if both noun phrases are in the same paragraph;

– IAntes, checks if the first expression is inside a quoted string;

5.1.3 Classifier

In this step, the feature vectors generated within the previous step are used for either training a new model or for applying a previously trained model to new texts. When training a new model, Reconcile uses data from two sources. One source is the features generated in the previous steps. The other is a class given by the corpus which is manually annotated with coreferential data in order to induce a new model using the configured machine learning algorithm. When applying a previously built model to a new set of texts, the feature vectors do not contain any information regarding the actual class of the instance pair. In this case, the classifier receives a feature vector representing a pair of noun phrases and returns a score indicating the likelihood of the two expressions being coreferent.

Reconcile provides different machine learning algorithms for training the coreference models. The available algorithms are: the learning algorithms in the Weka toolkit (Witten and Frank, 2005; Hall et al., 2009), accessed through the Weka API, and two implementations of Support Vector Machines, the `libSVM` library (Chang and Chih-Jen, 2001), and the SVM^{light} package (Joachims, 2002).

5.1.4 Clusterer

In this step, the system uses a clustering algorithm to group the anaphoric pairs that relate to the same entity into clusters. If the score of a given pair is below the predefined threshold of the classifier, the pair is ignored. Reconcile implements the single-link clustering, the best-first clustering and the most recent first clustering algorithm described in Chapter 3.

The chains extracted by Reconcile are annotated in the middle of the text on a copy of the input file (inline annotation). The system marks the texts with tags which delimit the noun phrases and it assigns one identifier for the noun phrase and one identifier to the chain to which it belongs. The first line of one output file is presented in figure 5.2.

```
<NP NO="0" CorefID="15">The teeth of
<NP NO="1" CorefID="1">the oldest orangutan</NP></NP> .
```

Figure 5.2: One line of an output file generated by Reconcile.

5.1.5 Corpus

The NP4E corpus (Hasler et al., 2006) is the product of a project whose aim was to develop annotation guidelines for noun phrase and event coreference for newswire texts in the domain of terrorism and security. The corpus has 50,000 words and the texts are a subset of the Reuters corpus (Rose et al., 2002). The complete annotation guidelines of the NP4E corpus is available at the website[3] of the Computational Linguistics Group of the University of Wolverhampton.

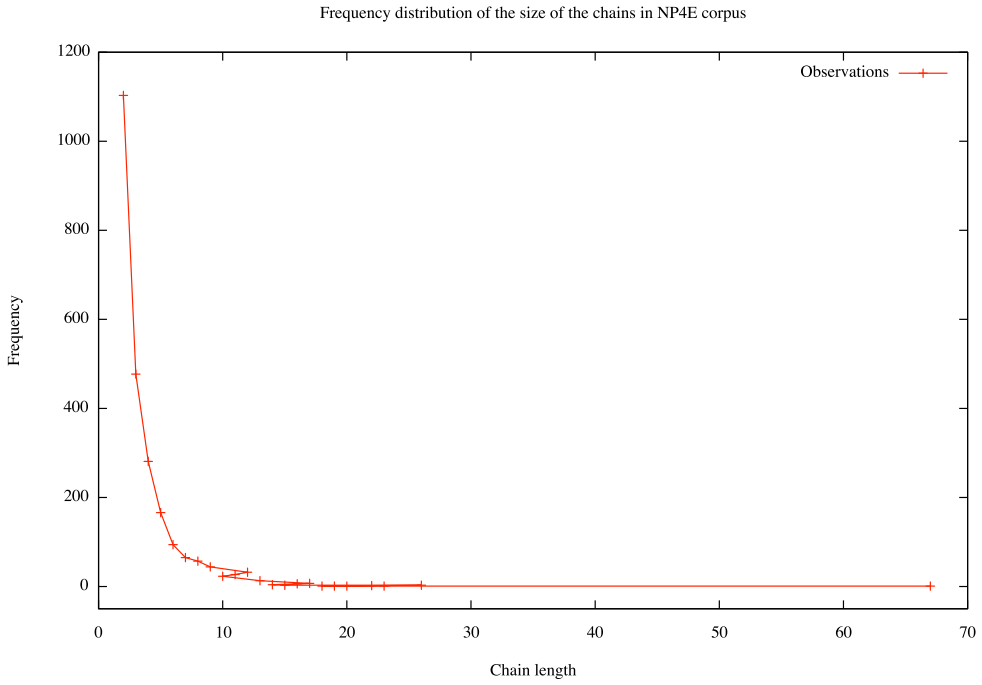

Figure 5.3: The NP4E corpus frequency distribution of chains' size.

The frequency distribution of the sizes of the chains annotated in the NP4E corpus is summarized in figure 5.3. Singleton chains, i.e., chains formed by

[3] http://clg.wlv.ac.uk/projects/NP4E/#corpus

only one expression were observed 9,228 out 11,640. Chains with two or more expressions correspond to 2,412 observations. Therefore, the vast majority of the chains, approximately 79%, are singleton chains. This is a characteristic of the annotation guidelines followed: all the noun phrases are annotated as markables, leaving the expressions that are new in the discourse in singleton chains.

Chapter 6

Alignment

A parallel corpus is a set of texts in which each text has one or more translations in different languages. Besides being parallel, the texts may be also aligned. The alignment consists of finding correspondence points between the translations of the texts, usually between a source text and a target text or translation.

Here, the target text means the translation of the source text. Therefore, different levels of translations can be aligned between the source and the target texts: at the paragraph level, at the sentence level, at the word level and even at the character level.

The main objective of this step is to provide a word level alignment between parallel texts. The input of this stage is a corpus of parallel texts with one sentence per line, for both sides of the parallel corpus. The output is one file containing the word indexes for each token in the line. Figure 6.1 shows the modules which compose the alignment pipeline.

This chapter is structured as follows. In section 6.1 the sentence alignment is described. In section 6.2, the intermediate stages between the sentence alignment and the word alignment are described. In the last section, 6.3, the word alignment is described.

6.1 Sentence Alignment

The objective of sentence alignment is that given two documents, the original text (the source) and its translation (the target), find which sentence or sentences in the target text are the translation of a given sentence in the source text. The

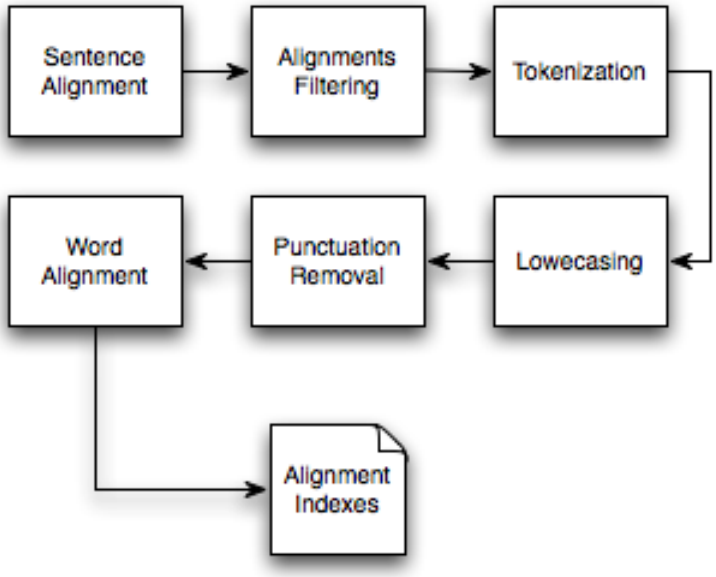

Figure 6.1: The alignment pipeline.

most common alignment observed in sentence alignment is when one sentence in the target text corresponds to one sentence in the source text (i.e. 1-1). This characteristic was observed by Gale and Church (1993) for the English-German and English-French language pairs and by Caseli et al. (2004); Caseli (2003) for the English-Portuguese language pair.

