Anaphora Resolution

Nuno Ricardo Pedruco Nobre

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Júri

Presidente: Professor Doutor Joaquim Armando Pires Jorge
Orientador: Professor Doutor Nuno João Neves Mamede
Co-Orientador: Professor Doutor Jorge Manuel Evangelista Baptista
Vogais: Professor Doutor Bruno Emanuel da Graça Martins

Maio 2011
I would like to thank Professor Nuno João Neves Mamede and Professor Jorge Manuel Evangelista Baptista for their valuable guidance. Their knowledge was an inestimable contribution to the conclusion of this work.

I would also like to thank Ana, Hugo, João, João, Pedro, Patrícia, Tiago and Ricardo, not only for their assistance in completing this work but also by the constant presence in these years. They are inextricably connected with my academic career and life.

Last but certainly most important, I would like to thank my parents for their unconditional support.

Lisboa, Maio 2011

Nuno Ricardo Pedruco Nobre
Aos meus pais.
Este documento analisa e compara algumas abordagens para a Resolução de Anáfora e descreve uma solução baseada no algoritmo de Mitkov adaptado à Língua Portuguesa. A solução desenvolvida propõe-se a resolver anáforas pronominais, nomeadamente os pronomes pessoais e possessivos de terceira pessoa, pronomes relativos e demonstrativos, utilizando um conjunto de parâmetros para determinar os respectivos antecedentes. Durante o desenvolvimento foi criada uma ferramenta de anotação manual, que permite o enriquecimento de forma rápida de textos com informação anafórica. O sistema apresentou na avaliação final uma medida-f de 33.5%.
This document analyses and compares some approaches to the Anaphora Resolution task and describes a Mitkov algorithm based solution adapted to the Portuguese Language. The developed system proposes to resolve pronominal anaphora, namely third person, personal and possessive pronouns, relative and demonstrative pronouns. During the system development a manual annotation tool was created, allowing to enrich text with anaphoric information on a quick way. The system presented an f-measure of 33.5%.
Palavras-chave

Resolução de Anáfora
Resolução Pronominal
Sistemas de processamento de Linguagem Natural
Extracção de informação

Keywords

Anaphora Resolution
Pronominal Resolution
Natural Language processing systems
Information retrieval
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Motivation</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Cohesion and Coreference</td>
<td>1</td>
</tr>
<tr>
<td>1.3</td>
<td>Forms of anaphora</td>
<td>2</td>
</tr>
<tr>
<td>1.3.1</td>
<td>Pronominal anaphora</td>
<td>3</td>
</tr>
<tr>
<td>1.3.2</td>
<td>Lexical noun phrase anaphora</td>
<td>4</td>
</tr>
<tr>
<td>1.3.3</td>
<td>Noun anaphora</td>
<td>6</td>
</tr>
<tr>
<td>1.3.4</td>
<td>Verb anaphora</td>
<td>6</td>
</tr>
<tr>
<td>1.3.5</td>
<td>Adverb anaphora</td>
<td>7</td>
</tr>
<tr>
<td>1.3.6</td>
<td>Zero anaphora</td>
<td>7</td>
</tr>
<tr>
<td>1.3.7</td>
<td>Intrasentential and intersentential anaphora</td>
<td>8</td>
</tr>
<tr>
<td>1.4</td>
<td>Anaphora resolution</td>
<td>8</td>
</tr>
<tr>
<td>1.4.1</td>
<td>Anaphora Resolution Factors</td>
<td>9</td>
</tr>
<tr>
<td>1.5</td>
<td>Dissertation Structure</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>Related Work</td>
<td>13</td>
</tr>
<tr>
<td>2.1</td>
<td>Introduction</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Statistical approaches</td>
<td>13</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Collocation patterns-based approach</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Machine learning approaches</td>
<td>15</td>
</tr>
</tbody>
</table>
List of Figures

2.1 Evaluation measures ............................................................ 14

3.1 $L^2F$ XIP processing chain .................................................. 28

3.2 Syntactic structure of the PP, *da João* ..................................... 31

3.3 Interaction between ARM and XIP Processing Chain ...................... 33

3.4 ARM output ........................................................................ 35

4.1 XIP chunk tree for example (4.5) ............................................. 41

4.2 XIP dependencies for example 4.5) ........................................... 41

4.3 XIP API domain ................................................................... 43

4.4 Configuration file structure ....................................................... 44

4.5 Anaphora Resolution Module Domain ......................................... 45

4.6 Anaphora Resolution Module execution ....................................... 47

5.1 Anaphora Manual Annotator ..................................................... 50
List of Tables

2.1 Co-occurrence patterns associated with the verb collect based on an excerpt from
the Hansard corpus .................................................... 15
2.2 RAPM overall assessment ........................................... 24
2.3 Systems overview: features ........................................... 25
2.4 Systems overview: evaluation ....................................... 25
4.1 Pronoun gender for coordinated NPs .............................. 39
1

Introduction

1.1 Motivation

In our daily conversation we use several linguistic mechanisms that provide better understanding. It is the aim of Natural Language Processing (NLP) to recognise those mechanisms, while producing intelligible and coherent information. The scope of this work, Anaphora Resolution, is the identification of a word or a string of words that functions as a regular grammatical substitute for a preceding word or a string of words.


Obama was awarded the Nobel Peace. The President of the United States was the winner of the this year Nobel Peace Prize.

In the previous example, O Presidente dos Estados Unidos is the anaphor and it refers to Obama that is its antecedent. The referential relation between the anaphor and its antecedent is called anaphora. The task of identifying the anaphoric relation between these elements is called anaphora resolution. Such is an important task since it allows the enrichment of obtained text information, by relating words and creating anaphoric chains. It is the goal of the present work to produce a system capable of performing such task.

Next it will be introduced a few notions that allow us to filter antecedent candidates for a known anaphor.

1.2 Cohesion and Coreference

Usually communication between people is coherent, meaning that a person does not transmit isolated and independent sentences. This is stated as cohesion [14].
(1.2) O Prémio Pessoa 2009 foi atribuído a D. Manuel Clemente, revelou o júri, reunido no Palácio de Seteais, em Sintra. Ele é o primeiro homem da Igreja a receber este prémio.

Pessoa Prize 2009 was awarded to D. Manuel Clemente, said the jury, gathered at Palácio de Seteais, Sintra. He is the first man of the Church to receive this award.

In example (1.2), although we do not have such explicit information, we can assume that the second sentence is related to the first one and that Ele refers to D. Manuel Clemente, and also that prémio refers to Prémio Pessoa.

When the anaphor and its antecedent have the same referent in the real world, like the previous examples, they are termed coreferential. Consider the following example:

(1.3) A obra de Chico Buarque está disponível na internet. Imagens raras e gravações do artista estão disponíveis no site do Instituto Tom Jobim.

The work of Chico Buarque is available on the Internet. Rare footage and recording of the artist are available at Instituto Tom Jobim’s website.

In example (1.3), the noun artista is the anaphor, the name Chico Buarque is the antecedent and there is a coreferential relation between both, as they refer to a real world person, the brazilian artist Chico Buarque.

Next, example (1.4) give us an example where coreference is not observed:

(1.4) Os homens são como as moedas; devemos tomá-los pelo seu valor, seja qual for o seu cunho.

Men are like coins, we should take them by their value, not their stamp.

Although both occurrences of pronoun seu and pronoun los are the anaphors of homens, they are not coreferential, as they do not refer to the same entity in the real world.

1.3 Forms of anaphora

This section, based on Mitkov work [14], presents the several existing types of anaphora.
1.3. FORMS OF ANAPHORA

1.3.1 Pronominal anaphora

According to Mitkov, this is the most common type of anaphora and occurs when the anaphor is a pronoun.

(1.5) Hoje estive com a Ana e o namorado dela.

Today I met Ana and her boyfriend.

In example (1.5) ela is the anaphor and Ana its antecendent.

Personal, possessive and demonstrative pronouns both singular and plural can function as pronominal anaphora. First and second person pronouns, singular or plural usually refer to the dialog interlocutors, thus these pronouns do not establish a coreference relation between elements present in the analysed sentences.

Indefinite and interrogative pronouns also do not function as anaphors. Indefinite pronouns, like muito, outro, algum (most, other, some), have an indefinite referent. The same happens with interrogative pronouns, e.g. quem, quanto (who, how much), whose indefinite referent is the scope of the question or indirect subclause they introduce.

In example (1.6) Alguém is an indefinite pronoun and in (1.7) Quem is an interrogative pronoun. For these two types of pronouns no antecedent is referred to, therefore are not classified as anaphors.

(1.6) Alguém bateu à porta.

Someone knocked on the door.

(1.7) Quem está aí?

Who is there?

Relative pronouns, on the other hand, have as their antecedent the immediate noun, being modified by the relative subclause. Because of this, they have an explicit short distanced antecedent. A special case of relatives, apparently having no antecedent, can be analysed as modifying an indefinite pronoun or a headless noun phrase (NP), i.e. whose head as been zeroed. These cases were not considered in this work.
Examples (1.8) and (1.9) present two non-anaphoric occurrences of pronoun que. In (1.8), que is an indefinite pronoun. In (1.9) que is a relative pronoun in a sentence where the head of the first NP was zeroed. In this example, if the head casa was present it would be the antecedent of the anaphor que.

(1.8) O que me incomoda é isso.
That is what troubles me.

(1.9) A (casa) que prefiro é esta.
(The house) Which I prefer is this.

In fact, knowing that a word is a pronoun is not enough to determine whether it is an anaphor or not. This comes from the fact that anaphora is a syntactic phenomenon instead of a morphologic one. A pronoun is defined as a word that can be the head of phrase, but some times, when appears with a noun at its right, it plays the role of a determinant [1].

(1.10) Esse livro é muito bom.
That book is very good.

In example (1.10) esse (that) is a demonstrative pronoun and it functions as the (demonstrative) determinant of book. Because of this, no anaphoric relation should be established between esse and another noun.

1.3.2 Lexical noun phrase anaphora

This form of anaphora occurs as definite noun phrases and proper names.

(1.11) Cavaco regozija-se com a candidatura de Constância à vice-presidência do BCE. O Presidente da República Portuguesa afirmou que “Portugal ficaria muito bem representado se o governador do Banco de Portugal, Vítor Constânio, viesse a ser escolhido para o cargo de vice-presidente do Banco Central Europeu”.
Cavaco welcomes the nomination of Constâncio to the vice-presidency of the ECB. The President of the Portuguese Republic said "Portugal would be very well represented the governor of the Bank of Portugal, Vitor Constâncio, were to be chosen for the post of vice-president of the European Central Bank.

In the previous example the definite NP O Presidente da República Portuguesa is the anaphor and the proper name Cavaco the antecedent.

