Extraction and Classification of Named Entities

Diogo Correia de Oliveira

MSc in Information Systems and Computer Engineering
IST – Instituto Superior Técnico
L^{2}F – Spoken Language Systems Laboratory – INESC ID Lisboa
dcoliveira@gmail.com

1 Introduction

BEING able to program a computer to fully understand our language has been an unfulfilled dream for many years in the scientific community. The computer science field of Natural Language Processing (NLP) is vast and is concerned with how computers interact with human beings, in particular by means of human (natural) languages. Named Entity Recognition (NER) is the NLP task that focuses on locating and classifying entities in a given text. For example, consider the following sentence: "James Cameron's latest movie premiered in December 2009 in the USA and its budget exceeded 200 million dollars." A system capable of performing NER should be able to *identify* four named entities (NEs) in this example: (a) *James Cameron*; (b) *December 2009*; (c) *USA*, and (d) *200 million dollars*. Furthermore, it should be able to *classify* them according to established criteria. A possible scenario is to classify *James Cameron* as `HUMAN`, *December 2009* as `TIME`, *USA* as `LOCATION` and *200 million dollars* as `AMOUNT`.

NER is important because it is one of the first steps towards extracting meaning out of text. The most common approaches that NER systems use are grammar-based techniques and statistical models. Typically, systems that use the former obtain better precision but at the cost of a lower recall. Moreover, these systems are often handcrafted by a team of computer engineers and computational linguists for a long time, in what is a slow, costly and time-consuming process. On the other hand, statistical NER systems usually require a large amount of manually annotated training data. The task of annotating corpora is also a costly and time-consuming process. Unfortunately, even the best NER systems are fragile, because even though they may have been built for one specific domain, they do not typically perform well on other domains (Poibeau & Kosseim [8]).

1.1 Goals

This thesis has continued the development and has increased the efficiency of a NER system called XIP (Xerox Incremental Parser).

The main goals of this thesis are: (a) to improve XIP by adding more lexical entries, by correcting rules and by adding new rules, specifically in the `AMOUNT`, `HUMAN` and `LOCATION` categories; (b) to establish a new way of presenting metonymy (see Section 4.5) and also to improve XIP's capacity for capturing these kinds of entities; (c) to contribute to the creation of a new set of directives for Portuguese texts, thereby replacing the ones that were used in the HAREM evaluation campaign, and (d) to evaluate the work by using well defined evaluation metrics such as precision and recall.

The overall results of the evaluation of this work were satisfactory. Even if they can not be directly compared with those from the Second HAREM, since the directives are

different, it is possible to say that the main objective of this thesis has been achieved: results seem to show a general trend of improvement.

2 State of the Art

The HAREM evaluation campaign brought together some systems that are involved in the NER task regarding the Portuguese language. Their common goal is to correctly identify and classify entities in any given text. The main purpose of this section is to provide a general view of the technologies that were used by the different systems and a brief summary of the results that were achieved.

The three main metrics used in this evaluation campaign were precision, recall and F-measure. Precision is a measure of the system's response quality and it measures the amount of right answers among all answers given by the system. Recall, on the other hand, measures the amount of right answers among all possible answers (not only those given by the system). Finally, F-measure combines precision and recall according to the following mathematical formula:

$$\text{F-measure } (\%) = \frac{2 \times \text{ Precision} \times \text{ Recall}}{\text{Precision } + \text{ Recall}} \times 100.$$

Table 1 presents a list of the systems and the technology they use in order to perform NER. As easily noticed, one aspect they all share is the use of manual rules. As time-consuming as they may be, they are still nowadays one of the most effective methods of writing disambiguation and context capture rules, and patterns. Even systems that use other knowledge sources, such as Wikipedia or ontologies, still need to rely, on some level, on manual rules in order to increase precision and recall. REMMA, for example, specifically tried to determine if the use of Wikipedia alone would improve the overall results and reached the conclusion that it did not. A hybrid solution is far more effective.