There are other possible types of alignments. There are the alignment cases in which no sentence is aligned with any sentence in the source or in the target (i.e. 0-1 and 1-0), and there are cases such as expansions, contractions and unions (Caseli, 2003). Expansions occur when the alignment is $n - m$ where $n < m$ and $n, m \geq 1$ takes place. Contractions occur when the alignment is $n - m$ where $n > m$ and $n, m \geq 1$ takes place. Unions occur when the alignment is $n - n$ where $n > 1$ takes place.

Several methods for sentence alignment were proposed in the context of the ARCADE project (Véronis and Langlais, 2000). The goal of the ARCADE project was to develop methods for sentence and word alignment of parallel texts. The approaches use different features as criteria to do sentence alignment. However, most of the approaches are based on the following information: length

of sentences, anchor words, cognate words, part-of-speech tags, among others. These information are used as the alignment criteria of the different methods that employ them.

Santos and Oksefjell (2000); Caseli (2003); Caseli et al. (2004) presents and compares some works for the English-Portuguese language pair. One of the best performing sentence alignment algorithm among the methods evaluated by Caseli et al. (2004) is the Translation Corpus Aligner (TCA), proposed for the English-Norwegian language pair in Hofland (1996). The TCA was adapted to the European Portuguese by Santos and Oksefjell (2000) and to the Brazilian Portuguese by Caseli (2003).

TCA relies on different alignment criteria to perform the sentence alignment: the sentences' length, a bilingual anchor words list, a simple heuristic to determine proper nouns candidates (capitalized words), and a list of special characters (punctuation such as the question mark, exclamation mark, and the full stop). Caseli et al. (2004) reports results from 90 to 100% of F-Measure for all the types of alignment on four different English-Portuguese corpora. The implementation of TCA used in this research is the TCAlign[1]

The input of TCAlign is input the alignment pipeline. As mentioned previously, the format of the input files must have one sentence per line. In the methodology presented in this study, the sentence boundaries for the English and Portuguese parts of the corpus are the same ones extracted by Reconcile and PALAVRAS, employed in the previous step. The output of TCAlign is a XML-like file with the alignments. The alignments can be one-to-one, multiple or omitted (0-1 or 1-0).

6.2 Alignment Intermediate Modules

The modules described in this section basically apply a series of transformations over the output of the sentence alignment module. The objective is to prepare the output of the sentence alignment to input in the word alignment system. The transformations (figure 6.1) are: alignments filtering, tokenization, lowercasing, and punctuation removal.

The input of the word alignment module must be formed only by one-to-one

[1] `http://www.nilc.icmc.usp.br/nilc/projects/aligners.htm`

alignments. As the output of the sentence alignment produces different types of alignments, a filtering step is required. In the alignments filtering module, all the alignments that are not one-to-one are discarded. Also, the output of this module is raw text as opposed to the XML-like layout of the previous step.

The next module is the tokenization module. Tokenization is necessary because the word alignment is performed over words and they need to be properly delimited to be processed.

The lowercase transformation is performed so that the word aligner does not consider words with different cases as two different samples in the corpus frequency distribution. Also with the objective of aiding the word aligner, all the punctuation are removed. Not having punctuation avoids the need of alignment of such tokens, improving the word aligner performance. The last module, the word alignment, is described in the next section.

6.3 Word Alignment

The word alignment problem can be defined as the problem of finding the correspondence between contiguous sequence of words that form the sentences in a parallel text. The word alignments do not have to be always of the type one-to-one. The alignments may be multiple or there may be no alignment points, in the same way as observed for the sentence alignment.

Och and Ney (2003) presents a comparison of different word alignment models. The word alignment method used for this project is one of the models in this comparison, GIZA++[2] (Och and Ney, 2003). GIZA++ is a statistical word alignment toolkit that uses the IBM models (Brown et al., 1994) and the Hidden-Markov alignment model (Och and Ney, 2000; Vogel et al., 1996) to find the best mappings between sequences of contiguous words in a parallel text.

Caseli (2007) compares two methods for word alignment in Brazilian Portuguese texts (the same corpus used in this project) and reports results for GIZA++ around 90% for precision and around 92% for recall.

GIZA++ is the last module in the alignment pipeline. It receives the texts processed by the previous steps. Before running the word aligner the texts are all concatenated into one single file that represents the whole corpus. The output of

[2] `http://code.google.com/p/giza-pp/`

the word aligner is a file in which each line of the input files is formed by pairs of indexes that represent the mappings between the tokens of the two parallel texts.

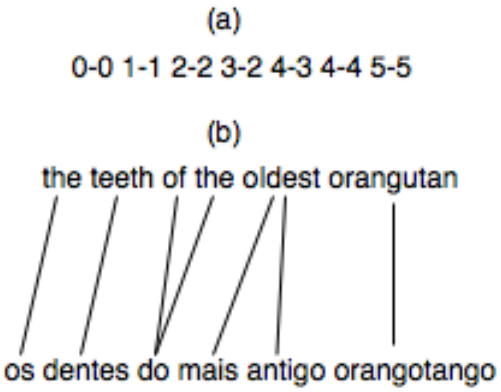

Figure 6.2: The representation of one line of the word aligner output.

One line of GIZA++ word alignment indexes file is shown in figure 6.2.a. Each pair of indexes is a mapping between two tokens in two parallel texts. One index in the source text may be paired with more than one index in the target text. The inverse is also true. In this line there are not examples of omissions (0-1 or 1-0) alignments but they may occur. A representation of the line with its one-to-one alignments as well as multiple alignments is shown in the same figure in 6.2.b. This file is used for performing the projection of noun phrases explained in chapter 7. In the figure, the two sentences, in English and in Portuguese come from the input files preprocessed by the alignment pipeline before the word aligner.

Chapter 7

Coreference Resolution for Portuguese

This chapter describes the Portuguese Coreference resolution system implemented for this project. The approach is the same as the one described in section 3.2.1 and follows the rationale of Soon et al. (2001); Ng (2002) and Stoyanov et al. (2010). In these approaches, pairs of expressions are generated, classified as coreferent or not and then clustered.

The system has two different stages or modes: the training mode and the resolution mode. In the next sections the system is presented in function of these two stages and both modes are described taking into consideration the modules that form them.

7.1 Training a Coreference Resolution Model

On the training mode, the system receives English coreference chains and noun phrases as input and generates a machine learning based classifier as its output. The training mode pipeline is shown in figure 7.1.

7.1.1 Generation of Training Data

The first module in the training pipeline is the instances generation module. This module is responsible for generating pairs of mentions. When training a classifier, the system receives as input the noun phrases and the coreference chains

extracted from the English part of the parallel corpus during the automatic corpus annotation step (chapter 5). In the current implementation, Reconcile is used for extracting the noun phrases.

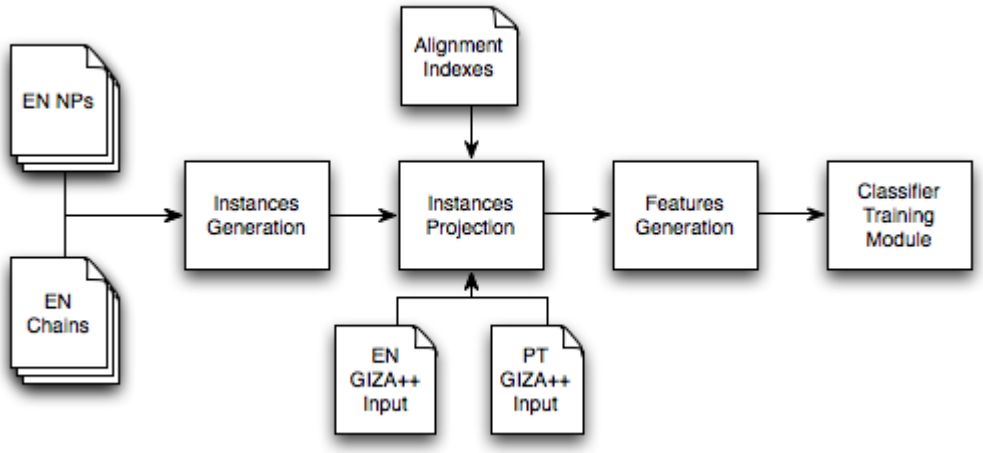

Figure 7.1: The coreference resolution system for Portuguese pipeline for training a classifier.