Usually this form of anaphora adds more information to the sentence, and increases the cohesiveness. One gets to know that Cavaco is the The President of the Republic. Lexical noun phrase anaphora may appear in several forms:

- it can have the same head as the antecedent:

(1.12) Hoje comi um bolo ao pequeno almoço. Aquele bolo estava mesmo saboroso.

Today I ate a cake at breakfast. That cake was really tasty.

The NP Aquele bolo is the anaphor for um bolo.

- it may be in the form of a synonym. In this case the antecedent is substituted by a word with similar meaning:

(1.13) O polícia mandou parar o automóvel e pediu ao condutor que saísse da viatura.

The police ordered the car to stop and asked the driver to leave the vehicle.

- in the form of generalisation/hypernymy:

(1.14) Lisboa recebe três "Óscars do turismo". A capital portuguesa recebeu três prémios na décima sexta edição do World Travel Awards.

Lisbon receives three "Oscars of tourism”. The Portuguese capital has received three awards in the sixteenth edition of the World Travel Awards.

In example (1.14) we can observe a generalization, as the antecedent Lisbon is referred again by the the capital of Portugal. Being a proper name, one can consider that the first term is most specific then the latter, which only states a property of the city.
- or specialisation/hyponymy:

(1.15) *Portuguesa* conquista medalha de ouro de judo. A *Ana Hormigo* derrotou na final a italiana Valentina Moscatt.

*Portuguese woman* wins gold medal in judo. *Ana Hormigo* beat at the final the italian contestant Valentina Moscatt.

Example (1.15) show us an hyponymy, as the anaphor NP, *A Ana Hormigo*, is a particular case of the antecedent *Portuguesa*.

### 1.3.3 Noun anaphora

Noun anaphora is the anaphoric relation between a name and a noun phrase. Bellow, in example (1.16) the NP *O gato* is substituted by the noun *Pantufas*.

(1.16) *O gato* está a dormir. *Pantufas*, como lhe chamam os donos, gosta de longas sestas.

*The cat* is sleeping. *Pantufas*, as the owners call him, enjoys long naps.

### 1.3.4 Verb anaphora

Verb anaphora occurs when a verb anaphor has a verb or verb phrase (VP) as an antecedent.

(1.17) Assim que o pai disse à criança para *parar de correr* ela *fê-lo*.

As the father told the child to *stop running* it *did so*.

Above, the anaphor *fê-lo* refers to the VP *parar de correr*. In this case, the verb *fazer* (*to do*) is called a *pro-verb*, as it refers to a verb in the previous discourse.
1.3. FORMS OF ANAPHORA

1.3.5 Adverb anaphora

In example (1.18), below, the anaphor *here* refers to the antecedent *Lisbon*.

\[(1.18) \text{Eu nasci em Lisboa e vivo aqui desde sempre.}\]

I was born in *Lisbon* and live *here* since ever.

1.3.6 Zero anaphora

Also called as ‘invisible’ anaphor [14], this form of anaphora does not involve an explicit word or a phrase. In fact, it occurs in sentences where words of phrases have been zeroed. This reduction avoids the repetition of correferent words, making the discourse simple and enhancing communication. Reduction does not affect the information conveyed by discourse and the meaning of the sentences is left unchanged by this zeroing.

It is possible to point out the four most common types of zero anaphora:

- Zero pronominal anaphora:

\[(1.19) \text{O António está com sono. } \emptyset \text{ Esteve acordado a noite toda.}\]

António is sleepy. (He) Was up all night.

The pronoun *ele* was omitted but we can still understand that the subject of *Esteve acordado a noite toda* was António.

- Zero noun anaphora occurs when the head noun of an NP is omitted; usually a determinant is left in place, to establish the anaphoric relation.

\[(1.20) \text{Se viesse mais cedo ainda haveria pão na padaria, agora não há nenhum } \emptyset.\]

If I had came earlier there should still be any bread in the bakary, now there are none.

In example (1.20), we consider *nenhum* as a determinant of the zeroed instance of *pão* (*bread*). In traditional grammar *nenhum* is considered a pronoun in ‘adjectival’ use when determining the noun in a fully pledged NP: *nenhum pão* (*no bread*); in this example, *nenhum* is said to be in its pronominal use.
- Zero verb anaphora arises when the verb is omitted and the antecedent is a verb in the previous sentence:

\[(1.21)\] O Pedro \textit{ganhou} um carro e a Ana \emptyset uma viagem.

Pedro won a car and Ana \emptyset a trip.

The zero verb anaphor \emptyset refers to the verb \textit{ganhar}.

- Verb phrase zero anaphora:

This form of anaphora is also termed \textit{ellipsis} and is the omission of a verb phrase which antecedent is a verb phrase in a previous sentence.

\[(1.22)\] O Pedro queria \textit{ir a Lisboa} mas a Ana não podia \emptyset.

Pedro wanted to \textit{go} to Lisbon, but Ana could not \emptyset.

The zero verb phrase anaphor stands for the verbal phrase \textit{ir a Lisboa}.

1.3.7 Intrasentential and intersentential anaphora

Anaphora can also be classified as intrasentential or intersentential, according to the location of the antecedent. If the anaphor and its antecedent occur in the same sentence, like in examples (2.21) and (2.22), it is called intrasentential anaphora. If they are located in different sentences, like in example (2.19), it is called intersentential anaphora.

1.4 Anaphora resolution

The process of anaphora resolution usually follows three steps:

1. anaphor identification;
2. antecedent candidates identification;
3. choosing the most likely antecedent candidate.
As this work will only focus on the analysis and resolution of the *pronominal anaphora*, the identification strategy can be simplified and confined to *anaphoric pronouns*.

Antecedent candidates identification will be made in the two or three sentences preceding the anaphor. This option is based in the fact that many pronominal anaphora resolution approaches [13] use this scope with satisfactory results (see Chapter 2).

Once the anaphors and their antecedent candidates are located, it is time to choose the most likely candidate. The next section will introduce some frequently used anaphora resolution factors that enable this choice.

### 1.4.1 Anaphora Resolution Factors

**Gender and number agreement**

Both the anaphor and antecedent must agree in number and gender.

(1.23) *Marta e Ana foram ao centro comercial. Elas estiveram lá a tarde toda.*

*Marta and Ana went to the mall. They were there all afternoon.*

On the above example it is possible to identify two anaphors in the second sentence: the feminine third-person plural personal pronoun *Elas* and the locative anaphoric adverb *lá*. By analyzing the previous sentence, two NPs (*Marta e Ana* and *o centro comercial*) are identified as possible antecedent candidates. By making use of gender and number agreement factor one can identify *Marta e Ana* as the antecedent of the pronominal anaphor *Elas* and *lá* as an adverb anaphor having *ao centro comercial* as its antecedent.

Since Portuguese nouns and pronouns are often explicitly marked for gender and number, this factor acquires great importance in the anaphora resolution process.

**Selectional restriction**

This factor is also referred as semantic restriction. If a selectional restriction is applied to an anaphor it should also be applied to its antecedent. Consider the next examples:
(1.24) O Pedro tirou um lápis do estojo e afiou-o.

Peter took *a pencil* from the case and honed it.

(1.25) O Pedro tirou um lápis do *estojo* e fechou-o.

Peter took a pencil from *the case* case and closed *it*.

In the previous examples, the semantical restriction applied to the anaphoric pronoun *o* must also be applied to its antecedents. In spite of the fact that there are 3 masculine singular antecedent candidates for pronoun *o* (v.g. *Pedro, lápis* and *estojo*), *Pedro* is discarded for being a human noun; while both *lápis* and *estojo* are non-human nouns, only one of them can adequately correspond to the distributional constraints imposed by the verbs *afiar (to hone)* and *fechar (to close)*, respectively. Hence, in (2.24) it is possible to hone the pencil, therefore, the antecedent is *um lápis*. In example (2.25) it is possible to close the case, therefore the antecedent is the NP *estojo*.

**Most recent Noun Phrase**

This is a weak factor for anaphora resolution. Usually, the most recent NP that matches number and gender of the anaphor can be the correct antecedent, but this is not always the case. Consider the example:

(1.26) A Filipa telefonou *à Joana*. Está sempre a telefonar-lhe.

Fílipa called *Joana*. (She) is always calling *her*.

The most recent NP is *Joana* so it would be chosen has the antecedent for *lhe*. However, the second sentence can also be interpreted if the zeroed subject were *Joana* and *lhe* would refer to *Fílipa* instead. Bellow, example (1.27) show us how weak this preference can be.

(1.27) A Fílipa pediu *à Joana* para *lhe* dar uma ajuda.

*Fílipa* asked *Joana* to help *her*. 
As the most recent NP is Joana, it would be chosen as the antecedent for lhe, but in this case the antecedent is A Filipa because the subclause depends on the verb pedir (to ask) and this verb imposes that the infinitive subclause subject be correferent of the indirect object, hence the dative pronoun lhe can only refer to the sentence’s main subject.

Subject preference

The subject preference factor gives preference to the subject of the previous sentence as the antecedent of a subject pronoun:

(1.28) O Artur ligou para o Mário. Ele queria pedir-lhe o carro emprestado.

Artur called Mario. He wanted to ask him to loan the car.

The subject of the above example, O Artur is the antecedent of the anaphor Ele. However this preference is not so strong. Again it is easy to find a counter-example:

(1.29) O Artur ligou para o Mário. Ele não atendeu.

Artur called Mario. He did not answered.

The person who did not answered the phone was Mário. In this case, subject preference does not hold.

As we see, some factors can be considered to be more important than others, mainly due to the analyzed language characteristics. In Portuguese, for example, gender and number agreement is a stronger factor than the most recent noun phrase, as we can exclude some candidates based on both candidate and anaphor gender and number. The proximity factor, i.e. the relative distance between an anaphor and the candidate antecedents, on the other hand, is not entirely determinantal in the anaphora resolution procedure. However this does not mean that some weaker factors should be seen as negligible. The use of several anaphora resolution factors in combination allows greater confidence in the anaphor antecedent identification.
1.5 Dissertation Structure

In Chapter 2, the three mainstream anaphora resolution approaches are described and compared. Chapter 3 introduces the solution implemented and Chapter 4 the work that involved its conception. In Chapter 5, evaluation criteria are presented as well as some auxiliary tools and the final results. Finally, Chapter 6 presents an overall assessment of the present work and points to further improvements.
Related Work

2.1 Introduction

The most important research on anaphora resolution is reported back to the 1960s [14]. The vast majority of this early stage is described as theoretically-oriented and ambitious work regarding the types of anaphora handled. These approaches were heavily based on domain and linguistic knowledge.

