Combining Rule-based and Statistical Methods for Named Entity Recognition in Portuguese

Eduardo Ferreira,[1] João Balsa,[1] António Branco[1]

[1]University of Lisbon, NLX–Natural Language and Speech Group
Faculdade de Ciências, Dep. Informática, Campo Grande, 1749-016 Lisboa, Portugal

{eferreira,jbalsa,antonio.branco}@di.fc.ul.pt

Abstract. *We present and discuss a tool for the recognition of expressions for named entities in Portuguese that resorts to a rule-based approach when dealing with numbers, measures, time and addresses, and uses a hybrid approach when dealing with names. The expressions for named entities are delimited and semantically classified by a XML-like markup. Evaluation results are presented.*[1]

1. Introduction

Named entity recognition (NER) is an important task in several aspects of the processing of information conveyed by natural languages. It becomes a key component when integrated into technologies that deal with large amounts of information, such as information retrieval and extraction systems, automatic summarization, machine translation or the annotation of corpora. More recently, automatic question answering has also become more dependent on high-performance named entity recognition tools.

The main goal of the NER recognizer we developed (LXNER, `http://lxner.di.fc.ul.pt`) was the recognition of expressions for named entities in Portuguese. The developed recognizer takes a segment of Portuguese text and identifies, circumscribes and classifies the expressions for named entities it contains. Furthermore, each expression is given a standard representation and embedded into an XML-like markup that retains all the crucial information about the named entity in question. To get a glimpse of the adopted markup, see the example at the end of Section 2.2.

The LXNER was developed under the scope of projects carried out at the NLX Group, including the TagShare [Barreto et al. 2006], QueXting and LT4eL projects. Unlike previous work whose main focus was the detection and classification of proper names into a limited set of predefined categories, we set out to fully expand and refine this set of categories, thus introducing an appropriate level of detail into the classification stage.

The five main categories, or types, of named entities we considered are related to numbers, measures, time, addresses and names. An initial thorough analysis of each of these categories allowed us to realize that different approaches should be used in order to maximize the performance of the recognizer.

The first four, number-based, categories mentioned above seemed to have relatively fixed structures, leading us into choosing a rule-based approach to handle them. Regarding the remaining, name-based, category, the expressions under its scope proved to have a much more free-form structure, and consequently more difficult to predict through

[1] We would like to thank the reviewers for their comments and suggestions.

hand-built rules. As an example, even with more complex rules, the identification of named entities representing events or works would be unfeasible (for instance, the title of a movie can be almost anything). This difficulty drove us to approach this type of expressions also with a stochastic method.

In Section 2, a description of the rule-based component of the recognizer is given, along with a full reference to the covered types. Section 3 contains an explanation of its hybrid component and the issues addressed throughout its implementation. The comparison of the developed recognizer with previously published work is presented in Section 4 and, in Section 5, some final remarks and future lines of work are presented.

2. Rule-based Component

The rule-based component handles expressions which can generally be considered as number-based. The following types of expressions constitute the most general types considered: (i) **Numbers** are expressions denoting numbers and are marked as `NUMEX`; (ii) **Measures** are terms expressing measure values and are marked as `MEASEX`; (iii) **Time** are terms expressing time and are marked as `TIMEX`; (iv) **Addresses** are expressions conveying addresses and are marked as `ADDREX`.

It may seem that addresses are not compatible with the number-based definition given above, but a justification for this option will be provided below. The following sections contain a more detailed reference to the types of expressions considered and some implementation details concerning their annotation process.

2.1. Types of Entities

Each of the types presented in the previous section can be further detailed by considering a list of subtypes, or subparts in the specific case of addresses, allowing for a more refined classification of these expressions. These lists, together with a brief explanation for each subtype or subpart, are presented throughout the rest of this section.