The instances generation module outputs two classes of pairs: coreferent pairs and non-coreferent pairs. The algorithm for generating coreferent pairs uses the coreference chains extracted by the coreference resolution system for English. It takes each chain and, for each non-pronominal mention it forms pairs of adjacent mentions. This is the same method as the one used by Ng and Cardie (2002b) for generating positive instances presented in section 3.2.2.

The algorithm for generating non-coreferent pairs is the same as the one used by Soon et al. (2001); Ng and Cardie (2002b), and consists of pairing mentions that appear in between the positive pairs with the anaphor of each pair. This method is presented in section 3.2.1.

7.1.2 Projection of Training Instances

The next step in the pipeline performs the projection of the instances from one part of the parallel corpus to the other. For that, the instances projection module uses the instance pairs generated by the previous step and the words mapping processed by the word aligner (described in section 6.3).

Besides the instance pairs and the words alignment file, in the current implementation, the projection algorithm requires the two files passed as input to GIZA++ to correctly process the antecedents and anaphors projection. Each line in the words alignment file refer to one of the lines in these two parallel texts. Figure 6.2 in section 6.3 shows the relationship between the words alignment file and the two GIZA++ input files.

Figure 7.2 shows a scheme of the projection process. The projection algorithm works as follows: for each instance pair, the first step is to take the head of the antecedent and the head of the anaphor of the pair (i.e. *Head* in figure 7.2) and to search for them in the line they occur in the GIZA++ input files. The objective is to find the token that corresponds to the head word of each expression in the pair (i.e. *String Match* in figure 7.2). Currently, the search is implemented as a simple string match search. If both the head of the antecedent and the head of the anaphor are found, the process goes to the next step.

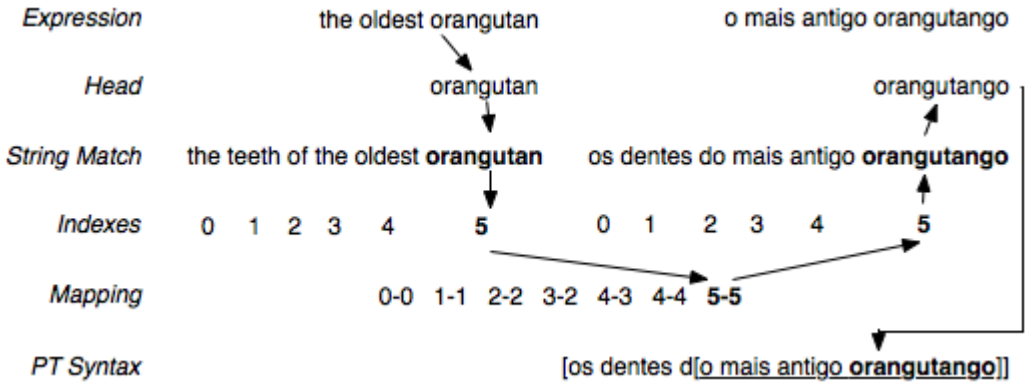

Figure 7.2: The representation of the projection of one expression.

Both heads of the antecedent and of the anaphor are mapped to indexes which are the position of the words in a sentence in the Portuguese part of the corpus (*Indexes* in figure 7.2). In the next step of the algorithm the indexes are used in conjunction with the word mapping (*Mapping* in figure 7.2) to find to which words each expression points to in Portuguese. If the Portuguese word found is the head of a noun phrase, the projection is made (*PT Syntax* in 7.2).

7.1.3 Features Extraction

After projecting the pairs, the list of projected instance pairs is processed and for each pair, a set of features are extracted. The features are based on previous work done in coreference resolution, mainly Soon et al. (2001); Ng and Cardie (2002b); Souza et al. (2008) and Recasens et al. (2010). The features extracted are:

- `head_match`: a boolean value that indicates whether the head of the antecedent and the head of the anaphor are the same.
- `subs_match`: true if the antecedent is a substring of the anaphor or if the anaphor is a substring of the antecedent. False otherwise.
- `ant_ne_type`: if the antecedent is a proper name, this feature assigns the type of the named entity recognized by the PALAVRAS parser. Possible values are labels for 157 different prototype classes such as animals, plants, humans, places, vehicles, among others. A complete list of the semantic tags implemented in PALAVRAS are available at the VISL (Visual Interactive Syntax Learning) website[1].
- `ana_ne_type`: the same as `ant_ne_type` but for the anaphor.
- `gender_agrmnt`: true if the heads of both expressions agree on gender. False otherwise.
- `number_agrmnt`: true if the heads of both expressions agree on number. False otherwise.
- `ant_subj`: true if the antecedent is the subject of the sentence where it appears. False otherwise.
- `ana_subj`: true if the anaphor the subject of the sentence where it appears. False otherwise.
- `ant_appos`: true if the antecedent is an apposition. False otherwise.
- `ana_appos`: true if the anaphor is an apposition. False otherwise.

[1] `http://beta.visl.sdu.dk/visl/pt/info/portsymbol.html#semtags_nouns`

- `sem_class_agrmnt`: true if the heads have the same semantic class
- `word_overlap`: computes the word overlap of the two expressions. The word overlap is calculated by taking all the tokens in the expression that are not punctuation and computing the number of tokens in the intersection of the two expressions divided by the sum of the number of tokens in each expression as follows: $over = \frac{2|ant\_tokens \cap ana\_tokens|}{|ant\_tokens|+|ana\_tokens|}$. The values can be one of *"0"*, for no overlap; *"1"*, for complete overlap; *"point25"* for a ratio of less than .25; *"point50"* for a ratio between .25 and .50; *"point75"*, for a ratio between .50 and .75; and *"less1"* for a ratio between .75 and 1 (not inclusive).

The features are extracted using the annotation provided by the PALAVRAS parsing system which provides deep parsing for the Portuguese language. The extracted feature vectors are written to an ARFF (Attribute-Relation File Format) to serve as input for the WEKA machine learning toolkit.

7.1.4 Classifier Induction

The last module of the coreference resolution system when run in training mode is the induction of a classifier. This module takes the ARFF generated and, through the WEKA API (Application Programming Interface) it creates a classifier using the instance pairs.

In the current implementation of the system, experiments with the JRip implementation of the decision rules algorithm proposed by Cohen (1995). Chapter 8 presents the evaluation with the results for the classifier. The output of the training process is the trained model and the ARFF generated by processing the projected instance pairs.

7.2 Extracting Coreference Chains

On resolution mode, the coreference resolution system receives Portuguese noun phrases as input and clusters them into chains, using the classifier induced in the training mode. The resolution mode pipeline is shown in figure 7.3. In the current implementation, the noun phrases are extracted using the PALAVRAS parsing system.

The instances generation algorithm for the resolution mode is different from the algorithm used in the training stage, described in section 7.1.1. There are not class labels. All the instances are unlabeled and the feature vector contains only the features processed in the feature generation step.

For each text passed to the system, the *Closest-First* clustering algorithm is employed (described in section 3.2.1). The classifier used is the one generated by the training mode. Each pair has the same features as the ones described in section 7.1.3, above.

Once a coreferent pair is found, it is stored in a graph structure. When all the noun phrases in a text were processed, the coreference chains are formed by using an algorithm to find the connected components in a graph.

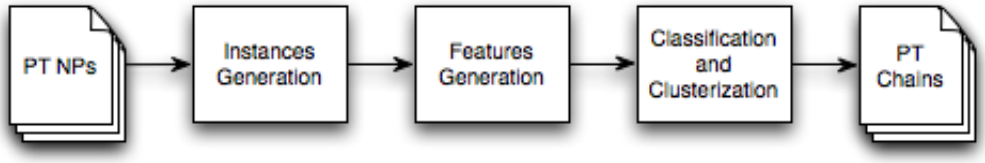

Figure 7.3: The coreference resolution system for Portuguese pipeline in resolution mode.