Over the 1990s, with the need of getting more robust, language independent and inexpensive NLP systems, researchers were encouraged to move away from the approaches based on extensive domain and linguistic knowledge. Also, the increasing availability of annotated corpora impelled the rising of new anaphora resolution systems. Annotated corpora, containing morphological, semantic and syntactic information, provide a powerful resource for many approaches, from co-occurrence rule derivation, to training machine learning algorithms or statistical approaches. It was the emergence of a new trend in anaphora resolution research [14].

It is possible to identify three mainstream anaphora resolution areas: statistical, machine learning and syntax-based approaches. This chapter describes some of the most influential strategies in those areas and presents their evaluation results.

Typically, in Natural Language Processing systems, three measures of assessment are used: precision, recall and f-measure. Some of the approaches present a forth measure, success rate. This last one is computed in the same way as precision. These measures are defined in Figure 2.1:

2.2 Statistical approaches

Statistical approaches process large amounts of annotated corpora analyzing the occurrence of anaphors and its candidates, regarding its morphosyntactic characteristics and semantic roles.
\[
\text{recall} = \frac{\text{Number of correctly resolved anaphors}}{\text{Number of all anaphors}}
\]

\[
\text{precision} = \frac{\text{Number of correctly resolved anaphors}}{\text{Number of anaphors attempted to be resolved}}
\]

\[
f\text{-measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

Figure 2.1: Evaluation measures

The analysis of this training corpus produces patterns which are used to identify anaphors in the test corpus.

2.2.1 Collocation patterns-based approach

Introduced by Ido Dagan and Alon Itai [6][7], this statistical approach resolves third person pronouns based on co-occurrence patterns. These patterns are automatically harvested from large corpora and are used to filter out unlikely antecedent candidates.

In this model the anaphor is tested by substituting it by all its candidate antecedents. Every antecedent must satisfy the selectional restrictions (see Chapter 1.4.1). The candidate that produces most frequent co-occurrence patterns is preferred.

The following example is used by the authors and was taken from the Canadian Hansard corpus, a set of proceedings from the Canadian Parliament:

\[(2.1) \text{ They knew full well that the companies held tax money aside for collection later on the basis that the government said it was going to collect it.} \]

Above, there are two occurrences of it. The first is the subject of collect and the second is its object.

Table 2.1 illustrates the co-occurrence patterns produced by the three antecedent candidates money, collection and government in the Hansard corpus. It also lists the number of times each one of these patterns occurred in the corpus. Using Table 2.1’s information it is possible to resolve that, in this example, government is the antecedent of the first it and money of the second.
The model operates in two phases: the acquisition phase were the corpus is processed and the statistical database is created, and the disambiguation phase were the anaphors are resolved using the database built before. The database contains collocation patterns for the following pairs: “subject-verb”, “verb-object” and “adjective-noun”.

Dagan and Itai evaluated their system by resolving anaphoric occurrences of pronoun *it*. They manually extracted random sentences from the *Hansard* corpus containing occurrences of the pronoun. These sentences were filtered out by removing the ones containing non-anaphoric occurrences of *it*, instances of anaphoric *it* whose antecedent was not an NP and instances where the anaphor was not involved in one of the syntactic relations mapped by the database described above. Finally, the cases in which the anaphor had only one possible antecedent were also removed. The experiment used 59 examples taken from the 29 million words corpus. The algorithm could not find the antecedent for 21 of the 59 examples. In the remaining ones the system proposed the correct antecedent for 87% of the cases.

### Table 2.1: Co-occurrence patterns associated with the verb *collect* based on an excerpt from the *Hansard* corpus

<table>
<thead>
<tr>
<th>Subject</th>
<th>Verb</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection</td>
<td>collect</td>
<td>0</td>
</tr>
<tr>
<td>money</td>
<td>collect</td>
<td>5</td>
</tr>
<tr>
<td>government</td>
<td>collect</td>
<td>198</td>
</tr>
<tr>
<td>collect</td>
<td>collection</td>
<td>0</td>
</tr>
<tr>
<td>collect</td>
<td>money</td>
<td>149</td>
</tr>
<tr>
<td>collect</td>
<td>government</td>
<td>0</td>
</tr>
</tbody>
</table>

2.3 Machine learning approaches

Natural language understanding requires large amounts of knowledge, like real-world, morphological, syntactic and semantic knowledge. Machine learning approaches gave the possibility of acquiring this information automatically. They use a set of patterns to extract knowledge from raw or annotated corpora and use it to produce decision trees, among other devices, like the systems presented in the following sections.
2.3.1 RESOLVE system

McCarthy and Lehnert’s approach [10] uses the C4.5 decision-tree algorithm [17] to learn how to classify coreferent noun phrases in the domain of business joint ventures.

The feature vectors used by RESOLVE were created based on all the pairings of anaphors and antecedents, taken from a text manually annotated for coreferential noun phrases. This text deal with joint Venture topics. The pairings that contained coreferent phrases formed positive instances, whereas those that contained non coreferent phrases formed negative instances.

From the 1230 feature vectors that were created from the entity references marked in 50 texts, 322 (26%) were positive and 908 (74%) were negative. The following features and values were used:

- **Name**: Does a reference contain a name? Possible values {yes, no}.

- **Joint venture child**: Does a reference refer to a joint-venture child, e.g. a company formed as a result of a tie-up among two or more entities? Possible values {yes, no, unknown}.

- **Alias**: Does one reference contain an alias of the other, i.e. does each of the two references contain a name and is one of the names a substring of the other name? Possible values {yes, no}.

- **Both joint venture child**: Do both references refer to a joint-venture child? Possible values {yes, no}.

- **Common NP**: Do both references share a common NP? Possible values {yes, no}.

- **Same sentence**: Do the references come from the same sentence? Possible values {yes, no}.

For the evaluation of RESOLVE, the MUC-5 English Joint Venture corpus was used. All pre-processing errors were manually post-edited. The best results achieved were 80.1% recall, 92.4% precision and 85.8% F-measure.
2.4 Syntax-based approaches

Syntax-based approaches operate on the rules and principles that control sentence structure, usually represented by syntactic trees.

2.4.1 Possessive Pronominal Anaphor Resolution in Portuguese Written Texts

Paraboni and Lima [15] focused their work on the Portuguese possessive prononominal anaphor (PPA), in particular on the third person possessive pronouns in intrasentential occurrences. According to them, PPAs are different from other kinds of anaphors, being the main difference the lack of gender and number agreement between PPAs and their antecedents.

(2.2) O Mário vai ter com as suas irmãs.

Mário goes to meet his sisters.

In example (2.2) the pronoun suas is a determinant of irmãs (sisters) and therefore it agrees with his head noun in gender and number, that is, suas is in the feminine plural form. However its antecedent, Mario, is a masculine singular noun.

To solve the possessive pronominal anaphora, six factors were defined, based on syntactic, semantic and pragmatic knowledge.

At a syntactic level, a number of factors were extracted by way of syntactic rules based on surface patterns. According to the authors, surface patterns are typical expressions in the domain, which give information about the PPAs antecedents.

- $F1$ - in the pattern $<NP$ and $|$ or PPA>, $<NP>$ must be elected the most probable antecedent of $<PPA>$. Ex: "John and his dog";

- $F2$ - in the pattern $<ofNP...ofPPA>$, $<NP>$ must be elected the most probable antecedent of $<PPA>$. This rule deals with some cases of syntactic parallelism. Ex: "the death of Suzy, of her children and...";
• $F_3$ - in the pattern $<\text{NP of PPA}>$, $<\text{NP}>$ is not a valid candidate for $<\text{PPA}>$. Ex: in “the death of his son”, ”death” is not a valid candidate;

• $F_4$ - in the pattern $<\text{NP of NP of NP... of NP}>$, only the full chain and the last NP can be considered candidates for PPAs antecedents, i.e., NPs in the middle of the chain can be discarded.

As the rules based on the syntactic level were not sufficient to discriminate among a large set of candidates, semantic knowledge was also used. The semantic approach considered the semantic relations that could be expressed by way of a possessive, such as ownership, part-of, subject or object. To apply this knowledge, object classes and possible possessive relations between them were created. For example, for the anaphor “their hunt” an antecedent of the class of $<\text{animals}>$ should be accepted.

• $F_5$ - There must be a valid possessive relation between a PPA and its antecedent.

A pragmatic factor was included to deal with the cases where semantic ambiguity arises among two or more acceptable candidates and abstract anaphors/antecedents, which cannot be solved by simply applying possessive relation rules. To solve this, a factor based on Brennan centering algorithm [3] and Mitkov [12] subject/object and domain concepts preference were used.

• $F_6$ - The sentence center will be preferred among remaining candidates.

The previous factors were grouped in three knowledge bases modules: surface patterns, possessive relations and sentence center. These modules work as specialist agents. A solver agent receives the anaphor to be analysed and writes its information to a blackboard. The specialists watch the blackboard and contribute to the resolution with their evaluation hypothesis. The specialist agent analyses all the contributions and choose the preferred antecedent candidate.

The system was evaluated using as a corpus a Brazilian Portuguese text on environment protection law, containing 198 PPAs and a scientific magazine article corpus with 100 PPAs. The results were 92.97% for the first text and 88% for the second text.
2.4. SYNTAX-BASED APPROACHES

2.4.2 Hobbs’s naïve approach

In 1978, Jerry Hobbs presented his syntax-based pronoun resolution algorithm. For Hobbs, parse trees represent the correct grammatical structure of sentences. Based on this, his algorithm acts on trees surface. The algorithm is described below:

1. Begin at the NP node immediately dominating the pronoun in the parse tree of the sentence S;

2. Go up the tree to the first NP or S node encountered. Call this node X, and call the path used to reach it $p$;

3. Traverse all branches below node X to the left of path $p$ in a left-to-right, breadth-first fashion. Propose as the antecedent any NP node encountered that has an NP or S node between it and X;

4. If the node X is the highest S node in the sentence, traverse the surface parse trees of previous sentences in the text in order or recency, the most recent first; each tree is traversed in a left-to-right, breadth-first manner, and when an NP node is encountered, it is proposed as the antecedent. If X is not the highest node in the sentence, proceed to step 5;

5. From node X, go up the tree to the first NP or S node encountered. Call this node X and call the path traversed to reach it $p$;

6. If X is an NP node and if the path $p$ to X did not pass through the N-bar node that X immediately dominates, propose X as the antecedent;

7. Traverse all branches below the node X to the left of path $p$ in a left-to-right. breadth-first manner. Propose any NP node encountered as the antecedent;

8. If X is the S node, traverse all branches of node X to the right of path $p$ in a left-to-right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP node encountered as the antecedent;

Hobbs’s algorithm considers plural and collective singular noun phrases and selects semantically compatible entities.