	Technologies		
System	Manual rules	Automatic rules	Other sources
CaGE	✓		Dictionaries, Gazetteer
PorTexTO	✓		
Priberam	✓		Ontology, Lexicon, Grammar
R3M	✓	✓	
REMBRANDT	✓		Wikipedia
REMMA	✓		Lists of words, Wikipedia
SEI-Geo	✓		Ontologies
XIP	✓		Lexicons, Lists of words, Grammars

Table 1. State of the Art: Comparison (Systems and Technologies).

HAREM's evaluation directives (Oliveira *et al.* [6]) are comprised of global (all entities are considered) and selective scenarios (each one combines some of the entities). Globally, Priberam is the best system both in the identification and classification tasks, having reached the highest F-measure result in both (approximately 71% and 57%, respectively). It is also the only system that was able to achieve a higher recall than precision in the identification task.

However, it is also important to notice that Priberam is not the most precise system among the participants. That award goes to SEI-Geo, which was able to identify 90% of all entities and classify 75%, even though it only deals with `LOCATION` entities. For this reason, SEI-Geo's recall is as low as 12%, which inevitably pushes this system to the bottom of the table.

A more realistic comparison is to consider Priberam, REMBRANDT and REMMA, whose common attempt to identify and classify all kinds of entities is an ambitious one. Although in this global scenario both REMBRANDT and REMMA are more precise than Priberam in the identification task (1 to 6% more precise), this difference is almost residual when compared to their recall differences in the same task: Priberam is 10 to 27% better.

Typically the presence of a low recall can be explained by two reasons: (a) the system is built to deal with a small set of all possible entities, or (b) some parts of the system are underdeveloped, either from lack of time or resources. PorTexTO, SEI-Geo and CaGE fall into the former, whereas REMMA and XIP fall into the latter. In XIP's case, there was not enough time to add as many lexical entries as desired. Consequently, XIP's results are very diverse.

On the one hand, XIP is the best system to classify temporal expressions, mainly because a lot of effort was put into improving the results on that particular category. On the other hand, the `AMOUNT` category proves to be one of XIP's weaknesses, for the F-measure was 42%, which is very low for a category whose excellent results are typical: Bick [1] reached a 95–97% F-measure during the First HAREM. Globally, XIP was the third best system, being only surpassed by Priberam, R3M and REMBRANDT (these two tied in second place), and at a low percentage distance from them (7–9% in identification and 2–3% in classification).

3 Architecture

3.1 Processing chain

L^2F has developed a processing chain that is comprised of several modules (see Figure 1). The chain is divided in three main stages: pre-processing, disambiguation (rule-driven and statistical), and syntactic analysis.

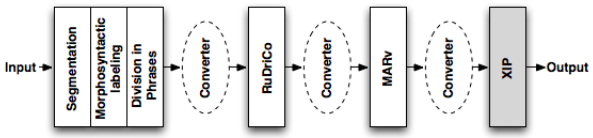


Fig. 1. XIP: The processing chain in which the system resides (from Romão [10, Section 3, Figure 3.1]).

3.1.1 Pre-processing

The pre-processing is comprised of three modules. The first one, the segmentation module, is mainly responsible for dividing the input into individual segments, also known

as “tokens”. Besides this, the module is also responsible for the early identification of certain types of NEs, namely: email addresses, ordinal numbers, numbers with “.” and “,”, IP and HTTP addresses, integers, several abbreviations with “.”, sequences of interrogation and exclamation marks, as well as ellipsis, punctuation marks, symbols and Roman numerals.

According to Mamede [4], one of the problems regarding this module was that it did not perform the identification of numbers written in full, such as “duzentos e trinta e cinco” (two hundred and thirty-five). This, as explained in Section 4.1, has been corrected and the identification is now fully operational.

Afterwards, the segmentation module’s output tokens are tagged with POS (part of speech) labels, such as “noun”, “verb” or ”adjective”, among others. This tagging is performed by the Palavroso system (Medeiros [5]), which is very old and needs replacement or improvement.

The final step of the pre-processing stage is the text division into sentences.

3.1.2 Disambiguation

The next stage of the processing chain is the disambiguation process, which is comprised of two steps: rule-driven morphosyntactic disambiguation, performed by RuDriCo (Pardal [7]), and statistical disambiguation, performed by MARv (Ribeiro *et al.* [9]).