Chapter 8

Evaluation

In this chapter the methodology presented on chapters 4 to 7 is evaluated and discussed. The approach is evaluated over the 646 texts that form the FAPESP corpus. Additionally, the coreference resolution system for Portuguese is tested using the Summ-It corpus and its results are compared to the gold standard annotation of this corpus.

8.1 Coreference Resolution for English

In this section, the configuration settings used for running the Reconcile coreference resolution system for English are presented. Furthermore, the results obtained are presented and discussed.

8.1.1 System Configuration

For the experiments described here, Reconcile was run using the JAR (Java ARchive) file version 1.0 provided at the tool's official site [1]. This mode of execution does not allow any change in the configuration of the system. Therefore, the default settings were used. These settings are summarized below.

Preprocessing

The preprocessing tools used for the experiment are listed in table 8.1. All the tools that rely on machine learning or statistical models that require training

[1] `http://www.cs.utah.edu/~ngilbert/ccount/click.php?id=1`

over annotated corpora were run with models trained by their developers and included in Reconcile as binary files.

Preprocessing Tools	
Module	**Tool**
Paragraph Splitter	Ad-hoc implementation
Sentence Splitter	OpenNLP
Tokenization	OpenNLP
PoS tagging	OpenNLP
Parsing	Berkeley parser
NER	Stanford NER system

Table 8.1: Preprocessing tools used for running Reconcile.

Features

The features used with this version of Reconcile are a subset of 62 out of the 89 features implemented. The features used in the default configuration of Reconcile are summarized in table 8.2. The features are classified into four different classes: lexical, grammatical, semantic and other.

The classes and some of the features are briefly described in section 5.1.2. The complete reference of all the features along with their descriptions is available in Stoyanov et al. (2009a).

Classifier Model

The classifier that comes bundled with Reconcile is an WEKA's implementation (Witten and Frank, 2005) of the Averaged Perceptron algorithm. The NP4E corpus was used for training the model. More information about the corpus may be found in section 5.1.5.

Clustering Mechanism

The default clustering method configured in the JAR distribution of Reconcile is the single-link clustering. The single-link clustering algorithm processes the transitive closure of all the linked pairs. For defining which pairs are coreferent, the system uses the value computed by the classifier for each instance pair. All

Extracted Features	
Feature Class	**Feature**
Lexical	SoonStr, ProStr, WordsStr, WordOverlap, Modifier, WordsSubstr, ProComp, PNStr, PNSubstr, InQuote1, InQuote2, BothProperNouns, Alias, IAntes, BothInQuotes, ContainsPN, ProperNoun, ProperName, HeadMatch.
Grammatical	Pronoun1, Pronoun2 , Definite1, Definite2, Demonstrative2, Embedded1, Embedded2, BothEmbedded, BothPronouns, BothSubjects, Subject1, Subject2, Appositive, MaximalNP, Gender, Number, Span, Binding, Contraindices, Syntax, Indefinite, Indefinite1, Prednom, Pronoun, Constraints, Agreement, PairType.
Semantic	Animacy, ClosestComp, WordNetClass, WordNetDist, WordNetSense, Subclass, WNSynonyms.
Other	SentNum, ParNum, AlwaysCompatible, RuleResolve, SameSentence, ConsecutiveSentences, Quantity, ProResolve, SameParagraph.

Table 8.2: Features utilized for running Reconcile.

the pairs are filtered using a given threshold. The threshold value for the default configuration is set on 0.45.

8.1.2 Coreference Chains Extraction Evaluation

The JAR distribution of Reconcile was run over the 646 files of the FAPESP corpus. The English coreference resolver recognized 127,942 noun phrases and extracted 94,990 coreference chains from the whole corpus. The authors report MUC and B^3 scores F-Measure between 60 and 70% for the MUC6 and MUC7 corpora. As the English part of the FAPESP corpus is not manually annotated with coreference chains, it is not possible to use the coreference scoring metrics to measure the performance of Reconcile. However, it is possible compute some simple statistics to describe the chains.

Most of the chains extracted are singleton chains: they are 82,272 out of 94,990 or 86.61%. The second most numerous chains are the ones formed by two

expressions, 7,367 or 7.76%. The frequency distribution of the sizes of chains extracted by Reconcile is presented in figure 8.1 and also in table 8.3, which presents the distribution for the first 20 chain sizes.

It is interesting to notice the large difference in the percentage of singleton chains and in the percentage of chains formed by two and three expressions (table 8.3). Whereas the singleton chains represent roughly 86% of the extracted chains, chains formed by two and three expressions represent about 7 and 2% of the extracted chains.

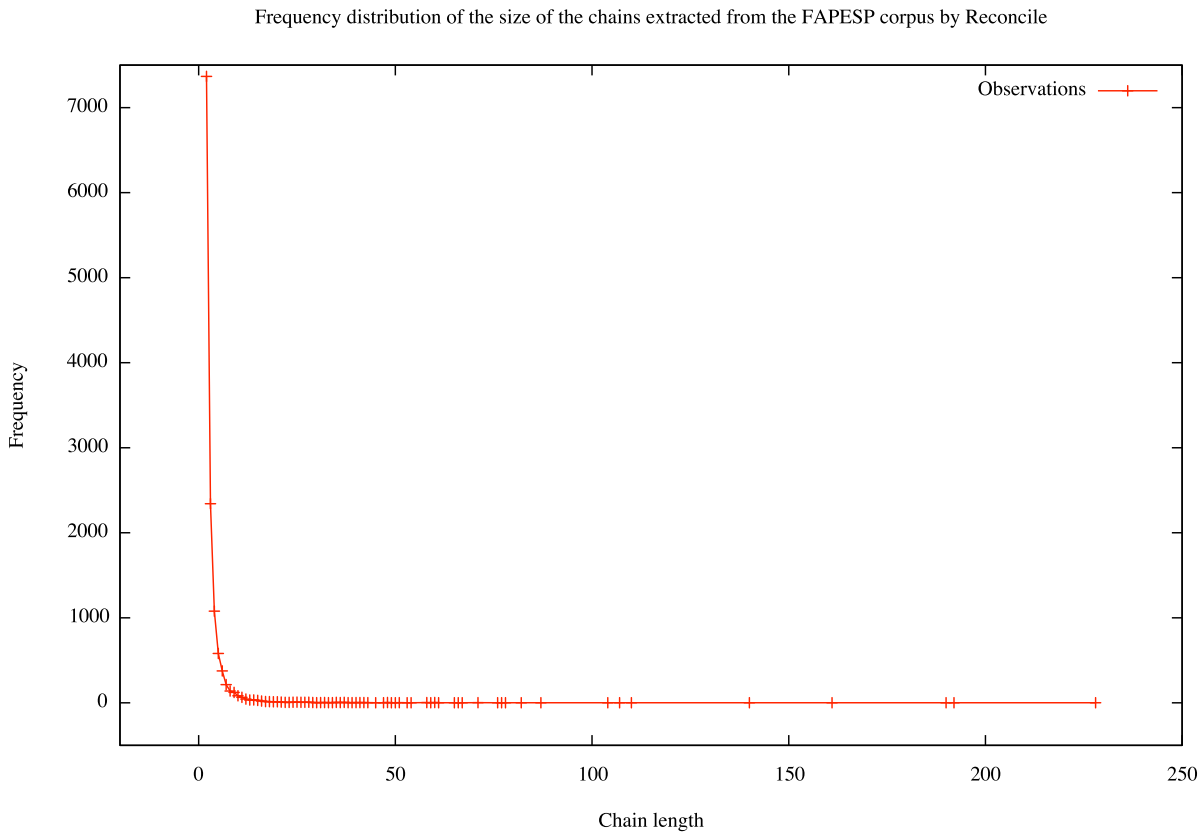

Figure 8.1: The frequency distribution of the sizes of the chains extracted by Reconcile from the FAPESP corpus.