(2.3) John sat on the sofa. Mary sat by the fireplace. They faced each other.

In the example above the algorithm would propose Mary and John, rather than Mary, the fireplace or the sofa.

Hobbs evaluated 300 pronouns from three different texts, all with different structures. These texts were manually analyzed, removing any pre-processing errors, and thus providing an accurate resource. He discovered that 98% of the antecedents were in the current and in the previous sentence. Hobbs’s algorithm worked in 88.3% of the cases and his version with selectional constraints worked in 91.7%. Then he tested the algorithm for only the cases in which more than one plausible antecedent occurred in the candidate set, getting the success rate of 81.8%.

2.4.3 Mitkov’s Anaphora Resolution System

Motivated by the need of a robust, real world operating algorithm, Ruslan Mitkov [14] developed a knowledge-poor approach for pronoun anaphora resolution.

This model operates over antecedent indicators. It receives the output of a POS parser and an NP extractor, locates noun phrases within a distance of two sentences, checks them for gender and number agreement with the anaphor and then applies the indicators to the remaining candidates by assigning them a score. The NP with the highest score is proposed as the antecedent.

Antecedent indicators

After locating noun phrases and passing through the gender and number agreement filter, the antecedent indicators are applied. They can be distinguished as boosting or impeding. The boosting indicators apply a positive score to the candidate and the impeding apply a negative one.

The indicators are listed below:
2.4. SYNTAX-BASED APPROACHES

- **First noun phrase**: A score of +1 is assigned to the first NP in a sentence;

- **Indicating verbs**: A score of +1 is assigned to those NPs immediately following a verb which is a member of a predefined set;

- **Lexical reiteration**: A score of +2 is assigned to those NPs repeated twice or more in the paragraph in which the pronoun appears and a score of +1 is assigned to those NPs repeated once in the paragraph;

- **Section heading preference**: A score of +1 is assigned to those NPs that also occur in the heading of the section in which the pronoun appears; This score is awarded in addition to the score of +1 obtained through *lexical reiteration* due to the repetition of a specific NP in a following passage;

- **Collocation match**: A score of +2 is assigned to those NPs that have an identical collocation pattern to the pronoun;

- **Immediate reference**: A score of +1 is assigned to those NPs appearing in constructions of the form ‘\(\ldots (\text{You}) \text{V1 NP} \ldots \text{con (you)} \text{V2 it (con (you) V3 it)}\)’ where \text{con} is \{ and/or/before/after/until \ldots \};

- **Sequential instructions**: A score of +2 is applied to NPs in the NP1 position of constructions of the form: ‘To V1 NP1, V2 NP2; (Sentence). To V3 it, V4 NP4’;

- **Term preference**: A score of +1 is applied to those NPs identified as representing terms in the genre of the text;

- **Boost pronoun**: As NPs, pronouns are permitted to enter the list of candidates of other pronouns;

- **Syntactic parallelism**: An NP in a previous clause, with the same syntactic role as the current anaphor is awarded a score of +1;

- **Frequent candidates**: The three NPs that occur most frequently as competing candidates of all pronouns in the text are awarded a score of +1.

- **Indefiniteness**: Indefinite NPs are assigned a score of -1;

- **Prepositional noun phrases**: NPs appearing in prepositional phrases are assigned a score of -1;
• Referential distance: NPs in the previous clause, but in the same sentence as the pronoun are assigned a score of +2. Those in the previous sentence to the pronoun are assigned a score of +1. The NPs in the sentence beyond that are assigned a score of 0 and more distant ones are assigned a score of -1.

It is possible to identify five main phases in the MARS operation.

1. The text to be processed was syntactically parsed, using Conexor’s FDG Parser [20], which returns the parts of speech, morphological lemmas, syntactic functions, grammatical number and dependency relations between tokens in the text, facilitating complex NP extraction;

2. Anaphoric pronouns are identified and non-anaphoric and non-nominal instances of it are filtered;

3. For each pronoun identified as anaphoric, candidates are extracted from the NPs in the heading of the section in which the pronoun appears; and from NPs in the current and preceding two sentences (if available) within the paragraph under consideration. Once identified, these candidates are subjected to further morphological and syntactic tests;

4. Preferential and impeding factors are applied to the sets of competing candidates. On the application, each factor applies a numerical score to each candidate;

5. The candidate with the highest composite score is selected as the antecedent of the pronoun.

MARS was tested on a set of technical manuals, with 247,401 words and 2,263 anaphoric pronouns, intrasentential and intersentential. Considering the pre-processing errors, the average success rate was 92.27%.

2.4.4 The Mitkov Algorithm for Anaphora Resolution in Portuguese

Chaves and Rino[4] described an implementation of Mitov Algorithm for Brazilian Portuguese which they called RAPM.
2.4. SYNTAX-BASED APPROACHES

RAPM works on a three sentence antecedent search scope. It receives an automatically annotated input and verifies words gender and number through an XML onomastic file having this information about proper nouns. RAPM processes the sentences in each anaphor three-sentence window, identifying potential NP candidates. Like in Mitov’s system, antecedent indicators are attributed to the NPs. Finally the most valued NP is marked as the antecedent.

Next the antecedent indicators used by RAPM are listed:

- *First NP (FNP)*
- *Lexical reiteration (LR)*
- *Indefinite NP (INP)*
- *Prepositional NP (PNP)*
- *Referential Distance (RD)*
- *Nearest NP (NNP)*
- *Proper Noun (PN)*
- *Syntactic parallelism (SP)*

Chaves and Rino assessed their RAPM using the same annotated corpora used by Coelho [5] containing law, literary and, newswire texts. Eight versions of RAPM were produced by combining the antecedent indicators. Each version was identified by “RAPM_n”, being “n” the amount of indicators used.

- RAPM_2: IS = {INP, RD}
- RAPM_3: IS = {INP, PNP, RD}
- RAPM_4: IS = {INP, PNP, RD, NNP}
- RAPM_5: IS = {FNP, LR, INP, PNP, RD}
- RAPM_6_SP: IS = {FNP, LR, INP, PNP, RD, SP}
- RAPM_6_NNP: IS = {FNP, LR, INP, PNP, RD, NNP}
• RAPM_6_PN: IS = \{FNP, LR, INP, PNP, RD, PN\}

• RAPM_8: IS = \{FNP, LR, INP, PNP, RD, SP, NNP, PN\}

According to Table 2.2, RAPM achieved a 67.01% success rate using all eight antecedent indicators.

<table>
<thead>
<tr>
<th>RAPM version</th>
<th>Success rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAPM_8</td>
<td>67.01</td>
</tr>
<tr>
<td>RAPM_3</td>
<td>66.02</td>
</tr>
<tr>
<td>RAPM_6_NNP</td>
<td>64.94</td>
</tr>
<tr>
<td>RAPM_6_PN</td>
<td>63.40</td>
</tr>
<tr>
<td>RAPM_2</td>
<td>62.50</td>
</tr>
<tr>
<td>RAPM_4</td>
<td>61.45</td>
</tr>
<tr>
<td>RAPM_8</td>
<td>61.21</td>
</tr>
<tr>
<td>RAPM_6_SP</td>
<td>60.26</td>
</tr>
</tbody>
</table>

Table 2.2: RAPM overall assessment

2.4.5 Overview

In this section, the evaluation results of the systems presented above will be compared.

Tables 2.3 and 2.4 provide an overview of the systems presented in this section. In both tables the systems are displayed in the rows and their properties in the columns. Table 2.3 compare the resolution type, the resolution method and the type of anaphora. Table 2.4 refers to the characteristics of the systems evaluation, compares the evaluation subject, whether any manual annotation was made for the evaluation and presents the best results obtained.

Before an analysis of the tables, it should be remarked that any comparison should be cautious. The systems follow different resolution procedures and try to solve different anaphora types. Most of the approaches try to solve pronominal anaphora, but the RESOLVE system focus its attention in coreferent noun phrases. All systems were tested in different corpora and only the collocation pattern-based approach, MARS and RAPM did not make use of manual pre-process of the corpus. This is an important characteristic, as the manual annotation focus the systems evaluation on the resolution algorithm instead of the entire system, ruling out any preprocessing errors.

All this considered the PPA Resolution in Portuguese, with a 92.97% success rate, shows the best results, followed by the RESOLVE system with a precision of 92.4% and MARS with a
### 2.4. SYNTAX-BASED APPROACHES

<table>
<thead>
<tr>
<th>System</th>
<th>Resolution type</th>
<th>Resolution method</th>
<th>Type of anaphora</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collocation pattern-based approach</td>
<td>Statistic analysis</td>
<td>Co-occurrence</td>
<td>Third person pronouns</td>
</tr>
<tr>
<td>RESOLVE</td>
<td>Machine learning</td>
<td>C4.5 algorithm</td>
<td>Coreferent noun phrases in the domain of joint business ventures</td>
</tr>
<tr>
<td>PPA Resolution in Portuguese</td>
<td>Syntax, semantic and pragmatic based</td>
<td>Surface patterns analysis</td>
<td>Third person possessive pronouns</td>
</tr>
<tr>
<td>Hobbs’s naïve approach</td>
<td>Syntax-based</td>
<td>Parse-tree analysis</td>
<td>Pronominal anaphora</td>
</tr>
<tr>
<td>MARS</td>
<td>Syntax-based</td>
<td>Antecedent indicators</td>
<td>Third person pronouns</td>
</tr>
<tr>
<td>RAPM</td>
<td>Syntax-based</td>
<td>Antecedent indicators</td>
<td>Third person personal pronouns</td>
</tr>
</tbody>
</table>

Table 2.3: Systems overview: features

<table>
<thead>
<tr>
<th>System</th>
<th>Evaluation subject</th>
<th>Manual annotation</th>
<th>Best Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collocation pattern-based approach</td>
<td>Parliamentary proceedings</td>
<td>✓</td>
<td>87% success rate</td>
</tr>
<tr>
<td>RESOLVE</td>
<td>MUC-5 English joint venture corpus</td>
<td>✔</td>
<td>80.1% recall 92.4% precision 85.8% f-measure</td>
</tr>
<tr>
<td>PPA Resolution in Portuguese</td>
<td>198 personal pronouns from an environment 100 personal pronouns from scientific magazine articles</td>
<td>N.A.</td>
<td>92.97% success rate</td>
</tr>
<tr>
<td>Hobbs’s naïve approach</td>
<td>100 pronouns from literary text 100 pronouns from a history book 100 pronouns from newspaper</td>
<td>✔</td>
<td>91.7% success rate</td>
</tr>
<tr>
<td>MARS</td>
<td>Technical manuals</td>
<td>✗</td>
<td>92.27% success</td>
</tr>
<tr>
<td>RAPM</td>
<td>Law, literary and newswire corpora</td>
<td>✗</td>
<td>67.01% success</td>
</tr>
</tbody>
</table>

Table 2.4: Systems overview: evaluation
success rate of 92.27%. Despite being one of the early anaphora resolution systems, Hobb’s naïve approach presents a success rate of 91.7%, the third best in the evaluation table, supporting the idea that it is still a valid benchmark among anaphora resolution systems.
3.1 Introduction

This Chapter describes a solution for the Anaphora Resolution for Portuguese based in the Mitkov’s Anaphora Resolution System (see Chapter 2.4.3). The system presented by Ruslan Mitkov is a knowledge-poor approach, as it avoids complex semantic text analysis, making use of a set of syntactic indicators to determine anaphoric antecedents. It is a system used in several languages, including Portuguese, with interesting results.

