Yet another suite of multilingual NLP tools

Marcos Garcia and Pablo Gamallo

Centro Singular de Investigación en Tecnoloxías da Información (CiTIUS)
Universidade de Santiago de Compostela — Galiza (Spain)
marcos.garcia.gonzalez, pablo.gamallo @usc.es

Abstract. This paper presents the current development of a multilingual suite for Natural Language Processing. It consists of a sentence chunker, a tokenizer, a PoS-tagger, a dictionary-based lemmatizer and a Named Entity Recognizer (both for *enamex* and *numex* expressions). The architecture of the pipeline and the main resources used for its development are described. Besides, the PoS-tagger and Named Entity Recognizer are evaluated against several state-of-the-art systems. The experiments performed in Portuguese and English show that, in spite of its simplicity, our system competes with some well known tools for NLP. It is entirely written in Perl and distributed under a GPL license.

Keywords: natural language processing, PoS-tagging, named entity recognition, portuguese, english

1 Introduction

This paper presents CitiusTools, a multilingual suite for Natural Language Processing (NLP) which performs the following tasks: sentence chunking, tokenization, PoS-tagging, lemmatization and Named Entity Recognition (NER). The suite is entirely written in Perl and distributed under a GPL license.[1]

The paper presents the architecture of the pipeline as well as its adaptation to Portuguese and English (the Spanish version was introduced in [6]). It is also presented a set of experiments aimed at knowing the performance of the PoS-tagger and the NE classifier modules. The results show that, in spite of its simplicity, our system behaves quite well when compared to some state-of-the-art suites such as Stanford CoreNLP or FreeLing. Besides, it performs notoriously better than the models provided by other systems such as OpenNLP.

Sect. 2 introduces some related work. Then, the architecture of the system is presented in Sect. 3. Sect. 4 shows the external resources used for its adaptation to Portuguese and English, while Sect. 5 contains the performed experiments. Finally, Sect. 6 describes the main conclusions of this paper.

2 Related Work

In the last years, several open-source NLP suites have been published, being available to the users. Some of them provide models for languages such as Por-

[1] http://proxectos.citius.usc.es/hpcpln/index.php

tuguese and English (evaluated in this paper), while others include analyzers for other varieties such as Spanish, Chinese, German or Arabic.

Stanford CoreNLP [11] is one of the best known suites, including modules like tokenizers, PoS-taggers, named entity recognizers, coreference resolution systems and syntactic parsers. It is written in Java and has been developed mainly for English, but recently there have been published models for languages such as Spanish, Chinese, German or Arabic.

FreeLing [12] is a suite of language analyzers (written in C++) which includes similar modules than the Stanford system, and also has tools for other tasks such as phonetic encoding. Most of FreeLing modules analyze data in Catalan, Spanish, Portuguese, English, or French (among others).

Another toolkit for NLP analysis written in Java is OpenNLP,[2] which performs most common NLP tasks. There are available models for several language for this system, including English, Spanish or German.

Finally, IXA pipes [1] (a modular set also written in Java) performs tokenization, PoS-tagging, NER and parsing. Among the languages covered by this tool (depending on the module) are Spanish, English, Basque, Italian or Galician.

The system presented in this paper is, to the best of our knowledge, the first one written entirely in Perl. It provides a simple, efficient and ready to use set of NLP tools with a performance close to the state-of-the-art.

3 Architecture

Our system consists of five modules that can be applied in a pipeline in order to perform NLP tasks. The current version contains the following tools:

3.1 Sentence chunker

This module is composed of a language-dependent list of abbreviations and a set of Finite State Automata (FST) aimed at identifying sentence boundaries.

The automata detect entities such as urls, e-mail addresses, and other elements containing dots that are not in sentence-ending position. Also, abbreviations ending in a dot character (e.g., *Dr.*, *corp.*, etc) are not marked as sentence boundaries (except if their context is covered by a FST).

The output of this module is the input text with one sentence per line.