Looking at the frequency distribution it is possible to observe that the smaller the chain size, the most frequent it is. Therefore, the frequency of very long chains tends to zero. The same behavior is observed on the frequency distribution of the manually annotated chains on the NP4E corpus, represented in figure 5.3.

As well as in the NP4E corpus, the most frequent chain size of the extracted

Extracted Chains Size Distribution			
Chain Size	Frequency	% (w/ singletons)	% (w/o singletons)
1	82,272	0.8661	-
2	7,367	0.0776	0.5793
3	2,342	0.0247	0.1841
4	1,078	0.0113	0.0848
5	581	0.0061	0.0457
6	375	0.0039	0.0295
7	214	0.0023	0.0168
8	139	0.0015	0.0109
9	123	0.0013	0.0097
10	83	0.0009	0.0065
11	65	0.0007	0.0051
12	44	0.0005	0.0035
13	33	0.0003	0.0026
14	34	0.0004	0.0027
15	29	0.0003	0.0023
16	21	0.0002	0.0017
17	15	0.0002	0.0012
18	14	0.0001	0.0011
19	12	0.0001	0.0009
20	12	0.0001	0.0009

Table 8.3: The frequency distribution of chains sizes extracted by Reconcile from the FAPESP corpus. The third column presents the percentage of chains of a given size taking into consideration the singleton chains. The fourth column shows the numbers for when singleton chains are not taken into consideration.

chains is one (singleton chains). The proportion of singletons in the chains extracted by Reconcile is higher than the proportion of singletons in the NP4E manual annotation, 86.61 and 79%, respectively. The increase in the number of singletons in the extracted chains can be partly explained by the mistakes Reconcile does when clustering the mentions.

The greatest part of the chains extracted are formed by noun phrases that share the same word as the head of the phrase. Taking all the non-singleton chains and comparing all the expressions in the chain in a pairwise fashion, about 53% of the pairs share the same head. In figure 8.2, some examples of such chains are presented as well as some problematic chains found in the automatically

annotated corpus.

The chains (*a*) and (*c*) are examples of chains in which all the expressions have the same head and all of them actually refer to the same entity. Furthermore, both chains contain all the expressions that refer to their respective entities in their respective texts. The difference between the two is that in (*a*) the head word is a common noun whereas in (*c*) the head word is a proper noun.

The chain (*b*) is also an example of chain in which all the items have the same head word. However, the first expression in (*b*) does not refer to the same entity as the other two expressions that belong to the chain. The "basic logical operations" are a different set of operations than the "reversible logical operations" in the context of the text from where the chain was extracted. This is one example of a problematic chain extracted by Reconcile.

(*a*) ⟨ *the capacity*, *a greater capacity*, *capacity*, *the capacity* ⟩
(*b*) ⟨ *basic logical operations*, *reversible logical operations*, *these reversible operations* ⟩
(*c*) ⟨ *Unesco*, *Unesco*, *Unesco* ⟩
(*d*) ⟨ *the Environment*, *Britain*, *the region*, *the region*, *the region*, *the environment*, *the soil* ⟩
(*e*) ⟨ *Larry Ellison*, *Oracle's president* ⟩

Figure 8.2: Chains extracted from the English part of the FAPESP corpus using Reconcile.

Another example of problematic chain extracted is chain (*d*). In this chain, Reconcile seems to have merged different chains into one chain. The first expression, "The Environment" does not refer to the same entity as "Britain", "the region", and "the soil". All four expressions refer to different entities and they should belong to four different chains.

Despite using different types of features that collect information from different linguistic resources, Reconcile presents several problems in the quality of the extracted chains. The errors and inaccuracies of this initial annotation propagate to the subsequent steps and this have consequences on the final performance of the coreference resolver for Portuguese.

8.2 Aligment

In this section, the parameters used to run the tools during the alignment process are presented. Also, the results of the sentential alignment are presented.

For running the TCAlign system, the default anchor word list provided with the package of the program was expanded with 252 new anchor words. The most frequent pairs of English-Portuguese words in the corpus were computed using the output of the IBM Model 1 implemented by GIZA++. This model gives the probability of a pair of words being a translation of each other. Only pairs with a probability above 90% were considered.

The sentence alignment results are summarized on table 8.4. The most frequent alignment computed by TCAlign was the one-to-one mapping, about 92% of pairs of sentences.

Sentential Alignment Results		
Alignment Type	Frequency	%
1:1	16,942	92.02
1:2	209	1.13
2:1	139	0.75
0:1	210	1.14
1:0	914	4.96

Table 8.4: The frequency distribution of the types of sentential alignments processed by TCAlign.

As described in section 6.2, a filter is employed after the sentence alignment and only the one-to-one mappings are used in the word alignment. Therefore, not many alignments are lost due to the fact that most of them are one-to-one alignments.

The word aligner is configured to run the IBM models 1, 2, 3 and 4 with 5, 3, 5, and 5 iterations respectively. The word aligner is run in training mode over all the sentence aligned corpus. The training algorithm is run in both directions, English-Portuguese and Portuguese-English and the results are merged to improve the results as proposed by Och and Ney (2003).

8.3 Coreference Resolution for Portuguese

In this section the results for the coreference resolution system for Portuguese are presented and discussed. The methodology applied follows the rationale explained in chapter 4.

8.3.1 Instances generation and projection

For generating the pairs, the module that generates instances relies on the English noun phrases and coreference chains extracted by Reconcile. Applying the pairs generation algorithm presented in chapter 7, the instance generation module created 21,849 positive instance pairs and 436,033 negative instance pairs. The pairs generation results are presented in table 8.5.

Pairs Generation Results		
Pairs	Frequency	%
Positive	21,849	4.78
Negative	436,033	95.22
Total	457,882	100

Table 8.5: Number of pairs generated by the instances generation module.

The instance projection module uses the pairs generated for the English part of the corpus. Having the head of each expression that form the pair, it tries to find a corresponding expression in the Portuguese part of the corpus. As reported in section 8.1.2, 127,942 noun phrases were extracted by Reconcile. It is important to notice that a small portion of the English noun phrases do not have a head word assigned to them. About 3.79% of them (4,854) are missing heads due to problems in the algorithm Reconcile employs to recognize the phrases' heads. The noun phrase head is important because it is used to perform the projection of mentions.

Pairs Projection Results		
Pairs	Frequency	%
Positive	3,569	7.67
Negative	43,174	92.33
Total	46,543	100

Table 8.6: Number of pairs projected.

Table 8.6 presents the number of projected pairs. The projection module projected 3,569 positive pairs and 43,174 negative pairs. The proportion between negative and positive pairs increased after the projection. This increase is partly explained by the way the mentions are projected. The projection algorithm uses the heads, the sentences where the noun phrases occur in GIZA++ input files. In addition, errors may occur when searching for the mentions of the pairs in the GIZA++ input files. If any of this information is not available or if any of these processes fail, the algorithm fails to project the mention and several instance pairs are lost.

8.3.2 Classification

The JRip WEKA's implementation of decision rules was used to induce a classifier capable of assessing coreferent and not coreferent pairs. The JRip algorithm, was run with 10-fold cross-validation and default parameters. It has correctly classified 45,944 out of 46,743 instances (approximately 98%). Table 8.7) summarizes the classification results.

Classification Results				
Algorithm	# correct	%	# incorrect	%
JRip	45,944	98.29	799	1.71

Table 8.7: The accuracy results for the JRip classifier.

The classifier has more difficulties identifying the *Coref* class than the *Not Coref* class as is shown by the F-Measure of the *Coref* class in table 8.8.

JRip Class Accuracy			
Class	Precision	Recall	F-Measure
Coref	0.934	0.835	0.882
Not Coref	0.986	0.995	0.991

Table 8.8: The accuracy by class for the JRip classifier.