The **Numbers** type covers the most frequent expressions related to numbers. Given the large variety of expressions covered, seven subtypes were necessary to correctly distinguish between them:

- **Arabic**: entities expressed by a sequence of digits, with the option of using a period to separate a string of three digits, counting from the right (e.g. *25765912*).
- **Decimal**: entities expressed by an arabic number followed by a decimal part, with a comma separating both parts (e.g. *-4.926,494*).
- **Non-compliant**: entities expressed by digits, the period and comma symbols, organized in any possible way other than as in arabic or decimal expressions (e.g. *-4,553,228.123*).
- **Roman**: entities expressed by the roman letters [IVXLCDM], in either uppercase or lowercase, obeying the well-formedness rules for roman numerals (e.g. *MCMXCIX*).
- **Cardinal**: entities that are expressed by a full or partial word description of an arabic or decimal number (e.g. *cinquenta e oito*).
- **Fraction**: entities expressed by arabic, decimal or cardinal numbers, and specific symbols or expressions representing division (e.g *89%*).

- **Magnitude class**: entities expressed by arabic, decimal or cardinal numbers together with expressions representing numerical magnitude (e.g. *5 dezenas de milhar*).

The **Measures** type covers several expressions related to measures. A less disperse range of expressions, in comparison to the previous type, meant that three subtypes were sufficient for the separation of the relevant expressions:

- **Currency**: expressions composed of an arabic, decimal or cardinal number followed by a word or expression representing a currency (e.g. *6 libras*).
- **Time**: expressions composed of an arabic, decimal or cardinal number followed by a word or expression representing a time measure (e.g. *vinte segundos*).
- **Scientific units**: expressions composed of an arabic, decimal or cardinal number followed by a word or expression representing a scientific unit (e.g. *56 toneladas*).

The most common expressions identifying points or stretches of the time line are within the scope of the **Time** type. As with the previous type, three subtypes proved to be adequate for the distinction between the expressions in question:

- **Date**: expressions representing a date, whose components can be a day of the week (e.g. *Segunda-Feira*), a day of the month (e.g. *27*), a month (e.g. *Novembro*) or a year (e.g. *2006*).
- **Time periods**: expressions made by arabic, roman or cardinal numbers and an explicit indication of a period of time concerning a specific year, decade or century (e.g. *década de oitenta*).
- **Time of the day**: expressions with different formats, indicating a specific time of the day (e.g. *às 13h30*).

Although the **Addresses** type is not fully number-based by its nature, it was included in this component of the recognizer due to a sufficiently limited amount of possible formats for writing addresses which are considered correct. Unlike the previous classes of expressions, no subtypes were defined, with the following list providing instead a description of what we consider to be the 3 main subparts of an address:

- **Global section**: expressions referring to the global position of a location (e.g. *Rua Almeida Garrett*). This subpart is mandatory for an address to be recognized.
- **Local section**: expressions referring to a specific position within the global position (e.g. *N^o 17 - 7^o Dto*).
- **Zip code**: expressions referring to what is known as the zip code of an address (e.g. *3654-548 Lisboa*).

2.2. Regular Expressions

This component was implemented resorting to regular expressions. The chosen tool for this purpose was JFlex [Klein 2004], a lexical analyzer generator for Java^{TM}. The main reasons behind this choice were the easy to learn and versatile syntax provided and the possibility of writing algorithms in Java^{TM}.

The input text must contain part-of-speech, inflection feature and lemma tags, according to the format defined in [Barreto et al. 2006]. These tags provide the necessary lexical information for the recognizer to correctly use the defined set of rules.

A key source of information on which the recognizer relies is a lexicon whose entries are single and multiword expressions that indicate the presence of a named entity. The rules were designed to be capable of handling additions made to lexicon at any time, which means that the named entity coverage can be expanded without the need for direct changes in the code. Another purpose that the lexicon serves is to hold the standard representation given to each of its entries, allowing for the generation of a final representation for a named entity.

An underlying mechanism worth noting is the inflection feature retrieval, which is responsible for determining these features from the lexical information associated with the words that compose a given named entity. Consider the following example of a cardinal numeral that illustrates the output produced by the recognizer:

```
<ENTITY><TEXT>mil e quinhentos</TEXT><OPTION ID=’1’><NUMEX>
   <TYPE>Cardinal</TYPE>
   <VALUE INFLECTION=’mp’>+1,500</VALUE>
</NUMEX></OPTION></ENTITY>
```

The text within the `TEXT` tags is replaced, in the original input, with the presented output. The `VALUE` tag contains an attribute called `INFLECTION` that stores the inflection features of the named entity. For this particular case, the cardinal numeral is masculine (gender) and plural (number).