According to Pardal [7], RuDriCo’s main goal is to provide for an adjustment of the results produced by a morphological analyzer to the specific needs of each parser. In order to achieve this, it modifies the segmentation that is done by the former. For example, it might contract expressions provided by the morphological analyzer, such as “ex-” and “aluno”, into one segment: “ex-aluno”; or it can perform the opposite and expand expressions such as “nas” into two segments: “em” and “as”. This will depend on what the parser might need. RuDriCo can also be used to solve (or introduce) morphosyntactic ambiguities.

Immediately after, MARv is used to analyze the labels that were attributed to each token in the previous step of the processing chain, and then choose the most likely label for each one. In order to achieve this, it employs the statistical model known as Hidden Markov Model (HMM). A HMM is a very important machine learning model in speech and language processing and it allows one to talk about both observed events (like words that we see in the input) and hidden events (like POS tags). There are many algorithms to compute the likelihood of a particular observation sequence. MARv uses the Viterbi algorithm, which can be analyzed in Jurafsky & Martin [3, see Ch. 6, Sec. 6.4].

3.1.3 Syntactic analysis

The third and final stage of the processing chain is the syntactic analysis performed by XIP. This is where the identification and classification of NEs occurs, and where the major work done for this thesis has taken place. XIP is a language-independent parser that takes textual input and provides linguistic information about it. XIP can modify and enrich lexical entries, construct chunks and other types of groupings, and build dependency relationships (Xerox [11]).

3.2 Features, Chunks, Dependencies, Lexicons and Grammars

XIP receives as input the converted data from MARv and is able to handle it in order to perform several tasks, namely: calculation of chunks and dependencies; adding lexical,

syntactic and semantic information; applying morphosyntactic disambiguation rules, and applying local grammars.

3.2.1 Features

The fundamental data representation unit in XIP is the node. A node has a category, feature-value pairs and "brother" nodes. Every node category and every feature must be declared in declaration files. Furthermore, features must be declared with their domain of possible values. They are an extremely important part of XIP, as they describe the properties of nodes. For example, the node below represents the noun "Diogo" and it has several features that are used as a means to express its properties:

```
Diogo: noun[people,individual,proper,firstname,sg,masc,maj]
```

3.2.2 Chunking rules

Chunking is the process by which sequences of categories are grouped into structures; this process is achieved through chunking rules. There are two types of chunking rules (Xerox [11]): immediate dependency and linear precedence rules (ID/LP rules), and sequence rules.

The first important aspect about chunking rules is that each one must be defined in a specific layer. This layer is represented by an integer number, ranging from 1 to 300, followed by the symbol ">". ID/LP rules are significantly different from sequence rules. While ID rules describe unordered sets of nodes and LP rules work with ID rules to establish some order between the categories, sequence rules describe an ordered sequence of nodes. Consider the following example of an ID rule:

```
1> NP -> det, noun, adj.
```

This rule is interpreted as follows: "whenever there is a sequence of a determiner, noun and adjective, regardless of the order in which they appear, create a Noun Phrase (NP) node". Obviously, this rule applies to more expressions than those desirable, e.g. "o carro preto" (the car black), "o preto carro" (the black car), "preto carro o" (black car the) and "carro preto o" (car black the). This is where LP rules come in. By being associated with ID rules, they can apply to a particular layer or be treated as a general constraint throughout the XIP grammar:

```
1> [det:+]  < [noun:+].
1> [noun:+] < [adj:+].
```

Thus, by stating that a determiner must precede a noun only in layer 1, and that a noun must precede an adjective also only in layer 1, the system is now setting constraints in this layer, which means that expressions such as "o preto carro" (the black car) will no longer be allowed. However, "o carro preto" (the car black) will.

The other kind of chunking rules, sequence rules, though conceptually different because they describe an ordered sequence of nodes, are almost equal to the ID/LP rules in terms of syntax. There are, however, some differences and additions: sequence rules do not use the `->` operator. Instead, they use the `=` operator, which matches the shortest possible sequence. In order to match the longest possible sequence, the `@=` operator is used. Also, there is an operator for applying negation (`~`) and another for applying disjunction (`;`).