The JRip algorithm generated a classifier that contains only two rules. The rules are presented in figure 8.3. The first rule assigns the coreferent label to every instance that has a true value in the head match feature. The second rule assigns the instance to the coreferent class if a combination of five features occurs.

```
if (head_match = 1) => class=C
if (number_agrmt = 1) and (ant_appos = 1) and
   (sem_class_agrmt = 1) and (word_overlap = point5) and
   (ana_appos = 0) => class=C
else => class=NC
```

Figure 8.3: The rules generated by the JRip algorithm.

If the condition expressed in the second rule does not hold, the classifier assigns the instance to the non coreferent class.

Only five instances are assigned to the coreferent class in the second rule. All the other instances that reach this rule receive the non coreferent class label. Therefore, this set of rules is basically a classifier that assigns the instance to the coreferent class in case the feature `head_match` is true. Otherwise, the classifier assigns the instance to the non coreferent class. The classifier identified the `head_match` attribute as the most informative one to determine whether a given instance is coreferent or not. The reason can be two-fold: (i) the unbalanced dataset, with more instances of one class than the other, or, (ii) features that are not able to help the machine learning algorithm to divide the dataset into two classes.

To understand why JRip chose the `head_match` feature as the most informative feature, it is necessary to look into the projected training dataset and even into the chains extracted by Reconcile. The greatest part of the expressions that form the chains extracted by Reconcile have the same head wordface (about 53%) and this characteristic is propagated to the projected instances.

Analyzing the 3,569 coreferent pairs, it was concluded that 2,978 (approximately 83%) of them have the same head wordface (`hm` pairs). This leaves only 591 (17%) pairs that are positive but that do not have the same head wordface (`nhm` pairs). The small amount of `nhm` pairs is not enough to help the classifier infer more useful features than the `head_match` feature to describe them. Also, the JRip algorithm infers that not having a head match is highly informative in the case of the *Not Coref* class. This fact is explained by the big difference between the number of *Not Coref* samples (43,174) and the number of

nhm samples (591). The outcome is that the machine learning algorithm treats the nhm pairs as dataset noise and ignore them as relevant samples to infer important features from.

In order to see whether the features are informative enough to classify the nhm pairs, a small experiment with a balanced dataset was carried out. Random copies of the nhm instance pairs were inserted in the dataset until the number of *Not Coref* and nhm instances is similar. The classifier trained used other features than head_match and presented accuracy results in the order of 70%. Examples of features used are number_agrmt, sem_class_agrmt, and ant_appos. With this experiment it is possible to conclude that the higher number of non coreferent instances in the dataset makes the task of discovering the relevant features for the nhm pairs more difficult and also that the features used are relevant to the problem.

8.3.3 Clustering

The coreference resolution system employs the generated classifier in the clustering mechanism described in section 7.2. The output of the coreference resolution system for Portuguese was scored using the MUC and the CEAF scores (presented in section 3.2.4) over the Summ-It corpus. Table 8.9 summarizes the results.

Coreference Chains Evaluation			
Score	Precision	Recall	F-Measure
MUC	8.53	6.12	7.12
CEAF	18.06	11.93	14.37

Table 8.9: The MUC and CEAF scores for the coreference resolution system for Portuguese.

The MUC score presents F-Measure of 7.12 and the CEAF score presents F-Measure of 14.37. One of the reasons why the MUC score has a lower F-Measures is because it penalizes missed links. This is explained because several chains extracted, when compared to their reference chain, present only part of the expressions that appear in the reference chain.

One baseline was implemented in order to compare with the performance of the system. The baseline was implemented with the same clustering system but

with only one rule in the classification step: if the two expressions in the pair share the same head word, they are coreferent. Otherwise, they are not coreferent. In practice, the baseline classifier is the same as the classifier produced by JRip.

The baseline was run over the same corpus and the results obtained were identical to the system's results as expected. Due to the fact that both classifiers share the same classification rules, their results should be if not identical, very similar.

The chains extracted by the coreference resolution for Portuguese are very similar to the chains extracted by Reconcile. However, they do not present any pronouns because the Portuguese system only deals with noun phrases with common nouns and proper nouns. Additionally, chains with different head words were also not extracted because all the chains share the same head words. This is a consequence of the classification step. Figure 8.4 presents some of the chains extracted by the system.

(*a*) ⟨ *o MCT (Ministério da Ciência e Tecnologia)*, *o MCT* ⟩
(*b*) ⟨ *a domesticação em o novo continente*, *uma domesticação independente*, *a domesticação*, *a domesticação de cães domésticos* ⟩

Figure 8.4: Chains extracted by the coreference resolution system for Portuguese.

Several chains extracted captured only part of the expressions that the chain should contain. One example is chain (*a*). The manually annotated chain contains four elements and the extracted one has only two: "the MCT (Ministry for Science and Technology)" and "the MCT". It should also contain "*a Ciência e Tecnologia*" ("the Science and Technology") and "*(Ciência e Tecnologia)*" ("(Science and Technology)"). As the two missing expressions do not share the same head word with the extracted mentions, the system was not able to cluster them in the same chain.

Some other chains extracted present more expressions than they should have, as is the case of chain (*b*) in figure 8.4. The manual annotation for the chain in (*b*) is formed by three expressions whereas the chain extracted contains four expressions. The coreference system included the fourth expression that probably is not coreferent with the entity represented by this chain. One interesting characteristic of the manual annotated chain is that all its three items have the same expressions: "*a domesticação*" ("the domestication"). In this example there is some error propagated by the program used to obtain the noun phrase

boundaries. The automatic noun phrase identification is not as precise as the identification made in the manual annotation and this leads the coreference resolution system to errors when classifying and clustering the pairs.

The results of the Portuguese coreference resolution are identical to the baseline and inferior to results achieved by other system that was tested in the same corpus with reported MUC score F-Measure of 51%(Souza et al., 2008). For the methodology presented in this text, it is important to remark that the accumulated error of the whole process influences the final step of the pipeline and this contributes in a negative way to the final results.

Chapter 9

Final Remarks

This dissertation presented a system with which is possible to extract coreference chains from Portuguese texts without having to resort to Portuguese corpora manually annotated with coreferential data. The system implements a method that automatically obtains data for training a supervised machine learning coreference resolver for Portuguese. The training data is generated by using an English-Portuguese parallel corpus over which the coreference chains annotated in the English part of the corpus are projected to the Portuguese part of the corpus. The coreference chains extraction system for Portuguese was tested in a corpus annotated with coreference chains in Portuguese. The results of running the system over this corpus are comparable to the baseline.

The whole process described in the methodology has a strong influence of the coreference annotation made for the English side of the parallel corpus. The errors generated in this step are propagated throughout the pipeline. Therefore, the use of a better performing coreference resolution system for annotating the English side of the corpus might improve the overall performance of the system.

Furthermore, the projection of mentions showed to have influence on the coreference resolution. Different methods for performing the projection might be implemented and tested. The method described in this dissertation is a good starting point to build upon. As future work, an evaluation of the projected pairs should be carried out in order to evaluate the the strong points and the pitfalls of the algorithm employed. A portion of the Portuguese side of the FAPESP corpus was annotated during the development of this dissertation but the process of evaluation could not be completed. With this annotation it will be possible to

take each chain projected and compare with its corresponding manual projection.

Finally, the main contribution of this work is to lay out a methodology for coreference resolution for Portuguese that does not rely on Portuguese corpora manually annotated with coreferential data. However, the implementation of the methodology presented in this dissertation needs further work in order to have results better than the baseline.

References

Aires, A. M., Coelho, J. C. B., Collovini, S., and Quaresma, P. (2004). Avaliação de Centering em Resolução Pronominal da Língua Portuguesa. In *5th International Workshop on Linguistically Interpreted Corpora of the Iberamia'2004*, Puebla, México.