3. Hybrid Component

The hybrid component handles expressions conveying names. They are marked as `NAMEX` and, since their range is extremely diversified, a list of subtypes is considered, which is a superset of [Chinchor 1997], allowing for a more refined classification: (i) **Persons** are expressions conveying names of people, with the option of considering the job or social status of a person if present (e.g. *Presidente Cavaco Silva*); (ii) **Organizations** are expressions conveying names of companies (e.g. *LG Electronics*) and political organizations (e.g. *ONU*); (iii) **Locations** are expressions referring to specific geographical locations (e.g. *Portugal*); (iv) **Works** are expressions referring to movies, books, paintings and similar works (e.g. *O Retrato de Dorian Gray*); (v) **Events** are expressions referring to competitions, conferences, workshops and similar events (e.g. *Fantasporto*); (vi) **Miscellaneous** are expressions referring to entities that cannot be classified according to any of the previous subtypes (e.g. *Boeing 747*).

Over the next sections, the process through which the annotation of the types of expressions is achieved will be explained, starting with the chosen data set and proceeding with the annotation itself.

3.1. Data Set

We used a data set which is a section of the corpus described in [Barreto et al. 2006]. It consists of newspaper text with approximately 260,000 tokens and is fully annotated with sentence/paragraph, part-of-speech, inflection feature, lemma and IOB tags. The IOB tagging scheme is an annotation layer that delimits and classifies named entities, meaning that every token in the corpus is marked with one of three tags in accordance with the MUC guidelines [Chinchor 1997]: `‘O’` (outside), `‘B’` (begin) or `‘I’` (inside).

Additionally, a suffix indicating the type of the named entity is appended to both `'B'` and `'I'` tags. The tags account for the defined entity types: person (`PER`), organization (`ORG`), location (`LOC`), work (`WRK`), event (`EVT`) and others (`MSC`). The frequency concerning named entities in the corpus are shown in Table 1. The total number of entities and the number of distinct entities in the corpus are presented for all types simultaneously and for each of the six individual types contemplated.

Table 1. Stats regarding named entities in the data set

	All types	Individual types					
		PER	ORG	LOC	WRK	EVT	MSC
Tokens	11,955	5,489	3,147	2,341	412	171	395
Types	4,897	2,171	1,230	974	268	111	143

3.2. Statistical Taggers

Since the statistical component of the recognizer only requires the presence of POS and IOB tags in the corpus, the remaining linguistic information is discarded at this stage. Given the expansion made to the IOB tagging scheme, the tagset is composed of 13 tags (the `'O'` tag and six different `'B'` and `'I'` tags).

Instead of working with just this tagset, other two variants of it were also considered from the start. This allowed us to determine how a statistical tagger would perform when used with the selected corpus annotated with each of these alternatives. The three defined tagsets are the following: (i) Fully Stripped (FS), where only the standard tags are used (`'B'`, `'I'`, `'O'`); (ii) Partially Stripped (PS), where from the standard tags, only the `'B'` tags carry a suffix that indicates the type of an entity (`'B-[type]'`, `'I'`, `'O'`); (iii) Fully Annotated (FA), where from the standard tags, both the `'B'` and `'I'` tags carry a suffix that indicates the type of an entity (`'B-[type]'`, `'I-[type]'`, `'O'`).

We selected two specific statistical taggers, TnT [Brants 2000] and MXPOST [Ratnaparkhi 1996], that would allow us to approach the problem at hand resorting to different techniques, since they are based on the Hidden Markov Models and Maximum Entropy algorithms, respectively, and have shown the best scores for POS tagging of Portuguese [Branco and Silva 2004].

The obtained results for each of the possible combinations between taggers and tagsets are presented in Table 2,[2] keeping in mind that the training and test sets used were the result of separating the corpus into 80% for training, with each of the tokens being composed of a word and its IOB tag (the POS tag is not considered at this stage), and the remaining 20% for testing.

The significative difference between the results for the FS tagset and the PS and FA tagsets can be explained by the absence of types in the former tagset. A statistical model created from a corpus annotated with the FS tagset will only be capable of detecting and circumscribing named entities, but not classifying them.