The system presented in this chapter receives the output from the Xerox Incremental Parser [22] integrated at the L2F processing chain.

3.2 Xerox Incremental Parser

The Xerox Incremental Parser (XIP) is a text parser that produces annotated text with relevant morphosyntactic and semantic information. XIP is able to receive several kinds of inputs to analyse: raw ASCII text, a sequence of tokenized and morphologically analysed words, a sequence of disambiguated words or an XML input file. From the input it is possible to extract several kinds of information from XIP using grammar rules, for example:

- Chunks: e.g., noun phrases, verb phrases;
- Dependencies: e.g., subject, object;
- Named entities: e.g., people, locals, organizations;

Before it is provided to XIP, the input text passes through a processing chain, composed of five main procedures. First the text is segmented in individual tokens. Then a morphosyntactic analysis is performed by the Palavroso system [11] that adds part-of-speech tags (e.g. noun,
verb) to the previously identified tokens. After this, there is a sentence segmentation, in which the text is segmented into sentences. The result of this operation is converted to XML format in order to be used by the morphossyntactic rule disambiguator, *RuDrICo* [16], where the possible ambiguities from the *Palavroso* result are corrected and word contractions are resolved ("do" = "de" + "o"). Finally, the data passes through a statistic disambiguator, *Marv* [18] based on the Viterbi algorithm. This last step chooses the most likely part-of-speech tag for each word. The existence of two morphossyntactic disambiguators is justified by the fact that Marv’s training corpus contains around 250,000 words, which is not a large enough to ensure a correct POS tagging. Figure 3.1 illustrates the processing chain.

![Figure 3.1: L^2F XIP processing chain](image)

The data is then provided to XIP itself where the local grammars are applied and some lexical information is added. At last, XIP segments data into *chunks* and calculates the dependencies between them.

### 3.2.1 Dependency Rules

As stated before, it is possible to implement dependency rules to locate and extract information from texts using XIP [22]. This acquires great importance in anaphora resolution, as many times recognizable patterns that evidence the existence of the anaphora phenomenon occur in texts.

Dependency rules are composed of three parts:

1. A regular expression pattern;
2. A collection of conditions about relations between the nodes of a chunk tree or the nodes themselves, independent of the tree structure;
3. A dependency term.
Next, is the dependency rules syntax:

\[
|\text{pattern}| \text{if} \ \langle \text{condition} \rangle \ \langle \text{dependency terms} \rangle
\]

The pattern contains a tree regular expression that describes the structural properties of parts of the input tree. The condition is any Boolean expression built from dependency terms, linear order statements, and operators. The pattern and condition are both optional.

Using this rules it was possible to locate patterns evidencing the following dependency relations:

- **ACANDIDATE(1,2)**: token 1 is a possible anaphor of token 2;
- **ACANDIDATE_POSS(1,2)**: according to the rules in Chapter 4.1.2, token 1 is the anaphor of token 2;
- **INVALID_ACANDIDATE(1,2)**: according to the rules in Chapter 4.1.2, token 1 cannot be the anaphor of token 2;
- **IMMEDIATE_REFERENCE(1,2)**: according to Chapter 2.4.3, token 1 is in immediate reference with token 2.

Next, some examples of sentences and identified relations are presented:

(3.1) A Maria viu a Isabel e cumprimentou-a.

*Maria saw Isabel and greeted her.*

In example (3.1), two **ACANDIDATE** relations are created: **ACANDIDATE(a, Maria)**, **ACANDIDATE(a, Isabel)**. A third relation, **IMMEDIATE_REFERENCE**, between a and Isabel is also created.

(3.2) O Miguel vai a casa dos seus pais.

*Miguel goes to his parents home.*

In example (3.2), although *casa* is the nearest noun to *seus* an **INVALID_ACANDIDATE** relation between *seus* and *casa* is found. In fact, the anaphor antecedent, in this case, is *Miguel*. 
(3.3) A *casa* de campo tem as *suas* paredes pintadas de branco.

The country *house* has *its* walls painted white.

Example (3.3) shows a case where two nouns, *casa* and *campo*, preceed the pronoun *suas*, and only one *ACANDIDATE* relation is created between *suas* and *casa*.

A dependency relation rule is presented in example (3.4). It recognizes *ACANDIDATE* relations between a pronoun existent in a PP and the head of phrase of an NP.

(3.4) Dependency rule example:

```
|?*, NP#1, ?*, PP{?*, pron#2[poss=˜]}|
if(HEAD(#3,#1) &
   #2[number]:#3[number] & #2[gender]:#3[gender] &
   ~ACANDIDATE(#2,#3))
   ACANDIDATE(#2,#3)
```

Looking in detail to the previous rule, the pattern recognizes NPs that occur at any position in a sentence, followed by a PP that contain a pronoun that is not possessive. Parts of speech or chunks followed by ’#n’ are variable attributions: (*NP#1* creates a variable ’#1’ that points to a *NP*), and pron#2 is a pronoun inside a PP that is not a possessive pronoun (poss= ). The condition is true if there is any element #3 that is the head of #1 (the NP) and this element has the same number and gender as #2 (the pronoun) and there is no *ACANDIDATE* relation between both yet. If all the conditions are verified, an *ACANDIDATE* dependency between #2 and #3 is created. For the sentence *O João deu ao Pedro um bolo feito por ele* two *ACANDIDATE* relations are created: *ACANDIDATE(ele, João)* and *ACANDIDATE(ele, Pedro)*.

Although dependency rules may seem a promising way to discover possible anaphors and antecedent candidates, one must stress out that these relations are only recognized in words in the same sentence. For intersentential anaphora (see Chapter 1.3.7), a multi-sentence analysis is required, which is out of reach of XIP.

Altogether, 17 rules were implemented. All rules are listed in detail in appendix A.
3.2. XEROX INCREMENTAL PARSER

3.2.1.1 Number and Gender Rules

In addition to the Dependency Rules, other rules were implemented. These rules do not create relations between text elements, but add more information to these elements in an effort to ensure a correct identification of gender and number on composed nouns.

In Portuguese, a common noun can be feminine and/or masculine. Proper names, however, behave differently: given names are usually associated to a specific gender and seldom accept plural; family names do not have gender and may accept a plural mark, even if they can be used in plural without any explicit working [2].

For example, some Proper Nouns can belong to a man or a woman. João is typically a masculine name, but there are women called João: one can also say Os Silvas (plural marking) as well as Os Silva (no plural mark). This means that we can not rely only on the noun gender to determine the subject gender.

(3.5) O Filipe estava a falar da João. Ele encontrou-a ontem.

Filipe was talking about João. He met her yesterday.

Example (3.5) shows two proper nouns, Filipe and João, and two anaphors, Ele and a. Looking to the nouns gender, there are only a gender agreement between Filipe and Ele, as one can not determine the gender of João. Because of this, we must search for more information. The prepositional phrase da João is composed by a preposition de, an article a and the noun João, as presented in Figure 3.2. The article is a determinant that explicitates the gender and number of the noun. In this case it it makes clear that João is singular feminine, and therefore there is a gender and number agreement with the anaphor a.

![Figure 3.2: Syntactic structure of the PP, da João](image)

Many proper names are ambiguous to common nouns (e.g. Reis); while common nouns may show gender-number variation (e.g. rei, rainha, reis, rainhas), proper names seem not to have
such properties. Furthermore, proper names combine themselves to form longer named entities (e.g. Pedro Reis) and its gender-number overall value is determined by the first (given) name in the string.

Due to these ambiguities, which would influence the anaphora resolution procedure, two generalizations were made:

1. In noun phrases or prepositional phrases, the gender and numbers of a noun is the same than the article;

2. The number feature (singular, plural) of a composed noun, is the same as the first noun.

These generalizations were achieved through the implementation of the following rules:

1. Article determines number and gender agreement:

   \[
   |\text{NP}(\text{art[masc]}, \text{noun[masc=+]}))| \sim \\
   |\text{NP}(\text{art[fem]}, \text{noun[fem=+]}))| \sim \\
   |\text{NP}(\text{art[sg]}, \text{noun[sg=+]}))| \sim \\
   |\text{NP}(\text{art[pl]}, \text{noun[pl=+]}))| \sim \\
   \\
   |\text{PP}(\text{art[masc]}, \text{noun[masc=+]}))| \sim \\
   |\text{PP}(\text{art[fem]}, \text{noun[fem=+, masc=+]})| \sim \\
   |\text{PP}(\text{art[sg]}, \text{noun[sg=+]}))| \sim \\
   |\text{PP}(\text{art[pl]}, \text{noun[pl=+]})| \sim \\
   
   \]

2. The first noun of a composound noun determines the number:

   \[
   |\text{NP}(??, \text{noun[sg=+]})\{??, \text{noun[sg]}\})| \sim \\
   |\text{NP}(??, \text{noun[masc=+]})(\text{noun[masc]})| \sim \\
   |\text{NP}(??, \text{noun[fem=+]})(\text{noun[fem]})| \sim \\
   
   \]

Instead of dependency relations, these rules change text elements features. Take the following rule as an example:

\[
|\text{NP}(\text{art[masc]}, \text{noun[masc=+]}))| \sim \\
\]

The rule matches a noun phrase in which the first element is an article and the second is a noun. If the article shows the masculine feature, the same feature is added to the noun. The tilde symbol (\sim) at the end of the rule states that no dependency is created.
3.3 Anaphora Resolution Module

The Anaphora Resolution Module operates independently of XIP processing chain. It receives the corpus to evaluate and runs XIP to obtain its result, on which will make the Anaphora analysis. This way XIP’s environment and complexity is abstracted, making the anaphora resolution an isolated procedure. Figure 3.3 illustrates the interaction between the ARM and XIP.

