Anaphora Resolution

Nuno Nobre

IST – Instituto Superior Técnico
L^{2}F – Spoken Language Systems Laboratory – INESC ID Lisboa
Rua Alves Redol 9, 1000-029 Lisboa, Portugal
nuno.nobre@ist.utl.pt

Abstract. 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%.

Keywords Anaphora resolution, Pronominal resolution, Natural Language processing systems, Information retrieval

1 Introduction

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.

(1.1) *Obama* foi laureado com o Nobel da Paz. *O Presidente dos Estados Unidos* foi este ano o vencedor do Prémio Nobel da Paz.

Obama was awarded the Peace Nobel. *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.

2 Related Work

2.1 Mitkov's Anaphora Resolution System

Motivated by the need of a robust, real world operating algorithm, Ruslan Mitkov [Mitkov2002] developed a knowledge-poor approach for pronominal 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 antecedent.

2.1.1 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.

It is possible to identify five main phases in MARS algorithm:

1. The text to be processed is parsed syntatically, using Conexor's FDG Parser [Tapanainen and Järvinen1997], 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 extrated from the NPs in the heading of the subsection in which the pronoun appears. and from NPs in the current and preceding two sentences (if available) whithin 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.

Evaluation MARS was tested on a set of technical manuals, with 247 401 words and 2263 anaphoric pronouns, intrasentential and intersentential. Considering the pre-processing errors the average success rate was 92.27%.

3 Architecture

The work presented in this Section receives the output of the Xerox Incremental Parser processing chain [Xerox2001] integrated at the L2F.

3.1 Xerox Incremental Parser

The Xerox Incremental Parser (XIP) is a text parser that produces annotated text with relevant morphossyntactic 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, organisations;

3.1.1 Dependency Rules It is possible to implement dependency rules to locate and extract information from texts using XIP. This acquires great importance in anaphora resolution, as many times recognizable patterns that evidence the existence of anaphora phenomenon occur in texts.

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 work in [Paraboni and Lima1998], token *1* is the anaphor of token *2*;
- *INVALID_ACANDIDATE(1,2)*: according to the work in [Paraboni and Lima1998], token *1* cannot be the anaphor of token *2*;

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 intersentencial anaphora a multi-sentence analysis is required, which is out of reach of XIP processing chain.

3.2 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.

4 Implementation

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

The Anaphora Resolution Module operates on three steps:

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

4.1 Anaphor Identification

Are identified as possible anaphors personal, possessive, relative and demonstrative pronouns of the third person, both singular and plural. All pronouns must be a head of a phrase.

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

(4.1) O *Filipe* viu-*se* ao espelho.

Filipe 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 *Filipe*. 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 *algu* (*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.

4.2 Antecedent Candidates Identification

After identifying an anaphor is time to find its antecedent candidates. 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 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 the subject of a sentence. In this case the number agreement constraint should hold, but in some cases the gender agreement would not hold.

(4.4) O *Manuel* e a *Ana* foram ao cinema. *Eles* gostam de filmes.

Manuel and *Ana* went to the cinema. *They* like movies.

In example (4.4), the pronoun *Eles* is the anaphor of *Manuel* 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 1 distinguishes the pronoun genders for all the possible cases.

	All nouns feminine	All nouns masculine	At least one noun masculine
pronoun gender	feminine	masculine	masculine

Table 1. Pronoun gender for coordinated NPs

Possesive Pronouns In Portuguese possessive pronouns do not hold gender or number agreement with their antecedents. This agreement occurs with the object of the sentence.

(4.5) O *Vitor* nao encontra as *suas* sapatilhas.

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 *sapatilhas* (*sneackers*) it determines.

4.3 Choosing the Antecedent Candidate

Once the anaphor is identified and the antecedent candidates are chosen, it is time to determine which one is the anaphor's antecedent. To perform this task, a set of parameters is used to score each candidate, according to their syntactic role in the analysed text.

These parameters were chosen based on Mitkov [Mitkov2002] and Chaves and Rino [Chaves and Rino2008] work. The rules defined in [Paraboni and Lima1998] allowed the creation of two more parameters: *PPPC* and *PPIC*. All implemented parameters and values are listed bellow:

- *First Noun Phrase (FNP)*: a score of +1 is assigned to the firstt 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 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 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 with 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 +1 is assigned to the candidate if is present on an *ACANDIDATE_POSS(A,C)* 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)* relation for anaphor "A" ;

4.4 XIP Interaction

The Anaphora Resolution Module operates on XIP's processing chain result, specially on the chunk trees and dependency relations extracted from corpora.

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.

4.5 XIP API

The XIP API implements the following concepts:

– 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;
– 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*, aswell itermediate elements, e.g. *NOUN* nodes, or leafs ones, e.g. *tokens*.

4.6 Anaphora Resolution Module

Domain The API presented in Section 4.5 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.

Algorithm During the resolution process, four main phases take place:

1. Dependency relations analysis;
2. Tree exploration analysis;
3. Post exploration analysis;
4. Document exportation.

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.

Items 2 and 3 are evaluated according to Section 4.2. Finally the candidate is evaluated by the parameters defined in Section 4.3 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 neares NP.

The document tree is then exported in XML format.

5 Evaluation

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 cames 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.

5.1 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.

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.

5.2 Final evaluation

To perform the system final evaluation, a corpora made available by the L2F was used. This way, during the development phase, there was no knowledge about the used corpora, making this phase independent from the final evaluation.

The evaluation corpora was composed by 20 texts of distinc 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.

The system achieved the following results:

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

6 Conclusion

The Natural Language Processment 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 texts annotation.

The final product is a system that proposes to resolve 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.

References

[Chaves and Rino2008] Chaves, A. R. and Rino, L. H. M. (2008). The mitkov algorithm for anaphora resolution in portuguese. In *PROPOR 2008*.

[Mitkov2002] Mitkov, R. (2002). *Anaphora Resolution*. Pearson Education.

[Paraboni and Lima1998] Paraboni, I. and Lima, V. L. S. (1998). Possessive pronominal anaphor resolution in portuguese written texts. In *Proceedings of the 17th international conference on Computational linguistics*, pages 1010–1014, Morristown, NJ, USA. Association for Computational Linguistics.

[Tapanainen and Järvinen1997] Tapanainen, P. and Järvinen, T. (1997). A non-projective resolution methods. In *Proceedings of the 5th Conference of Applied NAtural Language Processing (ANLP-5)*, pages 64–71, Washington, DC.

[Xerox2001] Xerox, R. C. E. (2001). Xerox incremental parser – user guide.