[2] Precision, Recall and F-measure, are obtained using the usual formula. Precision (P) measures the proportion of correctly annotated entities on the total number of entities annotated by the system. Recall (R) measures the proportion of correctly annotated entities on the total number of entities in the original data set. F_1 is $2PR/(P+R)$ here.

Table 2. Results for the statistical taggers with the defined tagsets

Tagger	Tagset	Precision	Recall	F_1
TnT	FS	82.34%	82.99%	0.8267
	PS	75.18%	75.55%	0.7536
	FA	75.21%	77.71%	0.7644
MXPOST	FS	86.76%	89.65%	0.8818
	PS	76.84%	77.92%	0.7737
	FA	75.63%	77.80%	0.7670

The other observation worth noting is the apparent similarity between the results obtained for the PS and FA tagsets. However, the two taggers have different tendencies for each of these tagsets, with the absence of the established entity types in the `I' tags proving to be a bottleneck in performance for TnT, whereas their presence helps to improve the performance levels of MXPOST. The lower performance values scored for Precision can be explained by the incorrect annotation of tokens with the `B' and `I' tags, leading to an increase in the total number of named entities identified by the taggers.

3.3. Error Analysis and Correction

After an analysis of the automatically generated reports containing the annotation errors produced by the taggers, we found several patterns in the annotation errors made by the statistical taggers. Given that this could prove to be a way of improving the performance scores obtained, the next step taken was the implementation of a rule-based application that would take the results provided by the taggers as its input, with the purpose of recovering incorrectly annotated named entities matching the defined error patterns.

Although some similarities were encountered while analyzing the reports, a clear division between the errors associated with the FS tagset and the PS and FA tagsets became apparent, which prompted us to develop two rule-based modules, one for each of the previous groups of tagsets. Our first experiments, involving these modules, provided very little performance gains, since the exclusive use of the IOB annotation layer could only provide so much information and would only allow us to make the following two very simple and straightforward corrections:

- Search for named entities whose first token was annotated with an `I' tag and change it to a `B' tag: `Pedro/I Martins/I` ⇒ `Pedro/B Martins/I`
- Search for two or more consecutive named entities without any tokens with an `O' tag separating them, leaving only the first `B' tag and changing all other occurrences to an `I' tag: `Pedro/B Martins/B` ⇒ `Pedro/B Martins/I`

With the available information for each token (the word itself and the IOB tag), any other kind of correction would involve the definition of rules based on directly coded single or multiword expressions. The consequence of this kind of approach would be the need to create a list of possible patterns, that could serve as hints for the presence of named entities, and the implementation of rules for each of them.

In order to introduce generality and be able to eliminate the need to check a word itself when analyzing a token, the POS annotation layer was brought into play, since it provides valuable information on the type of the word and allows for the implementation of more abstract rules with a wider range of error matching. This meant that the rules had to be restructured in order to contemplate both the IOB and POS annotation layers.

Moreover, each of these modules uses a lexicon whose entries are single and multiword expressions that serve as indicators for the presence of named entities belonging to the Person, Organization, Location and Event types. The remaining two types were not included in the lexicon due to their unpredictable nature, which means that, for example, we would have had to list movies, books, paintings and so forth, in order to allow the lexicon to become useful when it came to classify named entities of these two types.

The main underlying purpose of the lexicon is to aid the modules in circumscribing and identifying named entities. Since one of the modules is responsible for dealing with data sets annotated with either the PS or FA tagsets, the lexicon serves the alternative purpose of trying to improve the classification of previously annotated named entities.

The results presented in Table 3 are an update of the results obtained at the end of the previous section, with the former reflecting the execution of the rule-based correction modules over the latter. An overall improvement was achieved, with the main outstanding point being the performance values for MXPOST when dealing with the FS tagset, with increases of over 4 percentage points for the Precision, Recall and F_1 measures.