Like the approach presented by Mitkov (see Chapter 2.4.3), the ARM will only try to resolve pronominal anaphora, therefore there has to be a pronoun identification phase.
3.3.1 Output

The result of the ARM is an XML stream containing the input corpus, annotated with information about the existing syntactic structures. Each one of these structures will be tagged and numbered. Anaphoric nodes will have attributes referring to the type of anaphora and its antecedent. Next an example of this annotation is presented:

(3.5) Para o compositor John Cage, qualquer som podia ser música. Em seu entender, o ruído não existe, há apenas som.

For the composer John Cage, any sound could be music. In his view, there is no noise, only sound.

From example (3.30) would result the annotation presented in Figure 3.4. Looking at the resulting annotation the node with id 202 containing the pronoun seu has two attributes that do not exist in other nodes: the attribute anaphora=’[pronominal]’ which refers to the nature of the anaphor and the attribute antecedent=’[4]’ that points to the anaphor antecedent, in this case the PP with id 4. The antecedent itself is not the entire PP but its head, the noun John Case.
3.3. ANAPHORA RESOLUTION MODULE

Figure 3.4: ARM output
4.1 Introduction

The proposed solution for the Anaphora Resolution Module is developed in the Java programming language.

Like proposed on Chapter 1.4 the ARM operates in three steps:

1. anaphor identification;
2. antecedent candidates identification;
3. choosing the most likely antecedent candidate for each anaphor.

4.1.1 Anaphor Identification

The anaphor identification is based in Chapter 1.3.1. This means that all third person personal pronouns, including possessive, relative and demonstrative pronouns are identified as possible anaphors. All pronoun (but not possessive) mus be head of a phrase.

Although the rules above enhance a correct anaphor identification there are some exceptional cases to consider:

(4.1) O João viu-se ao espelho.

João saw himself at the mirror.

(4.2) Vendem-se casas.

Houses for sale.
(4.3) Precisa-se de ajuda.

Help is needed

The previous examples show different occurrences of pronoun se. In (4.1) it appears as a reflexive pronoun and anaphor of João. In example (4.2) se is linked to the transitive verb vendem. Because this verb is in the plural, its subject is casas, which allow us to consider this a passive-like pronominal construction, where the verb’s object is raised to the subject position, the verb agrees with the new subject and the reflexive pronoun is inserted (the agent is omitted). In (4.3), se is an indefinite pronoun linked to the intransitive verb precisa, equivalent to an indefinite subject node as alguém (someone). The two last cases are examples of non-anaphoric occurrences of pronoun se.

As one can see, the pronoun se presents several grammatical roles and the current XIP processing chain at L2F can not always identify them. Therefore, this pronoun will be excluded from the anaphor identification phase.

4.1.2 Antecedent Candidates Identification

After identifying an anaphor is time to find its antecedent candidates. As described in Chapter 1.4.1, the ARM only considers as possible candidates, nouns and pronouns within a distance of 3 sentences from the anaphor.

The system also considers gender and number agreement factors (see Chapter 1.4.1) between anaphor and the candidate. The fact that Portuguese has a rich morphology and nouns are often gender-number marked, brings great importance for the gender-number constraint. However this has two exceptions.

Coordinated NPs

Coordinated NPs occur when more than one NP are joined by a coordinative conjunction. In this case, the verb would be inflected in the plural as in (4.4).

(4.4) O João e a Ana vão ao cinema. Eles gostam de filmes.

João and Ana go to the cinema. They like movies.
In example (4.4), the pronoun *Eles* is the anaphor of *João* and *Ana*. Despite there is a feminine noun (*Ana*) in the coordinated NP the pronoun anaphor is masculine. The pronoun should only take the feminine form in the case where all nouns are feminine. Table 4.1 distinguishes the pronoun genders for all the possible cases.

<table>
<thead>
<tr>
<th>pronoun gender</th>
<th>All nouns feminine</th>
<th>All nouns masculine</th>
<th>At least one noun masculine</th>
</tr>
</thead>
<tbody>
<tr>
<td>feminine</td>
<td>masculine</td>
<td>masculine</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.1: Pronoun gender for coordinated NPs

**Possessive Pronouns**

In Portuguese, possessive pronouns do not show gender or number agreement with their antecedents. This agreement occurs with the noun they determine/modify.

(4.5) *O Vitor* não encontra as *suas* sapatinhas.

*Vitor* can not find *his* sneackers.

In example (4.5) *suas* is the anaphor of *Vitor*, despite the name is masculine and singular the pronoun is feminine and plural, in agreement with the noun *sapatinhas* (*sneackers*) it determines.

### 4.1.3 Choosing the Anaphor’s Antecedent

Once the anaphor is identified and the antecedent candidates are chosen, the ARM determines which candidate is the anaphor’s correct antecedent. To perform this task, a set of parameters are used to score each candidate, according to their syntactic role in the analysed text.

These parameters were chosen based on Mitkov [14] (see Chapter 2.4.3) and Chaves and Rino [4] (see Chapter 2.4.4). The rules defined in Chapter 3.2.1 allow the creation of two more parameters: *Possessive Pronoun Probable Candidate* and *Possessive Pronoun Invalid Candidate*. The assignment of each parameter value is described in Chapter 5.3.

All implemented parameters and their respective values are listed below:

- *First Noun Phrase (FNP)*: a score of +1 is assigned to the first NP in a sentence:
• *Collocation Match (CM)*: a score of +1 is assigned to those NPs that have an identical collocation pattern to the pronoun;

• *Syntactic Parallelism (SP)*: an NP in a previous clause with the same syntactic role as the current is awarded a score of +1;

• *Frequent Candidates (FC)*: the three NPs that occur most frequently as competing candidates of all pronouns in the text are awarded a score of +1;

• *Indefiniteness (IND)*: Indefinite NPs are assigned a score of -2;

• *Prepositional Noun Phrases (PPN)*: NPs appearing in prepositional phrases are assigned a score of -1;

• *Proper Noun (PN)*: a proper noun is awarded a score of +2;

• *Nearest NP (NNP)*: the nearest NP to the anaphor is awarded with a score of -1;

• *Referential Distance_0 (RD0)*: NPs in the previous clause, but in the same sentence as the pronoun are assigned a score of +2;

• *Referential Distance_2 (RD2)*: NPs in two sentences distance are assigned a score of -1;

• *Referential Distance_2+ (RD2+)*: NPs in more than two sentences distance are assigned a score of -3;

• *Possessive Pronoun Probable Candidate (PPPC)*: a score of +1 is assigned to the candidate if is present on an ACANDIDATE_POSS(A,C) (see Section 2.4.1) relation for anaphor “A”;  

• *Possessive Pronoun Invalid Candidate (PPPC)*: a score of -3 is assigned to the candidate if is present on an INVALID_ACANDIDATE(A,C) (see Section 2.4.1) relation for anaphor “A”;

### 4.2 XIP Interaction

The Anaphora Resolution Module operates on XIP’s processing chain result, specially on the chunk trees and dependency relations extracted from corpora (see Chapter 3.2).

Figure 4.1 and Figure 4.2 illustrate the chunk tree and dependency relations obtained from example (4.5).
4.2. **XIP INTERACTION**

(4.5) O Pedro lê um livro.

Pedro reads a book.

![XIP chunk tree for example (4.5)](image)

**Figure 4.1: XIP chunk tree for example (4.5)**

```
HEAD(Pedro,O Pedro)
HEAD(livro,um livro)
HEAD(lê,lê)
QUANTD(livro,um)
DETD(Pedro,O)
VDOMAIN(lê,lê)
SUBJ_PRE(lê,Pedro)
CDIR_POST(lê,livro)
NE_INDIVIDUAL_PEOPLE(Pedro)
```

**Figure 4.2: XIP dependencies for example 4.5)**

These structures are aggregated in an XML-like format that is used in the anaphora resolution process.

### 4.2.1 XIP XML output

The XIP XML output is composed by the following main elements [21]:

- **DEPENDENCY**: result from a linguistic analysis on NODE elements, performed by the dependency rules presented in Chapter 3.2.1;
- **FEATURES**: provides the features of the DEPENDENCY;
• LUNIT: a linguistic unit. Each LUNIT represents one sentence. Contains a list of NODES and DEPENDENCIES;

• NODE: contains the result of the morphsyntactic analysis. It can be a parent of other NODES or TOKENS;

• PARAMETER: contains the nodes that compose a dependency;

• PCDATA: provides a fragment of the input text;

• READING: provides the disambiguated lexical unit;

• TOKEN: the result of tokenization, morphological analysis, and lexical disambiguation;

• XIPRESULT: contains a list of LUNITS or a list of TOKENS.

Each of these elements contain several attributes, such as name, number or value, amongst others. All this information is structured in an XML tree.

The Anaphora Resolution Module operates on these structures. It parses the document, distinguishing all its elements and operates based on their features. Although there are several Java libraries capable of representing and manipulating XML it was decided to develop an API capable of abstracting the XML tree complexity, converting it into a domain specific structure. The main reason for such decision is the fact that much of the system operation would be done by XML structures manipulation besides of the resolution process itself, as these XML libraries are not specific for the anaphora resolution domain.

4.2.2 XIP API

Analysing the XIP XML elements presented on Chapter 4.2.1, the following domain objects were identified and implemented:

• Dependency: contains the information about XIP dependencies;

• Feature: contains nodes properties, such as masculine our feminine, singular or plural, among others;

• Token: represents the XIP TOKEN;
4.3. ANAPHORA RESOLUTION MODULE

- XipDocument: contains a chunk tree mapped by XIPNodes and the Dependencies of the analysed corpus;

- XIPNode: represents a XIP NODE. It is the basic structure of a chunk tree. It can represent the root element of a sentence, the TOP, as well as intermediate elements, e.g. NOUN nodes, or leaf ones, e.g. tokens.

Figure 4.3 illustrates this domain.

Figure 4.3: XIP API domain

4.3 Anaphora Resolution Module

4.3.1 General Configuration

The Anaphora Resolution Module has to deal with several variable settings that influence the system’s processing, for example which types of pronouns are evaluated, or the antecedent candidates evaluation parameters. These are settings that due to their impact on the system’s overall performance should be accessible and easily changeable. To achieve this requirements a configuration file containing these variables was created.