3.2.3 Dependency rules

Dependency rules take the sequences of constituent nodes identified by the chunking rules and identify relationships between them (Xerox [11]). This section presents a brief overview of their syntax, operators, and some examples.

```
|pattern| if <condition> <dependency_terms>.
```

The `pattern` contains a Tree Regular Expression (TRE) that describes the structural properties of parts of the input tree. The `condition` is any Boolean expression supported by XIP, and the `dependency_terms` are the consequent of the rule. The first dependency rules to be executed are the ones that establish the relationships between the nodes, as seen in the next example:

```
| NP#1{?*, #2[last]} |
    HEAD(#2, #1)
```

This rule identifies HEAD relations, for example "a bela rapariga" (the beautiful girl) ⇒ HEAD(rapariga,a bela rapariga).

3.2.4 Lexicons and Grammars

XIP allows the definition of custom lexicons (lexicon files), which add new features that are not stored in the standard lexicon. Having a rich vocabulary in the system can be very beneficial for improving its recall. In XIP, a lexicon file begins by simply stating `Vocabulary:`, which tells the XIP engine that the file contains a custom lexicon. Only afterwards come the actual additions to the vocabulary. The lexical rules attempt to provide a more precise interpretation of the tokens associated with a node (Xerox [11]).

Local grammars are text files that contain chunking rules and each file may contain ID/LP and sequence rules. Essentially, we use different local grammar files to capture desirable sequences of nodes and to attribute features to them. We employ a division based on different categories of NEs. For example, whereas the file `LGLocation` is aimed at capturing sequences of nodes related to the `LOCATION` category, the file `LGPeople` will capture sequences of nodes related to the `INDIVIDUAL` type (`HUMAN` category).

4 Improvements

4.1 Segmentation

At the Segmentation stage, there is a script written in Perl that is responsible for early identification of standard NEs (see Section 3.1.1). Numbers written in full, such as "trezentos e quarenta e dois" (three hundred and forty-two), however, were not being captured. Consequently, their capture was being delayed until a later stage, in which rules joined the various tokens to form a single one, which was the number itself.

The script has been improved in order to support detection of numbers written in full, ranging from 0 (zero) to 999.999.999 (novecentos e noventa e nove milhões, novecentos e noventa e nove mil, novecentos e noventa e nove). The script now successfully detects all numbers written in full (case-insensitive), and with different writing styles. For example, the number 1.101 (one thousand one hundred and one) is read: "mil cento e um", but can be written as mentioned, or "mil, cento e um", or "mil e cento e um", or "mil, e cento e um", and so on.

4.2 Consistency

Since XIP is inserted in a long processing chain, errors may appear due to inconsistencies between the several modules. One of the most common among them arises when a particular rule in XIP is expecting a lemma for a token, but over time that lemma has been changed and so the rule is not matched. Lemmas are often changed as a result of improvements in RuDriCo. For example, at one point in time XIP may have a rule to capture "Presidente da República" (President of the Republic) token by token, but if RuDriCo's rules are improved and "Presidente da República" becomes an unique token, XIP's rule will not match.

This task has been extremely important in order to guarantee that not only XIP's rules can keep up with RuDriCo's (according to Diniz [2], RuDriCo 2.0 currently has 28.733 contraction rules, 9,3 times more than one year ago), but also that the overall results are not compromised.

4.3 Classification directives

A new set of NE classification directives was developed, different from (although inspired by) the Second HAREM evaluation campaign. The main changes were introduced in the `ORGANIZATION` and `PERSON` categories, having created a top category, `HUMAN`, which divides into two main types: `INDIVIDUAL` and `COLLECTIVE`. The former treats people's names and jobs, whereas the latter treats administrations (such as Governments), institutions and other groups (such as musical groups).

The `LOCATION` category remained practically unchanged, with the exception of its `VIRTUAL` type, which now only has two subtypes: `SITE`, for capturing websites and the like, and `DOCUMENTS`, for capturing entities like Laws, Regulations, etc.

Many other things have also been changed, such as the delimitation of certain named entities, especially in the `AMOUNT` category, or the inclusion of a new XML tag for representing metonymy. A new type has also been created under the `AMOUNT` category: `SPORTS RESULTS`.