3.2 Tokenizer

The next module of the suite splits each identified sentence into its tokens. It is a rule-based tokenizer enriched with few language-dependent adaptations.

First, the tokenizer identifies compound punctuation (such as ellipsis) and other punctuation inside numerical expressions. After that, a simple blank-space tokenizer is applied (which also splits the punctuation which do not belong to larger expressions).

[2] `http://opennlp.apache.org/`

Then, a battery of language-dependent rules is applied in order to split contractions (e.g. *don't* > *do/not*, in English), verb+pronoun forms (e.g., *mantém-se* > *mantém/se*, in Portuguese) and other elements which are useful for further NLP analysis. Note that some forms can be ambiguous between a contracted and a non-contracted element: *desse*, in Portuguese, could be a single token form of the verb *dar* (to give), or a contracted form of a preposition and a demonstrative (*de/esse*). As the decision for splitting these forms depends on their PoS-tag, the tokenizer does not split them. Thus, as in other works [8], these forms are analyzed by the PoS-tagger, which will split them (or not), according to the selected PoS-tag. Those cases where an element of the contraction may represent two different tokens (with a different PoS-tag, e.g., *I'd* > *I would* or *I had*, in English) are also splitted in this step, but the lemma will be provided by the disambiguation of the PoS-tagger.

The output of this module is a vector of tokens representing each previously identified sentence.

3.3 PoS-tagger

The PoS-tagger assigns a morphosyntactic tag (from a set of predefined tags, the *tagset*) to each token.

This module is a bayesian classifier based on bigrams of tokens. It uses additive smoothing, which is commonly a component of bayesian classifiers. In order to label a token, the classifier calculates the probability of each tag (t_i) linked to the token, taking into account a set of contextual features $A_{\mathrm{i}} \ldots A_{\mathrm{n}}$:

$$P(t_i \mid A_1, ...A_n) = P(t_i) \prod_{i=0}^{N} P(A_i \mid t_i) \tag{1}$$

The best set of features, selected in preliminary tests, was the following:

- $t_{\mathrm{i\text{-}1}}$: the PoS-tag of the previous token.
- $t_{\mathrm{i+1}}$: the PoS-tag of the next token.
- $(k_{\mathrm{i}}, t_{\mathrm{i\text{-}1}})$: the cooccurrence of the ambiguous token k_i together with the tag of the previous token.
- $(k_{\mathrm{i}}, t_{\mathrm{i+1}})$: the coccurrecence of the ambiguous token k_i together with the tag of the next token.

The model needs to be trained with a labeled corpus and a dictionary with the possible PoS-tags for each known token. The algorithm disambiguates the tokens from left to right, so the left context of an ambiguous token is an already labeled one. Thus, the features concerning the tag of the next token ($t_{\mathrm{i+1}}$) include the probabilities of the different tags that could be associated with this token.

This strategy is similar to the Hidden Markom Models (HMM) algorithm proposed in [2]. The main difference is that our system handles the PoS-tagging as an individual classification problem (token by token), instead of searching for the best sequence of PoS-tags. Its computational efficiency is the main reason for the use of this simple approach.

The tagsets of the PoS-taggers follow the EAGLES guidelines [10]. For Portuguese, it has been used a tagset with 193 elements. The English tagset has 27 tags. Both of them have 9 extra tags for punctuation. The difference between these tagsets come from the complex verbal conjugation and nominal inflection of Portuguese. However, note that the classifier does not use the 193 elements in Portuguese: it just uses 21 tags for disambiguating the morphosyntactic category (e.g., noun, adjective) of each word. The other information (gender, number, tense, etc.) is then taken from the labeled dictionary.

The output of the PoS-tagger is the input vector enriched with a morphosyntactic label for each token.

3.4 Named Entity Identifier

The next module of the pipeline is a FST identifier of *numex* and *enamex* (named entities) expressions.