The configuration file contains the following information:
• number of sentences to be analysed (default: 3);

• types of pronouns to be analysed (default: personal, possessive and relative);

• antecedent candidates parameters (default: values presented in Chapter 4.1.3)

This file is loaded on the ARM initiation, setting up the analysis parameters. Figure 4.4 illustrates the configuration file structure.

```
<ARM-CONFIG>
  <SENTENCE-LIMIT>
    3
  </SENTENCE-LIMIT>
  <PRONOUN-TYPES>
    <TYPES>typeA</TYPES>
    <TYPES>typeB</TYPES>
  </PRONOUN-TYPES>
  <ANTECEDENT-INDICATORS>
    <INDICATOR>
      <NAME>name</NAME>
      <ACRONYM>acronym</ACRONYM>
      <VALUE>value</VALUE>
    </INDICATOR>
    ...
  </ANTECEDENT-INDICATORS>
</ARM-CONFIG>
```

Figure 4.4: Configuration file structure

4.3.2 Domain

The API presented in Chapter 4.2.2 facilitates the process of text analysis, but for the anaphora resolution more structures had to be added. The following concepts were considered and implemented:

• Anaphor: represents the pronoun node identified as an anaphor. Contains a sorted set of candidates, ordered by candidate score;

• Candidate: contains the reference to the candidate node and a list of indicators;

• Indicator: represents a candidate evaluation parameter. It contains the name of the parameter and its value.

Figure 4.5 illustrates the ARM domain.
4.3.3 Algorithm

During the resolution process, four main phases take place:

1. Dependency relations analysis;
2. Tree exploration analysis;
3. Post exploration analysis;

First, the dependencies present in text are analyzed, inserting a new feature on the nodes that compose those dependencies. This way, when a word is parsed, it already contains the information that exists in a dependency, avoiding a dependency search for each analyzed word.

Next the exploration analysis takes place. For each sentence a search for anaphors and possible candidates is performed. Before a candidate is associated to an anaphor, it goes through a series of filters:

1. Sentence limit: the candidate must be within a 3 sentences distance;
2. Gender agreement: the candidate must agree in gender with the anaphor;
3. Number agreement: the candidate must agree in number with the anaphor.
CHAPTER 4. IMPLEMENTATION

Items 2 and 3 are evaluated according to Chapter 4.1.2. Finally the candidate is evaluated by the parameters defined in Chapter 4.1.3 and added to the anaphor candidate list.

When the exploration is complete, all anaphors are located as well as their possible antecedents. At this time a post-exploration phase takes place. All the discovered anaphors are iterated and for all of their antecedent candidates the following indicators are evaluated:

- Sequential Instruction (SI);
- Syntactic Parallelism (SP);
- Frequent Candidates (FC);
- Nearest NP (NNP).

These indicators can only be evaluated at this stage because this is when some necessary information is available, for example, the Frequent Candidates, or which of the candidates is in the nearest NP.

The document tree is then exported in the format described in Chapter 3.3.1.

Figure 4.6 illustrates the ARM execution.

4.4 Input

The Anaphora Resolution Module offers three distinct ways to analyse a text:

1. Input string to be processed by XIP;
2. Input file containing the corpus to be processed by XIP;
3. XML input file containing XIP’s output to be analysed.

For the the first two options the ARM must be in an environment with access to the XIP processing chain, as it has to launch a XIP process containing the input as parameters and read the process’s output. The latter one offers some independence from XIP. One has only to have the result from the XIP processment result to start the anaphora analysis. This feature was implemented using the Command design pattern [8]. The Command pattern provides flexibility
as it allows a complete decoupling between the invoker object and the receiver, which has the task to execute the invoked operation.
This Chapter describes the methods used to assess the performance of the Anaphora Resolution Module are described. In Section 5.5 the results obtained are presented and a comparison with other approaches is made.

5.1 Introduction

To obtain the evaluation measures two annotated corpora were used: the result provided by ARM and the same corpus manually annotated. By comparing both files it is possible to obtain the real number of anaphors and antecedents in a text and the ones identified and resolved by the system.

The task of manually annotating texts can be a time-consuming, complex and error-prone task. This comes mainly from the syntax of the annotation language or the type of information to be introduced. In order to promote this task an Anaphora Manual Annotator was developed.

5.2 Anaphora Manual Annotator

The Anaphora Manual Annotator is an Eclipse Rich Client Platform (Eclipse RCP) based application, developed to reduce the complexity of the manual annotation task, allowing the production and edition of annotated texts containing anaphora information. Figure 5.1 illustrates the application.

In this figure, two sentences are shown: *O Pedro e o Filipe foram às compras. Eles compraram dois bolos* (Pedro and Filipe went shopping. They bought two cakes). The anaphor Eles (*they*) is moved and dropped over the two antecedents, one at a time. The first antecedent of eles is already marked, as it is indicated on the right window. The drag-and-drop of eles over Filipe is being done.
5.2.1 Eclipse Rich Client Platform

The Eclipse RCP is a framework developed by the Eclipse Foundation open source community. It allows the development of portable applications for multiple operating systems using the core and user interface plugins of the Eclipse IDE. The main advantage of using such platform is the fact that the manual annotator was not built from the beginning, what would be a time consuming task. Instead a set of features provided by a stable and tested framework were used.

Application

The application uses the API described in Chapter 4.2.2 to load the analysed documents. In order to allow the annotation task two more concepts were defined and implemented:

1. Anaphora: represents an Anaphora. It contains the anaphor node and a set of antecedent candidates;
2. Annotator: singleton object containing the application data, such as opened documents and their anaphors.

When a document is loaded into the Manual Annotator, a XipDocument, provided by the XIPAPI is created, containing the original document information. This new document is made available to the user. It allows the user to browse the tree structure in order to inspect its contents. The annotation task is made by dragging and dropping the anaphor node into its antecedent. Repeating this task with the same anaphor to different nodes adds a list of antecedents to the same anaphor.

Features

The Anaphora Manual Annotator provides the following features:

- Open a XIP result XML document;
- List opened documents;
- View documents dependency rules;
- Link a pronoun to its antecedent, creating an anaphora, by drag&drop actions;
- View document anaphoras;
- Remove document anaphoras;
- Export XIP documents to annotated documents;
- Open previously annotated documents;
- Inspect document nodes (tokens and nodes);
- View node features.

Using this application it was possible to annotate the training corpus used during the development phase, as well as the evaluation corpus. This last set of texts were annotated by a linguist of the research team at l2f.
5.3 Antecedent Parameters Value Training

The antecedent indicators values were initially the same from Mitkov (see Chapter 2.4.3) and Chaves and Rino’s work (see Chapter 2.4.4), but as the Anaphora Resolution Module is a different system, effectiveness should be affected and a new score was required. To avoid a subjective evaluation a genetic algorithm was implemented, allowing the training of the indicators used, in order to reach an optimal combination of values.

5.3.1 Genetic Algorithm

Genetic Algorithms are based in the natural selection process and are typically used in optimization problems where the best possible solution for a given problem is to be found. When the algorithm starts, a set of base solutions, called individuals, is created. During its execution these individuals suffer a series of operations similar to gene transfer in sexual reproduction. A fitness function measures the proximity of each individual to the optimal solution. Individuals with higher fitness are considered best fit and have greater probability to share their genes with other elements of the population, generating new individuals possibly with a good fitness value also. On the other side, worst fit elements have greater probability to be removed from the population.

For the Anaphora Resolution problem, the individual was defined as the set of antecedent indicators and the f-measure value obtained for evaluating the Anaphora Resolution Module with these parameters represents its fitness.

The following algorithm was implemented. based on the work of Russel&Norvig [19]:

1. New population generation

2. While the best individual fitness is less than THRESHOLD
   
   (a) Generate new empty population
   
   (b) While new population size is less than MAX_POPULATION_SIZE
      
      i. Add a small set of the best individuals to the new population (Elitism)
      
      ii. Select two individuals
      
      iii. Generate new individual by crossing over the previous selected individuals
iv. Calculate the new individual fitness
v. Add new individual to new population
(c) Population = new population

3. return individual with greater fitness value

The following procedures were implemented:

- Fitness function: runs the Anaphora Resolution Module. It uses the indicators defined for the individual and compares its results with a manually annotated text to obtain the f-measure;

- Selection: selects two individuals from the population. An individual with higher fitness has greater probability of being selected, according to the following expression:
  \[ \text{prob} = \frac{\text{individual\_fitness}}{\text{population\_max\_fitness}} \times 100 \]

- Crossover: generates a new individual with 50% genes from the first parent and 50% from the second - uniform crossover.

- Mutation: each new individual has 1% probability of having one or more gene mutated

5.4 Procedure

During the development phase, a series of newspaper articles available at Linguateca [9] were used. The language present on this kind of articles is characterized as direct, informative and written in an impersonal manner.

5.4.1 Training phase

As described on Chapter 5.3 a set of parameters were used to determine the anaphors antecedents. Examples of such indicators can be seen in the Mitkov’s Anaphora Resolution System (see Chapter 2.4.3) and RAPM (see Chapter 2.4.4) and were used as the basis of the work here developed. However the ARM is a different system and operates in a different environment,
for example, none of the systems use the same preprocessing tools. Also the evaluated texts here used to assess the system’s performance are different, so for the same parameters different results were expectable.

Using the algorithm described in Chapter 5.3.1 it was possible to learn which of these indicators were effective and their best values.

A set of 20 texts were used in this training phase, achieving an overall 41.61% f-measure. The indicator values were the following:

- First Noun Phrase: +1;
- Collocation Match: +1;
- Syntactic Parallelism: +1;
- Frequent Candidates: +1;
- Proper Noun: +2;
- Nearest NP: -1;
- Indefinitess: -2;
- Prepositional Noun Phrase: -1;
- Referential Distance(+2): -3;
- Referential Distance(2): -1;
- Referential Distance(0): 2;
- Possessive Pronoun Probable Candidate: +1;
- Possessive Pronoun Invalid Candidate: -3.

5.5 Final Evaluation

To perform the system final evaluation, a corpus manually annotated and made available by the research team at L2F was used. This way, during the development phase, there was no knowledge about this corpus used for the final evaluation.
The evaluation corpus was composed by 20 texts of distinct genres: 8 texts from online forum messages, 1 from a legal corpus and 11 texts from news articles. The corpora contained the total of 692 pronouns, in which 334 of these were evaluated by the ARM, according to Chapter 4.1.1.

The system achieved the following results:

- recall: 30%;
- precision: 38%;
- f-measure: 33.5%.

Looking at the final results there are some considerations to be taken. The Anaphora Resolution Module operates on the output of the XIP processing chain. Each pre-processing error affects greatly the overall results. This fact is observable on precision and recall values. The precision value is higher than the recall, leading to infear that some resolution errors came from pre-processing stage. Other reason for the achieved results is the broad scope of the anaphora resolution module, since it encompasses different types of anaphors. ARM tries to resolve a possessive, relative and personal pronouns while other systems refered to in the related work only dealt with on of these anaphor types. Finally, the analysed corpora was diversified, containing different types of language.
The Natural Language Processing is a vast area, and even a subset like the pronominal anaphora study proved to be very complex.

The objective of this work, was to study the Anaphora Resolution problem for Portuguese, and the development of a system capable of such task. During the development phase, two auxiliary tools were created: an API that facilitates the operation with XIP processing chain and an application to promote manual annotation of anaphora in texts.