4.4 Lexicon and grammar improvement

All categories (`AMOUNT`, `HUMAN` and `LOCATION`) have been improved in terms of lexicon and grammar. These changes include: increased vocabulary and rules; correction of chunking rules (see Section 4.2); new dependency rules. For example, regarding the `AMOUNT` category, there has been a major restructuring of the lexical files, having added a total of 212 lexical entries in areas such as units of frequency, volume, length and mass and 1.031 new currency expressions; also, the delimitation of `AMOUNT` named entities has changed by having altered dependency rules.

The `HUMAN` category has seen many improvements, one of which is particularly important: the way people's names are captured. The strategy that was being used was to mark the start and end of a name through the use of features: `start_people` and `end_people`, respectively. This strategy, however, was somewhat limited because more complex names have prepositions followed by articles ("de", "da", "das", "do", "dos") and conjunctions ("e") and, as a result, the resulting tree would consist of several separate nodes instead of one node. Since XIP does not allow the partial classification of a node, but only of the whole node itself, these names would be classified separately, and not as an unique name. In order to correct this, we have improved the strategy: instead of simply marking the start and end of a name, we now also mark the places where a

name continues: the prepositions and conjunctions. We have done this by adding a third feature: `cont_people` (from "continue" people). Consequently, we are now able to produce a single node that represents the whole name, even if it is long and complicated.

Moreover, we have added a very large number of new lexical entries to the `HUMAN` category: 3.272 names and surnames from 23 different languages.

4.5 Metonymy

One of the improvements that has been made in the course of this thesis has been to enhance the detection and classification of metonymy and to provide a clearer way of presenting it through the inclusion of the `MET-CAT` tag. Before this was implemented, there were many cases of metonymy that were being captured, but they were spread over several categories. As a result, metonymy passed almost unnoticed in the system, because it was not being explicitly represented.

Currently, the rules that exist in the system to treat metonymy with the `MET-CAT` tag are based on the relations between the various elements of the sentence (relations-based rules). Unlike syntax-based rules, relations-based rules grasp the (deeper) connection between the sentence's elements. Consider, for example, the sentence: "o Brasil, irritado com Portugal, não votou a favor da moção" (Brazil, angry with Portugal, did not vote in favor of the motion). In this case, it is important to capture and treat the `SUBJ` relation between "Brasil" (Brazil) and "votar" (to vote), because a syntax-based rule will surely not match this sentence, due to the separation that exists between the subject and the verb. The unpredictability of how the sentence is constructed makes it impractical to treat metonymy in this way.

Before this thesis, most of the existing rules were based on the relations between the elements of the sentence. Moreover, all NEs were being classified according to the HAREM's directives. Therefore, on the one hand, we have tried to convert the rules that already existed in order to comply with the new set of directives; on the other hand, we have created new rules for dealing with metonymy, which were syntax-based in the beginning, but that have been gradually converted into relations-based rules.

5 Evaluation

5.1 Context

The typical evaluation methodology followed in this kind of NLP task consists of processing a large amount of non annotated corpus with the system that is to be evaluated, and afterwards compare the answers with the (previously) annotated version of the same corpus. The evaluation corpus (henceforth called "Golden Collection" (GC)) is the same that was used in the HAREM evaluation campaign, but it was re-annotated following the new set of directives, instead of the directives of HAREM. For all intents and purposes, this new GC has two versions: one in which there are entities marked for metonymy, and another in which metonymy is left out. This was meant to evaluate the impact that metonymy induces in our system.

5.2 Evaluation metrics

The GC that was used in this evaluation consists of 129 documents, 2.274 paragraphs and 5.569 named entities (out of which 283 are metonymy NEs). There are some important concepts in this evaluation that need to be understood before analyzing the results,

namely the possible states for a NE: correct, missing or spurious. As far as the evaluation programs are concerned, a named entity can be either correct, missing, or spurious. A NE is *correct* if it is equal to the one in the GC: for identification purposes, this means that the NE must contain the same elements; for classification purposes, this means that it was accorded the same category, type and subtype. A NE is *missing* if there is a NE in the GC but the system fails to correctly detect any of its elements. Finally, a NE is *spurious* if the system marks a NE that does not exist in the GC.