Aone, C. and Bennett, S. W. (1995). Evaluating automated and manual acquisition of anaphora resolution strategies. *Proceedings of the 33rd annual meeting on Association for Computational Linguistics* -, pages 122–129.

Aziz, W., Pardo, T. A. S., and Paraboni, I. (2008). Building a Spanish-Portuguese parallel corpus for statistical machine translation. In *Companion Proceedings of the XIV Brazilian Symposium on Multimedia and the Web*, pages 369–371, New York, New York, USA. ACM.

Aziz, W., Pardo, T. A. S., and Paraboni, I. (2009). Statistical Phrase-based Machine Translation: Experiments with Brazilian Portuguese. In *XXIX CSBC VII Encontro Nacional de Inteligência Artificial (ENIA-2009)*, pages 769–778.

Bagga, A. and Baldwin, B. (1998). Algorithms for scoring coreference chains. *Proceedings of the Linguistic Coreference Workshop at The First International Conference on Language Resources and Evaluation (LREC'98)*, pages 563–566.

Bean, D. and Riloff, E. (2004). Unsupervised Learning of Contextual Role Knowledge for Coreference Resolution. In *Proc. of HLT/NAACL*, pages 297–304.

Bengtson, E. and Roth, D. (2008). Understanding the value of features for coreference resolution. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, (October):294–303.

Berger, A. L., Pietra, S. A. D., and Pietra, V. J. D. (1996). A maximum entropy approach to natural language. *Computational Linguistics*, 22(1):39–71.

Bick, E. (2000). *The parsing system PALAVRAS: automatic grammatical analysis of Portuguese in a constraint grammar framework*. Phd, Arhus.

Brown, P., Pietra, S., Pietra, V., and Mercer, R. (1994). The mathematic of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311.

Cardie, C. and Wagstaff, K. (1999). Noun phrase coreference as clustering. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, number 1995, pages 82–89. Association for Computational Linguistics.

Caseli, H. D. M. (2003). *Alinhamento sentencial de textos paralelos português-inglês*. Master thesis, USP.

Caseli, H. D. M. (2007). *Indução de léxicos bilíngües e regras para a tradução automática Indução de léxicos bilíngües e regras para a tradução automática*. Phd, USP.

Caseli, H. D. M., da Paz Silva, A. M., and Nunes, M. d. G. V. (2004). Evaluation of Methods for Sentence and Lexical Alignment of Brazilian Portuguese and English Parallel Texts. In *Proceedings of the XVII Brazilian Symposium on Artificial Intelligence (SBIA)*, pages 184–193.

Chang, C.-C. and Chih-Jen, L. (2001). LIBSVM: a library for support vector machines.

Chaves, A. (2007). *A resolução de anáforas pronominais da Língua Portuguesa com base no algoritmo de Mitkov*. Master thesis, Universidade Federal de São Carlos.

Chaves, A. and Rino, L. (2008). The Mitkov Algorithm for Anaphora Resolution in Portuguese. *Proceedings of the 8th international conference on Computational Processing of the Portuguese Language*, page 60.

Coelho, T. T. (2005). *Resolução de anáfora pronominal em português utilizando o algoritmo de Lappin e Leass*. Masters, Unicamp.

Cohen, W. (1995). Fast effective rule induction. In *12th International Workshop Conference on Machine Learning*, pages 115–123. Morgan Kaufmann Publishers, Inc.

Collovini, S., Carbonel, T. I., Fuchs, J. T., and Vieira, R. (2007). Summ-it: Um corpus anotado com informacoes discursivas visando à sumarizacao automática. In *TIL - V Workshop em Tecnologia da Informação e da Linguagem Humana*, pages 1605–1614, Rio de Janeiro.

Collovini, S. and Vieira, R. (2006a). Anáforas nominais definidas : balanceamento de corpus e classificação. In *Proceedings of the International Joint Conference IBERAMIA/SBIA/SBRN 2006 - 4th Workshop in Information and Human Language Technology (TIL'2006)*, page 10, Ribeirão Preto, Brazil.

Collovini, S. and Vieira, R. (2006b). Learning Discourse-new References in Portuguese Texts. In *TIL 2006*, pages 267–276.

Connolly, D., Burger, J. D., and Day, D. S. (1997). A machine learning approach to anaphoric reference. In *New Methods in Language Processing*, pages 133–144.

Cuevas, R., Honda, W., de Lucena, D., Paraboni, I., and Oliveira, P. (2008). Portuguese Pronoun Resolution: Resources and Evaluation. *Lecture Notes in Computer Science*, 4919:344.

Cuevas, R. and Paraboni, I. (2008). A Machine Learning Approach to Portuguese Pronoun Resolution. *Proceedings of the 11th Ibero-American conference on AI: Advances in Artificial Intelligence*, pages 262–271.

Denis, P. (2007). *New Learning Models for Robust Reference Resolution*. Phd.

Denis, P. and Baldridge, J. (2007). Global, Joint Determination of Anaphoricity and Coreference Resolution using Integer Programming. *In HLT-NAACL*.

Denis, P. and Baldridge, J. (2008). Specialized models and ranking for coreference resolution. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, number October, pages 660–669. Association for Computational Linguistics.

Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts.

Finkel, J. R., Grenager, T., and Manning, C. D. (2005). Incorporating non-local information into information extraction systems by Gibbs sampling. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, (June):363–370.

Gale, W. and Church, K. (1993). A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102.

Gildea, D. and Jurafsky, D. (2002). Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245–288.

Haghighi, A. and Klein, D. (2007). Unsupervised coreference resolution in a nonparametric bayesian model. *Annual Meeting-Association for Computational Linguistics*, 45(1):848.

Haghighi, A. and Klein, D. (2009). Simple Coreference Resolution with Rich Syntactic and Semantic Features. In *EMNLP '09: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, number August, pages 1152–1161, Singapore. Association for Computational Linguistics.

Hall, M., National, H., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software : An Update. *SIGKDD Explorations*, 11(1):10–18.

Halliday, M. A. K. and Hasan, R. (1976). *Cohesion in English*. Longman, New York, 1 edition.

Harabagiu, S., Bunescu, R., and Maiorano, S. (2001). Text and Knowledge Mining for Coreference Resolution. *North American Chapter Of The Association For Computational Linguistics*, pages 1–8.

Hasler, L., Orasan, C., and Naumann, K. (2006). NPs for events: Experiments in coreference annotation. *Proceedings of the 5th edition of the International Conference on Language Resources and Evaluation (LREC2006)*, pages 1167–1172.

Hofland, K. (1996). A program for aligning English and Norwegian sentences. In Hockey, S., Ide, N., and Perissinotto, G., editors, *Research in Humanities Computing*, number July, pages 165–178. Oxford University Press.

Hoste, V. and Pauw, G. D. (2006). KNACK-2002: a Richly Annotated Corpus of Dutch Written Text. In *Proceedings of The fifth international conference on Language Resources and Evaluation*, pages 1432–1437. ELRA.

Joachims, T. (2002). *Learning to Classify Text Using Support Vector Machines*. Kluwer Academic Publishers / Springer.

Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics*. Prentice-Hall, 2nd edition.

Kehler, A. (1997). Probabilistic coreference in information extraction. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 163–173. Association for Computational Linguistics.

Klein, D. and Manning, C. D. (2003). Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*, pages 3–10.

Koch, I. G. V. (2002). *Coesão Textual*. Editora Contexto, 17 edition.

Lappin, S. and Leass, J. (1994). An Algorithm for Pronominal Anaphora Resolution. *Computational Linguistics*, 20(4):535–561.

Luo, X. (2005). On coreference resolution performance metrics. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32.

Luo, X. (2007). Coreference or Not : A Twin Model for Coreference Resolution. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, number April, pages 73–80, Rochester, New York. Association for Computational Linguistics.

Luo, X., Ittycheriah, A., Jing, H., Kambhatla, N., and Roukos, S. (2004). A Mention-Synchronous Coreference Resolution Algorithm Based On the Bell Tree. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 135–142, Barcelona, Spain.