Before starting the identification process, this module takes advantage of a lemmatized dictionary (see Section 4) in order to assign a lemma for each token. It also uses the predicted PoS-tag for disambiguating tokens with different lemmas depending on their morphosyntactic category.

For identifying *numex* expressions (in our system: dates, currencies, numbers, measures and quantities), it is applied a set of language-dependent FSTs that cover the most common forms of representing these elements in each language.

The named entities (*enamex* expressions) are identified taking into account both their capitalization and possible functional words inside them (e.g., *Banco* ***de*** *Portugal*). In order to better identify the boundaries of the *enamex* expressions, this module also needs a list of words which can be both common words at sentence beginning position and the first element of a named entity (e.g., *Neves*, which can be a capitalized noun and a proper noun —surname or location— in Portuguese). These ambiguous forms are obtained semi-automatically using dictionaries and lists of gazetteers.

The output of this module is the input vector enriched with the identification of the *numex* and *enamex* expressions, as well as with the lemmas provided by the dictionary.

3.5 Named Entity Classifier

The named entity classifier module assigns each *enamex* one of the following labels: *person*, *organization*, *location* or *misc* (miscellaneous).

In order to classify an entity, this module uses large lists of encyclopedic gazetteers together with a set of rules for semantic disambiguation.

The gazetteers were automatically extracted from structured resources such as Freebase[3] and DBpedia,[4] and enriched with semi-structured knowledge ob-

[3] `http://www.freebase.com`
[4] `http://www.dbpedia.org`

tained from the infoboxes and category trees of Wikipedia.[5] The gazetteers consist in four lists of entities (one for each semantic category). Besides, the system also uses small lists of *trigger words*, which are nouns that can subclassify an entity (e.g., "singer" for the class *person* or "company" for *organization*). The trigger words were also automatically extracted from the category trees of Wikipedia. Finally, a list of the most frequent personal names for each language (which are not common nouns) is used.

Concerning the disambiguation rules, they are applied using the following strategy for each named entity:

1. If the entity appears only in one of the gazetteers lists, it is classified with the class it belongs to.
2. If the entity appears in several lists (or if it does not appear in any), the context is analyzed. This context includes two windows (*before* and *after*) of three tokens each. If a trigger word is found in the context, the entity is classified as belonging to the trigger word class (with some restrictions such as trigger words in preposition phrases. "Caixa Geral" will not be labeled as *person* even if the trigger word "president" occurs in the context: ***president*** *of Caixa Geral*).
3. If the entity starts (or is) a frequent personal name present in the list, it is classified as *person*.
4. If the entity is ambiguous (it appears in more than one list or contains trigger words from different classes) and it cannot be disambiguated by its context, it is selected the most probable class (*prior probability*), by computing the distribution of the gazetteers in the Wikipedia.
5. If the context is not enough to disambiguate the entity, a rule verifies whether it contains a trigger word or the first token of a gazetteer inside. If there are more than one option, the gazetteers are preferred over the trigger words, and in case of ambiguity the prior probability is also computed.
6. If the previous rules cannot classify the entity, it is labeled as *misc*.

Note that the rules are mainly language-independent. In our case, only one rule had to be changed when adapting the system for English: a trigger word inside an entity appears in final position, instead of in the beginning, as in Portuguese (*National* ***Museum*** versus ***Museu*** *Nacional*).

Even though the performance of this module depends on the quality and persistence of the gazetteers, the use of contextual features together with the combination of rules that analyze the internal form of each entity allow the system to keep reasonable accuracy even in unknown forms.

4 Resources

This section briefly describes the external resources used by the different NLP modules of our system. Tab. 1 includes a summary of these data.

[5] `http://www.wikipedia.org`

Table 1. Summary of the size of the resources: dictionaries, PoS-tagger training corpora, NER testing corpora and total number of gazetters.