The three metrics that have been used both in the HAREM evaluation campaign and in this thesis are *precision*, *recall* and *F-measure*, which have been defined in Section 2.

5.3 Evaluation results

There are two types of scenarios in the evaluation of this thesis: global and selective scenarios. Whereas the former consists of evaluating all categories, the latter consists of evaluating a subset of the categories.

Regarding the identification task, the `AMOUNT` category is the only one in which XIP identified more entities than those present in the GC (out of 546 entities in the GC, XIP identified 852: 367 were correct, but 485 were spurious). The reasons for this large number of spurious NEs differ in nature, but the most significant cause were just plainly misidentifications, like in the sentence "O vírus H5N1" (the H5N1 virus).

In contrast, the `HUMAN` category suffers from the inverse problem: a large number of missed NEs (recall = 51.6% in Relaxed ALT), but also an important quantity of spurious, i.e., misidentified NEs (36.3%).

The `LOCATION` category produced the best results in the identification task, with 82.8–83.7% precision and 62.7% recall, thus yielding an F-measure of 71.4–71.7%.

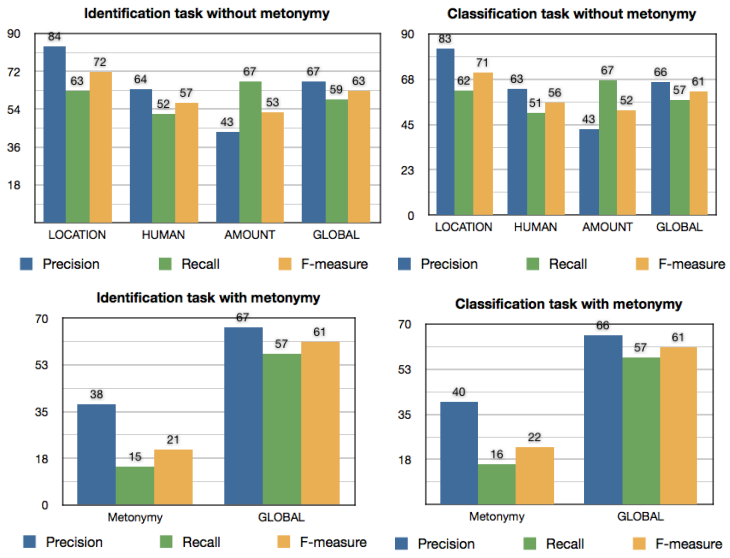
Finally, the `GLOBAL` scenario produced balanced results, with an overall precision of 66.4% in the Strict ALT scenario and 67.3% in the Relaxed ALT scenario. The recall was 58.6% and 58.8% respectively and the final F-measure was 62.3% and 62.8%. While not directly comparable, these results apparently indicate an improvement trend in the performance of the system since the Second HAREM.

Regarding the classification task, the `AMOUNT` category remains the one with most problems, which in this case is shown by the system's maximum score (1171.5 out of a possible 747.75). This is a direct consequence of the large number of spurious entities: with more entities to analyze, the system's maximum score increases to the point where it may even exceed the GC's maximum possible score, which in this case does. As a consequence, the precision is also drastically reduced (42.7%).

Globally, the results are satisfactory, especially regarding recall: 57.2% in the Strict ALT scenario and 57.4% in the Relaxed ALT scenario. This means that out of 10 possible solutions in the GC, XIP is able to retrieve almost 6. Simultaneously, out of 10 given answers, almost 7 are correct (65.1% and 66.0% of precision). The factors that continue to diminish the final results are the precision of the `AMOUNT` category and the recall of the `HUMAN` category.

Finally, regarding the identification and classification of metonymy: globally, the identification task with metonymy presents (as expected) worse results than without metonymy, even if the difference seems small, because of the limited number of cases affected by metonymy and here treated. Everything needs to be improved: from the amount of spurious entities (68 out of 109, which results in a 37.6% precision) to the amount of missed entities (242 out of 283, which results in a recall of only 14.5%), there is still much room for improvement.

Below there are four charts that sum up the results here discussed.