Table 3. Results for the rule-based correction modules with the defined tagsets

Tagger	Tagset	Precision	Recall	F_1
TnT	FS	87.11%	86.53%	0.8682
	PS	75.85%	78.89%	0.7734
	FA	76.68%	80.54%	0.7856
MXPOST	FS	91.37%	93.81%	0.9257
	PS	80.07%	82.54%	0.8129
	FA	79.08%	82.04%	0.8053

The fact that mistakes are made by the error correction modules in some specific cases led us to consider alternative approaches. One of them was the use of a memory based learning method. But, although we have not explored this approach thoroughly, we think this would perform as poorly as the taggers did on their own.

3.4. Composed Method

The results presented in the previous two sections led us to separate the tasks involved in the named entity recognition process. Instead of simultaneously circumscribing and classifying entities, we chose to isolate these two steps and perform them in their logical order. The main motivation behind this course of action was the disappointing results when using the PS and FA tagsets with both taggers (simultaneous circumscription and classification) and the positive results for the FS tagset (equivalent to the circumscription task only).

The first step of this new method has already been shown and consists of annotating the data set with the FS tagset using MXPOST and the rule-based module for error correction. In order to build upon the results provided by this previous stage, we had to make some changes in the data formatting process that had been adopted so far.

The new statistical model was created with TnT only, from the same training set previously mentioned, but with the difference that each token now consists of the word and the tag given to it at the end of the previous step. The results are presented in Table 4. The values in the first row were obtained by directly using TnT over the results of the previous circumscription step and without the intervention of the rule-based module, whereas the second row presents the final results obtained after this module.

Table 4. Tests with POS tags and rule-based modules

Error correction	Precision	Recall	F_1
Before	77.10%	76.88%	0.7699
After	86.53%	84.94%	0.8573

With the exception of token accuracy, there is not a significative difference between the initial results presented in Table 2 for the FA tagset and those in the first row. However, the final results in the second row are superior to those in Table 3 for both taggers, although there is still a deficit in token accuracy.

After reviewing the automatically generated reports containing the annotation errors, we established that one of the main sources of errors is a frequent lack of ability to distinguish between the Organization and Location types for specific named entities.

The most common case occurs when a reference to a country is made. As an example, consider the entity `Portugal/B-ORG`, which represents a specific geographical location and a political organization. This kind of ambiguity as to the semantic meaning of an entity cannot easily be resolved by the tagger nor by the rule-based module. This means that, for the majority of these cases, the entity will be incorrectly annotated as a location: `Portugal/B-LOC`.

4. Evaluation and Related Work

One initiative that produced some evaluation results is HAREM [HAREM 2006]. This initiative consisted of a joint evaluation of some Named Entity Recognition systems that focused specifically in Portuguese texts. A full comparison of our results with the ones produced by the participants in this initiative is not possible (the premisses do not coincide, namely in what respects semantic classification).

4.1. Evaluation of the Rule-Based Component

Concerning the evaluation of the rule-based component, we chose to use an annotated section of a larger corpus that is known as the Golden Collection texts, provided by the HAREM initiative. Only a subset of the annotated named entities present in the data set was considered, due to the remaining types not being dealt with by this component of the recognizer. The results obtained are presented in Table 5.

Some issues regarding these results should be highlighted. The quantity entities, telephone numbers and screen resolution values were considered quantities in that corpus, which cannot be considered a very conventional decision, from our point of view. In addition, some entities were composed of a numeral and an invalid unit, which prevented its annotation by the recognizer.

Table 5. Results for the rule-based component with the HAREM corpus

Entity type		Precision	Recall	F_1
Value	Quantity	93.28%	96.72%	0.9497
	Currency	83.93%	82.46%	0.8319
Time	Hour	100.00%	91.30%	0.9546
	Date	76.75%	76.47%	0.7661
Global score		85.19%	85.91%	0.8555

Concerning the date entities, a relatively large subset of the errors produced by the LXNER are references to years, composed of arabic numbers placed within parentheses. These are extremely complicated cases to handle with a rule-based approach, since there is a total absence of contextual hints in order to allow the recognizer to decide correctly when annotating them. If these cases were to be ignored while running the recognizer, it would be possible to achieve higher performance values for this type of named entity. Specifically, we would obtain 92.04% Precision (a 19.92% increase), 91.63% Recall (a 19.82% increase) and 0.9183 F_1 (a 19.87% increase).