Language	*Dictionary*	PoS-tagger (train)	NER (test)	Gazetteers
Portuguese	1.250M	130k	75k	100k
English	350k	1M	524k	1.5M

4.1 Portuguese

For training the PoS-tagger for Portuguese (and also for extracting some lists described above), we used the dictionary of FreeLing based on the Label-Lex lexicon [4]. It consists of ≈ 1.250 million pairs token-tag from about $120k$ lemmas.

For training the PoS-tagging we used a subset of the CoNLL version of the Bosque 8.0, with about $130k$ tokens.[6] For testing, we used a different subset of the Bosque and three small corpora of European Portuguese (EP) news, Brazilian Portuguese (BP) news and a Wikipedia articles.

For testing the named entity classification, there were used both a subset of the labeled version of the Bosque ($\approx 20k$ tokens) and the Corpus-Web (with about $55k$ tokens of different varieties of Portuguese) [9].

In order to build the gazetteers, the Portuguese version of the Wikipedia was used for extracting entity names. Apart from that, large lists of countries and cities were also merged, together with the most common names and surnames in Portuguese an other lists of gazetteers freely available (such as the FreeLing data), generating the following lists: $59,421$ *person* entities, $14,197$ *organizations*, $34,590$ *locations* and 838 for *misc* gazetteers.

4.2 English

For English, the morph_english dict.v1.4 was used, with about $350k$ token-tag pairs from $\approx 77.5k$ lemmas.[7] For training and testing the PoS-tagger we used the Brown corpus, with ≈ 1.2 million tokens:[8] ≈ 1 million tokens were randomly selected for training, while the tests were carried out with the other $200k$ tokens. Both the dictionary and the corpora had to be adapted and converted to the same tagset.

The classification of named entities was evaluated using two corpora: the IEER,[9] with $68,402$ tokens and classification of *person*, *location* and *organization* entities (not *misc*), and the SemCor Corpus,[10] with a size of $455,597$ tokens and annotation of the four *enamex* classes. The PoS-tags of this last corpus had been predicted (not manually revised).

[6] `http://www.linguateca.pt/floresta/CoNLL-X/`
[7] `ftp://ftp.cis.upenn.edu/pub/xtag/morph-1.5/morph-1.5.tar.gz`.
[8] `http://clu.uni.no/icame/brown/bcm.html`
[9] `http://www.itl.nist.gov/iad/894.01/tests/ie-er/er_99/er_99.htm`
[10] `http://www.gabormelli.com/RKB/SemCor_Corpus`

The English gazetteers were extracted from Freebase and DBpedia, enriched with lists of countries and capitals and the most common names and surnames in this language. The final versions had the following size: 922, 767 for *person*, 126, 334 for *organization*, 351, 151 for *location* and 94, 525 for *misc*.

5 Evaluation

This section describes the evaluation experiments performed on the two main modules of the system: the PoS-tagger (CitiusTagger) and the NE classifier (CitiusNEC). The experiments were carried out in Portuguese and English, using three NLP suites for comparison: FreeLing (for Portuguese), and Apache OpenNLP and Stanford CoreNLP (for English).[11]

It is important to note that some results are not strictly comparable, since we used the models provided by each software. On the one hand, these models were trained with different resources (corpora, lexicons, gazetteers...), having also different tagsets (quickly adapted for doing the experiments). On the other hand, the alignment between the gold-standard and the test files also involved variation on the results (as it is shown below).

So, the objective of this evaluation is not to know what is the best system for PoS-tagging and NE classifying texts in Portuguese and English, but to have a decent comparison of our system analyzing the same data as other NLP suites.

5.1 PoS-tagger

The first set of experiments compared the performance of the PoS-tagger in Portuguese and English.

Table 2. PoS-tagging results (precision) for Portuguese.