The final product is a system that resolves pronominal anaphora. The Anaphora Resolution Module obtained an f-measure of 30.5%, but it must be taken into consideration the fact that it evaluates a wide range of pronouns and operates on the output of another system. Any pre-processment errors would affect the overall performance. To improve the results more work should be done. The rules presented in Chapter 4.1.2 enhance a correct anaphor candidate identification. However, a high number of candidates was verified in several cases. Reducing the amount of possible antecedents would result in a better classification. The study of new antecedent indicators would also enhance a better candidate identification. These parameters were based on the work of Mitkov [14] and Chaves and Rino [4]. Therefore, specific new parameters for the information provided by the XIP processing chain would influence the results. Another strategy that could provide better results is the study of new dependency relation rules (see Chapter 3.2.1). These rules are a powerful resource to harvest relations between words. Finally, an architecture modification could be considered. As observed in Chapter 2.4.5, machine learning approaches can achieve interesting results. Therefore the Anaphora Resolution Module architecture could incorporate features of these approaches, using information gathered from previously processed texts to resolve anaphora.
Bibliography


[8] E. Gamma, R. Helm, R. E. Johnson, and J. Vlissides. Design Patterns. Elements of Reusable Object-Oriented Software. Addison-Wesley, March 1995.


Dependency Rules

// noun has same gender and number of article
|NP{art[masc], noun[masc=+]}| ~
|NP{art[fem], noun[fem=+]}| ~
|NP{art[sg], noun[sg=+]}| ~
|NP{art[pl], noun[pl=+]}| ~

|PP{art[masc], noun[masc=+]}| ~
|PP{art[fem], noun[fem=+,masc=~]}| ~
|PP{art[sg], noun[sg=+]}| ~
|PP{art[pl], noun[pl=+]}| ~

A.1 Multi Word Entries

// number features from sub-names are passed to top-noun
// Maria dos Santos -> singular
|NP{?*, noun[sg=+]{?*,noun[sg]}}| ~

// gender features from sub-names are passed to top-noun
// Maria dos Santos -> singular
|NP{?*, noun[masc=+]{noun[masc]}}| ~
|NP{?*, noun[fem=+]{noun[fem]}}| ~

A.2 Pronoun rules
APPENDIX A. DEPENDENCY RULES

//rule 1.1: Anaphor in PP, antecent in NP
//example: O João deu ao Pedro um bolo feito por ele.
//result: ACANDIDATE(ele, João)

|?*, NP#1, ?*, PP{?*, pron#2[poss="]}| if(HEAD(#3,#1) &
   ~ACANDIDATE(#2,#3) &
   #2[number]:#3[number] & #2[gender]:#3[gender])
 ) ACANDIDATE(#2,#3)

//rule 1.2: Anaphor in PP, antecent in PP
//example:
//result:

|?*, PP#1, ?*, PP{?*, pron#2[poss="]}| if(HEAD(#3,#1) &
   ~ACANDIDATE(#2,#3) &
   #2[number]:#3[number] & #2[gender]:#3[gender])
 ) ACANDIDATE(#2,#3)

//rule 1.3: Anaphor in NP, antecent in NP
//example: O João deu um bolo à Maria e ela ficou contente.
//result: ACANDIDATE(ela, Maria)

|?*, NP#1, ?*, NP{?*, pron#2[poss="]}| if(HEAD(#3,#1) &
   ~ACANDIDATE(#2,#3) &
   #2[number]:#3[number] & #2[gender]:#3[gender])
 ) ACANDIDATE(#2,#3)
A.3 Coordination rules

//rule 2.1: Anaphor in PP, Antecedent in coordinated NPs
//example: A Joana e a Rita levaram um bolo feito por elas.
//result: ACANDIDATE(elas, Joana), ACANDIDATE(elas, Rita)

|?*, NP#1,?*, NP#2, ?, PP{?*, pron#3[poss=−]}|  
  if(#3[pl] &  
    COORD(#4,#5) &  
    COORD(#6,#7) &  
    HEAD(#5,#1) &  
    HEAD(#7,#2) &  
    ~ACANDIDATE(#3,#5) &  
    ~ACANDIDATE(#3,#7) &  
    #3[gender]:#5[gender] &  
    #3[gender]:#7[gender]  
  )  
  ACANDIDATE(#3,#5),ACANDIDATE(#3,#7)

//rule 2.2: Anaphor in NP, Antecedent in coordinated NPs
//example: O João e a Maria foram para casa deles
//result: ACANDIDATE(ele, João), ACANDIDATE(ele, Maria)

|?*, NP#1,?*, NP#2, ?, NP{?*, pron#3[poss=−]}|  
  if(#3[pl] &  
    COORD(#4,#5) &  
    COORD(#6,#7) &  
    HEAD(#5,#1) &  
    HEAD(#7,#2) &  
    ~ACANDIDATE(#3,#5) &  
    ~ACANDIDATE(#3,#7) &
APPENDIX A. DEPENDENCY RULES

#3[gender]:#5[gender] &
#3[gender]:#7[gender]
)
ACANDIDATE(#3,#5),ACANDIDATE(#3,#7)

//rule 2.3: Anaphor in PP, Antecedent in coordinated PPs
//example:
//result:

|?*, PP#1,?*, PP#2, ?*, PP{?*, pron#3[poss=−]}|
if(#3[pl] &
COORD(#4,#5) &
COORD(#6,#7) &
HEAD(#5,#1) &
HEAD(#7,#2) &
¬ACANDIDATE(#3,#5) &
¬ACANDIDATE(#3,#7) &
#3[gender]:#5[gender] &
#3[gender]:#7[gender]
)
ACANDIDATE(#3,#5),ACANDIDATE(#3,#7)

//rule 2.4: Anaphor in NP, Antecedent in coordinated PPs

|?*, PP#1,?*, PP#2, ?*, PP{?*, pron#3[poss=−]}|
if(#3[pl] &
COORD(#4,#5) &
COORD(#6,#7) &
HEAD(#5,#1) &
HEAD(#7,#2) &
¬ACANDIDATE(#3,#5) &
¬ACANDIDATE(#3,#7) &
A.4 Possessive pronouns

//rule 3.1 - F1: Anaphor is possessive pronoun in NP, Antecedent in NP
//example: O pedro e a sua mulher
//result: ACANDIDATE_POSS(sua, Pedro)
|?*, NP#1, CONJ, NP{?*, pron#2[poss=+]}|
  if(HEAD(#3,#1) &
    ^ACANDIDATE_POSS(#2,#3)
  )
  ACANDIDATE_POSS(#2,#3)

//rule 3.2 - F2: Anaphor is possessive pronoun in PP, Antecedent in PP
//example: A escolha do Pedro, da sua Mulher...
//result: ACANDIDATE_POSS(sua, Pedro)
|?*, PP#1, PP{?*, pron#2[poss=+]}|
  if(HEAD(#3,#1) &
    ^ACANDIDATE_POSS(#2,#3)
  )
  ACANDIDATE_POSS(#2,#3)

//rule 3.3 - F3: Anaphor is possessive pronoun in PP, Antecedent in NP
//example: A casa do seu filho.
//result: INVALID_ACANDIDATE(seu, filho)
|?*, NP#1, PP{?*, pron#2[poss=+]}|
  if(HEAD(#3,#1) &
    ^INVALID_ACANDIDATE(#2,#3)
INVALID_ACANDIDATE(#2,#3)

//
// 4 - Immediate reference
//
// rule 4.1 -
//example: O Pedro pode esperar pelo Zé ou ligar-lhe.
//result: IMMEDIATE_REFERENCE(lhe, Zé)

[?*, VINF*, PP#1{?*,noun#4}, CONJ, VINF, NP#2{pron#3[pers]}]
if(HEAD(#4,#1) && HEAD(#3,#2) && 
  ~IMMEDIATE_REFERENCE(#3,#4) &
  #3[number]:#4[number] & #3[gender]:#4[gender])
IMMEDIATE_REFERENCE(#3,#4)

// rule 4.2 -
//example: Eu queria falar com a Maria e liguei-lhe.
//result: IMMEDIATE_REFERENCE(lhe, Maria)

[?*, NP*, VF*, VINF*, PP#1{?*,noun#4}, CONJ, VF, NP#2{pron#3[pers]}]
if(HEAD(#4,#1) && HEAD(#3,#2) && 
  ~IMMEDIATE_REFERENCE(#3,#4) &
  #3[number]:#4[number] & #3[gender]:#4[gender])
IMMEDIATE_REFERENCE(#3,#4)

// rule 4.3 -
//example: Tira o bolo e embala-o
//result: IMMEDIATE_REFERENCE(a, televisão)

[?*, NP*, VF, NP#1{?*,noun#4}, CONJ, NP*, VF, NP#2{pron#3[pers]}]
if(HEAD(#4,#1) && HEAD(#3,#2) && 
  ~IMMEDIATE_REFERENCE(#3,#4) &
A.4. POSSESSIVE PRONOUNS

#3[number]:#4[number] & #3[gender]:#4[gender])
IMMEDIATE_REFERENCE(#3,#4)

//rule 4.4 -
//example: Tira o bolo, embala-o.
//result: IMMEDIATE_REFERENCE(o, bolo)
|?*, NP*, VF, NP#1{?*,noun#4}, PUNCT, NP*, VF, NP#2{pron#3[pers]}|
if(HEAD(#4,#1) && HEAD(#3,#2) &&
   ~IMMEDIATE_REFERENCE(#3,#4) &&
   #3[number]:#4[number] & #3[gender]:#4[gender])
IMMEDIATE_REFERENCE(#3,#4)

//rule 4.5 -
//example: Tira-o e embala-o.
//result: IMMEDIATE_REFERENCE(o, o)
|?*, NP*, VF, NP#1{?*,pron#4}, CONJ, NP*, VF, NP#2{pron#3[pers]}|
if(HEAD(#4,#1) && HEAD(#3,#2) &&
   ~IMMEDIATE_REFERENCE(#3,#4) &&
   #3[number]:#4[number] & #3[gender]:#4[gender])
IMMEDIATE_REFERENCE(#3,#4)

//rule 4.6 -
//example: Tira o bolo, embala-o.
//result: IMMEDIATE_REFERENCE(o, o)
|?*, NP*, VF, NP#1{?*,pron#4}, PUNCT, NP*, VF, NP#2{pron#3[pers]}|
if(HEAD(#4,#1) && HEAD(#3,#2) &&
   ~IMMEDIATE_REFERENCE(#3,#4) &&
   #3[number]:#4[number] & #3[gender]:#4[gender])
IMMEDIATE_REFERENCE(#3,#4)