4.2. The Hybrid Component

The HAREM initiative involved three different tasks: identification, semantic classification and morphological classification [Santos et al. 2006]. Only the first two are relevant here. As to the identification task, the best results obtained for our recognizer and for the HAREM initiative are presented in Table 6.

Table 6. Comparison between HAREM results and our system

System	Precision	Recall	F_1
Best HAREM (Cortex2CEM)	87.33%	-	-
Best HAREM (Cortex1REM)	-	87.00%	0.8323
Our system (using TnT)	87.11%	86.53%	0.8682
Our system (using MXPOST)	91.37%	93.81%	0.9257

Although our results with TnT for Precision and Recall are slightly worse when compared with the two different systems that produced the best HAREM results, concerning the F-measure, our system performs better. But clearly the best results are obtained when using the MXPOST tagger, which produces results well above 90% for all measures.

In what concerns semantic classification, it is difficult to establish a fair comparison between systems due to the specificities of the HAREM evaluation setup. Nevertheless, the best LXNER results for the F_1 (0.8573), presented in Table 4, seem to be generally better when compared with the ones obtained by the participants in HAREM (but again, we would like to stress that further evaluation is still needed).

5. Concluding Remarks

This paper presented a tool for the recognition of named entities in Portuguese. Its rule-based component is responsible for dealing with numbers, measures, time and addresses, whereas its hybrid component deals with names. Every entity is annotated with a XML-like markup that contains a standard representation and information regarding the entity.

Due to the absence of a data set representative of all named entity types considered, the number-based, rule-based component was only partially tested with the help of the Golden Collection corpus, globally scoring 85.19% Precision, 85.91% Recall and 0.8555 F_1. As for the name-based, hybrid component, it globally scored 86.53% Precision, 84.94% Recall and 0.8573 F_1. Overall, in as much as they lend themselves to be compared, these scores seem to be generally better than the ones reported in the literature for NER in Portuguese.

Future work on the hybrid component will focus on the expansion of the rule-based modules in order for them to be able to cover more error cases. As for the rule-based component, it has reached a very stable development level and a possible improvement that can be made at this stage seem to be the addition of new subtypes or the expansion of some of the current subtypes. Two other alternative paths will be followed for the hybrid component, with the first of them being the expansion of the data set, which could theoretically allow for the generation of more robust and capable models. The second alternative will focus on the usage of more information about a token, which could improve the performance levels registered at the present time.

References

Barreto, F., Branco, A., Ferreira, E., Mendes, A., Nascimento, M., Nunes, F., and Silva, J. (2006). Open resources and tools for the shallow processing of portuguese. In [Calzolari et al. 2006].

Branco, A. and Silva, J. (2004). Evaluating solutions for the rapid development of state-of-the-art POS taggers for portuguese. In Lino, M. T., Xavier, M. F., Ferreira, F., Costa, R., and Silva, R., editors, *Proc. LREC2004*, pages 507–510, Paris. ELRA.

Brants, T. (2000). TnT - a statistical part-of-speech tagger. In *Proceedings of the 3rd Applied Natural Language Processing Conference*, pages 224–231.

Calzolari, N., Choukri, K., Gangemi, A., Maegaard, B., Mariani, J., Odjik, J., and Tapias, D., editors (2006). *Proceedings of LREC2006*. ELRA.

Chinchor, N. (1997). MUC-7 named entity task definition (version 3.5). Available at: `http://www-nlpir.nist.gov/related_projects/muc/proceedings/ne_task.html`.

HAREM (2006). HAREM - avaliação conjunta de sistemas de reconhecimento de entidades mencionadas. Available at: `http://poloxldb.linguateca.pt/harem`.

Klein, G. (2004). JFlex user's manual (version 1.4.1). Available at: `http://www.jflex.de/manual.html`.

Ratnaparkhi, A. (1996). A maximum entropy part-of-speech tagger. In *Proceedings of the Empirical Methods on Natural Language Processing Conference, ACL*, pages 133–142.

Santos, D., Seco, N., Cardoso, N., and Vilela, R. (2006). HAREM: An advanced NER evaluation contest for portuguese. In [Calzolari et al. 2006], pages 1986–1991.