Corpus	*Size*	CitiusTagger	FreeLing
Bosque	80,881	96.07	96.62
EP News	13,964	96.70	97.76
BP News	11,476	95.73	96.99
Wikipedia	17,149	95.76	96.13
Macro-average	—	96.06	96.88
Micro-average	—	96.06	96.72

Tab. 2 contains the results for Portuguese. Our bayesian PoS-tagger were compared to the HMM model of FreeLing [12, 8], analyzing the four mentioned corpora (see Section 4). The results include the precision (true positives / true

[11] The output of each system as well as the gold-standard files can be obtained in the following url: `http://gramatica.usc.es/~marcos/slate15.zip`.

positives + false negatives) on each corpora as well as the macro and micro-average values (macro-average is the harmonic mean of the results from each corpus while micro-average values are computed from the sum of all the true and false positives and negatives from each corpora).

When compared to the HMM model, our system behaves quite similar in every corpora (with a maximum difference of -1.2 in BP News), with average results of 96% precision. Note that this comparison is strict, since both the gold-standard and the testing corpora were perfectly aligned. Besides, the tagset of our system and the FreeLing one were almost identical.

In English, the bayesian PoS-tagger was compared to three different models (in one corpus): the maximum entropy and perceptron classifiers of OpenNLP (1 and 2, respectively) and the Stanford POS Tagger (maximum entropy) [13].

The output of the external systems (OpenNLP and Stanford) were automatically converted to the same tagset of the gold-standard.

Table 3. PoS-tagging results (precision) for English. OpenNLP_1 is a maximum entropy model, while OpenNLP_2 is a perceptron classifier. Test corpus has a size of 209,406 tokens.

CitiusTagger	OpenNLP_1	OpenNLP_2	Stanford
93.55	91.72	90.93	91.12

The results (Tab. 3) show that our PoS-tagger behaves as good as the maximum entropy and perceptron models. Actually, the precision of the bayesian model is almost 2% higher, but the evaluation cannot be strict: some minority tags (e.g. FW for *foreign words*) appeared in the gold-standard but not in the tagsets of these taggers (and vice versa).

However, these experiments (together with the Portuguese ones) suggest that the bayesian model achieves a high performance despite its simplicity.

5.2 Named Entity Classifier

Concerning NE classification, the Portuguese system was also compared to the FreeLing AdaBoost classifier [3, 7] in two corpora: Bosque and Corpus-Web.

Tab. 4 shows the results of these two classifiers in the referred corpora. In Bosque, our system achieved slightly better results than the AdaBoost classifier, while in Corpus-Web, the FreeLing module had better results.

Again, the average results show that a simple system (based on resources and rules) has similar performance than a supervised classifier.

In English, the resource-based method was compared to the OpenNLP (Name Finder models)[12] and to the Stanford NER (CRF with distributional similarity features in an IOB2 classifier)[13] [5].

[12] `http://opennlp.sourceforge.net/models/english/namefind/`

[13] `http://nlp.stanford.edu/software/conll.distsim.iob2.crf.ser.gz`

Table 4. Named entity classification results (f-score) for Portuguese. *NEs* refers to the number of full *enamex* entities (not tokens) in each corpus.

Corpus	*Tokens*	*NEs*	CitiusNEC	FreeLing
Bosque	19,579	1,027	90.07	88,89
Corpus-Web	55,305	3,666	73.76	75.31
Micro-average	—	—	81.92	82.10
Macro-average	—	—	77.33	78.22

Table 5. Named entity classification results (f-score) for English.

Corpus	*Tokens*	*NEs*	CitiusNEC	OpenNLP	Stanford
IEER	68,402	3,384	75.95	52.77	75.86
SemCor	455,597	9,696	58.81	44.85	65.57
Macro-average	—	—	63.38	48.90	70.72
Micro-average	—	—	63.23	47.10	68.63

The output of these systems were automatically converted to the CoNLL IOB format (used in both versions of the IEER and SemCor corpora).

The results of the named entity classifiers (Tab. 5) show that in the IEER corpus, our system behaves as good as the Stanford model, while in SemCor, the former increased our performance in more than 7%. In average, our resource-based classifier had much better performance ($\approx 5\%$) than the OpenNLP system, while the Stanford one increased our results in $5\% - 7\%$ f-score.