6 Conclusions

Named Entity Recognition is one of the most known tasks in Natural Language Processing. This study aimed at improving the performance of a NLP system developed at L^2F/INESC-ID Lisboa, by developing the NER modules responsible for the identification and classification of NEs from the HUMAN (INDIVIDUAL and COLLECTIVE), LOCATION and AMOUNT categories.

Section 2 carried out a comparison of the 8 systems that took part in the Second HAREM evaluation campaign. This evaluation initiative was a collective effort of the Portuguese NLP community aimed at assessing the performance of those different systems in the complex task of identifying and subsequently classifying named entities in Portuguese texts.

In Section 3, one of these systems was analyzed in more detail: XIP, a language-independent parser, which takes textual input and provides linguistic information about it. The NLP chain, in which XIP is inserted, was described, covering its three main stages: pre-processing, disambiguation and syntactic rules. Then, the XIP's main characteristics were detailed: features, chunking rules, dependency rules, disambiguation rules, lexicons and local grammars.

Section 4 presented the improvements made to XIP during this study. In particular, this section showed how each category was improved either by adding more lexical entries or by correcting/adding chunking/dependency rules.

Finally, Section 5 presented the evaluation of the NLP chain in the NER task after all these improvements were introduced. The overall results were satisfactory, particularly for the LOCATION category, where above 70% F-measure was attained, both in the identification and classification tasks. It is possible to say that the main objective of this thesis has been achieved: results seem to show a general trend of improvement.

Bibliography

[1] BICK, ECKHARD. 2007. Functional aspects on Portuguese NER. *Chap. 12, pages 145–155 of:* SANTOS, DIANA & CARDOSO, NUNO (eds), *Reconhecimento de entidades mencionadas em português – Documentação e actas do HAREM, a primeira avaliação conjunta na área*. Digitally published.

[2] DINIZ, CLÁUDIO. 2010. *RuDriCo 2 - A converter based on declarative transformation rules*. Master thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

[3] JURAFSKY, DANIEL & MARTIN, JAMES H. 2008. *Speech and Language Processing: An introduction to natural language processing, computational linguistics, and speech recognition*. 2 edn. Prentice Hall.

[4] MAMEDE, NUNO. 2009. *A cadeia de processamento de Língua Natural do L^2F (em Dezembro de 2009)*. Technical report. Laboratório de Sistemas de Língua Falada (L^2F), INESC-ID Lisboa.

[5] MEDEIROS, JOSÉ CARLOS. 1995. *Processamento Morfológico e Correcção Ortográfica do Português*. Master thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

[6] OLIVEIRA, HUGO GONÇALO; MOTA, CRISTINA; FREITAS, CLÁUDIA; SANTOS, DIANA & CARVALHO, PAULA. 2008. Avaliação à medida do Segundo HAREM. *Chap. 5, pages 97–129 of:* MOTA, CRISTINA & SANTOS, DIANA (eds), *Desafios na avaliação conjunta do reconhecimento de entidades mencionadas: O Segundo HAREM*. Digitally published.

[7] PARDAL, JOANA PAULO. 2007. *Manual do Utilizador do RuDriCo*. Technical report. Laboratório de Sistemas de Língua Falada (L^2F), INESC-ID Lisboa.

[8] POIBEAU, THIERRY & KOSSEIM, LEILA. 2000. Proper Name Extraction from Non-Journalistic Texts. *Language and Computers, Computational Linguistics in the Netherlands*, 144–157.

[9] RIBEIRO, RICARDO DANIEL; OLIVEIRA, LUÍS C. & TRANCOSO, ISABEL. 2003. Using Morphosyntactic Information in TTS Systems: Comparing Strategies for European Portuguese. *Pages 143–150 of: PROPOR'2003 - 6th Workshop on Computational Processing of the Portuguese Language*. Heidelberg: Springer-Verlag.

[10] ROMÃO, LUÍS. 2007. *Reconhecimento de Entidades Mencionadas em Língua Portuguesa: Locais, Pessoas, Organizações e Acontecimentos*. Master thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

[11] XEROX. 2003. *Xerox Incremental Parser – XIP Reference Guide*. Xerox Research Centre Europe.