Markert, K., Nissim, M., and Modjeska, N. N. (2003). Using the Web for Nominal Anaphora Resolution. In *EACL Workshop on the Computational Treatment of Anaphora*, number 2, pages 39–46.

McCallum, A. and Wellner, B. (2004). *Conditional models of identity uncertainty with application to noun coreference*. MIT Press, Cambridge, Massachusetts.

McCarthy, J. F. and Lehnert, W. G. (1995). Using Decision Trees for Coreference Resolution. In *International Joint Conference on Artificial Intelligence*, pages 1050–1055.

Miller, G. A. (1995). WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Mitkov, R. (2002). *Anaphora Resolution*. Cambridge University Press.

Ng, V. (2002). Machine learning for coreference resolution: Recent successes and future challenges. Technical report, Cornell University, Ithaca.

Ng, V. (2004). Learning noun phrase anaphoricity to improve coreference resolution: Issues in representation and optimization. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 151. Association for Computational Linguistics.

Ng, V. (2005). Machine learning for coreference resolution: From local classification to global ranking. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, number June, pages 157–164, Ann Arbor, Michigan. Association for Computational Linguistics.

Ng, V. (2007a). Semantic Class Induction and Coreference Resolution. In *Proceedings of ACL 2007*, number June, pages 536–543. Association for Computational Linguistics.

Ng, V. (2007b). Shallow Semantics for Coreference Resolution. In *Proceedings of IJCAI 2007*, pages 1689–1694.

Ng, V. (2010). Supervised Noun Phrase Coreference Research : The First Fifteen Years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, number July, pages 1396–1411.

Ng, V. and Cardie, C. (2002a). Combining sample selection and error-driven pruning for machine learning of coreference rules. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02*, (July):55–62.

Ng, V. and Cardie, C. (2002b). Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 104–111, Philadelphia, United States.

Nicolae, C. and Nicolae, G. (2006). BESTCUT: A Graph Algorithm for Coreference Resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, number July, pages 275–283, Sydney, Australia. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 440–447. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Paraboni, I. (1997). *Uma arquitetura para a resolução de referências pronominais possessivas no processamento de textos em Língua Portuguesa*. Masters thesis, PUC-RS.

Paraboni, I. and Lima, V. L. S. D. (1998). Possessive Pronominal Anaphor Resolution in Portuguese Written Texts - Project Notes. In *17th International Conference on Computational Linguistics (COLING-98)*, pages 1010–1014, Montreal, Quebec, Canada. Morgan Kaufmann Publishers.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, (July):433–440.

Petrov, S. and Klein, D. (2007). Improved inference for unlexicalized parsing. *Proceedings of NAACL HLT 2007*, (April):404–411.

Ponzetto, S. P. and Strube, M. (2006a). Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 33(June):192–199.

Ponzetto, S. P. and Strube, M. (2006b). Semantic role labeling for coreference resolution. *Proceedings of the Eleventh Conference of the*, pages 143–146.

Popescu-Belis, A. and Robba, I. (1998). Three New Methods for Evaluating Reference Resolution. In *Proceedings of the LREC'98 Workshop on Linguistic Coreference*, pages 43–48, Granada, Spain.

Postolache, O., Cristea, D., and Orasan, C. (2006). Transferring Coreference Chains through Word Alignment. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, Genoa, Italy.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Recasens, M. and Hovy, E. (2009). A deeper look into features for coreference resolution. *Anaphora Processing and Applications*, (i):29–42.

Recasens, M., Màrquez, L., Sapena, E., Martí, M. A., Taulé, M., Hoste, V., Poesio, M., and Versley, Y. (2010). SemEval-2010 Task 1 : Coreference Resolution in Multiple Languages. *Computational Linguistics*, (July):1–8.

Recasens, M. and Martí, M. A. (2009). AnCora-CO: Coreferentially annotated corpora for Spanish and Catalan. *Language Resources and Evaluation*, 44(4):341–345.

Rose, T., Stevenson, M., and Whitehead, M. (2002). The Reuters Corpus Volume 1-from yesterday's news to tomorrow's language resources. In *Proceedings of the Third International Conference on Language Resources and Evaluation*, volume 1, pages 29–31. Citeseer.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2 edition.

Santos, D. and Oksefjell, S. (2000). An evaluation of the Translation Corpus Aligner, with special reference to the language pair English-Portuguese. *T Nordgård, T. ed., NODALIDA*, 99:191–205.

Santos, D. N. D. A. (2008). *Resolução de anáfora pronominal em português utilizando o algoritmo de Hobbs*. Masters, Unicamp.

Soon, W. M., Ng, H. T., and Lim, D. C. Y. (2001). A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Computational Linguistics*, 27(4):521–544.

Souza, J. G. C. D., Gonçalves, P. N., and Vieira, R. (2008). Learning Coreference Resolution for Portuguese Texts. In Teixeira, A., Lima, V. L. S. D., Oliveira, L. C. D., and Quaresma, P., editors, *Computational Processing of the Portuguese Language - 8th International Conference, PROPOR 2008*, pages 153–163, Aveiro, Portugal. Springer Berlin / Heidelberg.

Stoyanov, V., Cardie, C., Gilbert, N., and Buttler, D. (2010). Coreference Resolution with Reconcile. In *Proceedings of the Joint Conference of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*. Association for Computational Linguistics.

Stoyanov, V., Cardie, C., Gilbert, N., Riloff, E., and Hysom, D. (2009a). Reconcile : A Coreference Resolution Research Platform. Technical report, Lawrence Livermore National Laboratory (LLNL), Livermore, United States.

Stoyanov, V., Gilbert, N., Cardie, C., and Riloff, E. (2009b). Conundrums in Noun Phrase Coreference Resolution : Making Sense of the State-of-the-Art. In *ACL-IJCNLP '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2*, number August, pages 656–664, Suntec, Singapore. Association for Computational Linguistics.

Strube, M., Rapp, S., and Müller, C. (2002). The influence of minimum edit distance on reference resolution. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02*, (July):312–319.

Uryupina, O. (2004). Linguistically Motivated Sample Selection for Coreference Resolution. In *Proceedings of DAARC-2004*, Furnas.

Véronis, J. and Langlais, P. (2000). *Evaluation of parallel text alignment systems - The ARCADE project*, pages 369–388. Kluwer Academic Publishers.

Versley, Y., Ponzetto, S. P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., and Moschitti, A. (2008). BART : A Modular Toolkit for Coreference Resolution. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, number 2006, pages 1–4.

Vieira, R. (1998). *Definite description processing in unrestricted text*. Phd, University of Edinburgh.

Vieira, R., Gonçalves, P. N., and de Souza, J. (2008). Processamento computacional de anáfora e correferência. *Revista de Estudos da Linguagem*, 16(1):263–284.

Vilain, M., Burger, J., Aberdeen, J., and Connolly, D. (1995). A model-theoretic coreference scoring scheme. *Proceedings of the 6th conference on Message understanding*, pages 45–52.

Vogel, S., Ney, H., and Tillmann, C. (1996). HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841, Morristown, NJ, USA. Association for Computational Linguistics.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann.

Yang, X., , Zhou, G., Su, J., and Tan, C. L. (2003). Coreference Resolution Using Competition Learning Approach. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 176–183, Sapporo, Japan. Association for Computational Linguistics.

Yang, X. and Su, J. (2007). Coreference Resolution Using Semantic Relatedness Information from Automatically Discovered Patterns. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, number

June, pages 528–535, Prague, Czech Republic. Association for Computational Linguistics.

Yang, X., Su, J., Lang, J., Tan, C. L., Liu, T., and Li, S. (2008). An Entity-Mention Model for Coreference Resolution with Inductive Logic Programming. In *Proceedings of ACL-08: HLT*, number June, pages 843–851, Columbus, Ohio. Association for Computational Linguistics.