Finally, it was carried out a test aimed at knowing the processing speed of the evaluated systems. They were used for labelling a Spanish corpus of 100,000 tokens (in an Intel Core2 2.5GHz processor with 4gb of RAM running Debian Jessie). The systems needed the following time for applying the pipeline (sentence chunker, tokenizer, PoS-tagger and NER): OpenNLP (only NER): 1m48s; FreeLing: 2m27s; CitiusTools: 2m38s and Stanford CoreNLP: 11m25s.

In sum, the evaluations performed with the two main modules of our pipeline —CitiusTagger and CitiusNEC— suggest that they achieve very good results (some of them comparable to state-of-the-art systems) despite their simplicity and their quick adaptation to Portuguese and English. This is in accordance with the results obtained for Spanish, such as it was described in [6].

6 Conclusions and Further Work

This paper presented the current version of CitiusTools, a multilingual suite for NLP which includes modules for the most common tasks of this field.

The modules, written in Perl, combine some rule-based and supervised models which take advantage of external resources such as lexicons, labeled corpora or large lists of gazetteers.

Two different modules (PoS-tagger and NER) were evaluated in Portuguese and English, compared to some of the best NLP tools available for these languages. The results showed that the performance of our system is similar than the state-of-the-art, even if it has been quickly adapted to these languages.

In current work, we are adapting all the modules in the suite to two new languages (Galician and French), and we expect to include (in further work) a deterministic module for coreference resolution.

References

1. Agerri, R., Bermudez, J., Rigau, G.: IXA pipeline: Efficient and Ready to Use Multilingual NLP tools. In *Proceedings of the 9th Language Resources and Evaluation Conference (LREC 2014)*, Reykjavik (2014).
2. Brants, T.: TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of the 6th Conference on Applied Natural Language Processing (ANLP)*. Association for Computational Linguistics (2000).
3. Carreras, X., Màrquez, Ll., Padró, Ll.: A Simple Named Entity Extractor using AdaBoost. In *Proceedings of the Conference on Natural Language Learning (CoNLL 2003) Shared Task*. Edmonton (2003).
4. Eleutério, S., Ranchhod, E., Mota, C., Carvalho, P.: Dicionários Electrónicos do Português. Características e Aplicações. In *Actas del VIII Simposio Internacional de Comunicación Social*, pp. 636–642, Santiago de Cuba (2003).
5. Finkel, J., Grenager, T., Manning, C.: Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pp. 363-370 (2005).
6. Gamallo, P., Pichel, J.C., Garcia, M., Abuín, J.M., Fernández Pena, T.: Análisis morfosintáctico y clasificación de entidades nombradas en un entorno Big Data. *Procesamiento del Lenguaje Natural*, 53, pp. 17–24 (2014).
7. Garcia, M.: *Extracção de Relações Semânticas. Recursos, Ferramentas e Estratégias.* PhD Thesis, University of Santiago de Compostela (2014).
8. Garcia, M., Gamallo, P.: Análise Morfossintáctica para Português Europeu e Galego: Problemas, Soluções e Avaliação. In *LinguaMÁTICA*, 2(2), pp. 59–67 (2010).
9. Garcia, M., Gamallo, P.: Multilingual corpora with coreferential annotation of person entities. In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC 2014)*, pp. 3229–3233, Reykjavik (2014).
10. Leach, G., Wilson, A.: Recommendations for the Morphosyntactic Annotation of Corpora. Techincal Report, EAGLES: Expert Advisory Group on Language Engineering Standard (1996)
11. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014): System Demonstrations*, pp. 55-60 (2014).
12. Padró, Ll.: Analizadores Multilingües en FreeLing. In *LinguaMÁTICA*, 3(2), pp. 13–20 (2011).
13. Toutanova, K., Klein, D., Manning, C., Singer, Y.: Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the Human Language Technology and the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, pp. 252–259, Edmonton (2003).