Using Morphosyntactic Post-processing to Improve POS-tagging Accuracy

Marcos Garcia, Pablo Gamallo

ProLNat Group, Department of Spanish Language
University of Santiago de Compostela
Avda. de Castelao, 15782 – Santiago de Compostela
marcosgg@gmail.com,pablo.gamallo@usc.es
http://gramatica.usc.es/pln

Abstract. Many of the errors produced by up-to-date POS-taggers could be considered as morphologic, syntactic or semantic. Once statistical tagging does not deal with semantic ambiguity, the correction of (morpho)syntactic errors emerges as one of the possibilities to improve the accuracy of this task. This work describes a method that applies a robust parser with correction rules over a POS-tagging output. We outline its preliminary results on a European Portuguese corpus, showing that the most common tagging errors could be corrected up to 50% with very basic and linguistically motivated rules.

Key words: POS-tagging, rule-based correction, shallow processing.

1 Introduction

Part-of-speech tagging is one of the most studied tasks in NLP. Being a very important process in the area, there are many tools and implementation models that carry out this task, obtaining very good accuracy over a large number of languages. The analysis of current POS-tagger results shows that many of the errors arise from the sparseness of morphosyntactic or lexico-semantic data. Once statistical models reach the best POS-tagging values, even with knowledge-poor linguistic information, the application of correction rules in a post-processing step could increase the precision of tagging. Some works have implemented similar solutions in different steps of the POS-tagging process [5], improving the accuracy of the task.

We use a grammar compiler that generates parsers to apply over texts tagged with a HMM-based POS-tagger trained for European Portuguese (EP). The parser takes the output of the POS-tagger as its input, and generates the new output in different formats, including the same used by the POS-tagger. Thus, developing basic grammars with correction rules can help us to perform a post-processing of the tagged text.

One of the main issues, therefore, is to find whether exist systematic errors that can be corrected using morphosyntactic (or other) patterns, taking into account the potential generation of new tagging errors.

In the remaining of this paper, we will show the main features of the grammar compiler and its application on the tagging pipeline. Furthermore, we will show the current status of our research, concerning the analysis of the most frequent errors produced by the POS-tagger as well as the evaluation of the applied rules.

2 Correction procedure

In order to apply the generated parsers over a tagged output, we decided to train the POS-tagger module of the FreeLing suite [6, 3], providing a free POS-tagger software for EP.

Although the POS-tagger is still work in progress, the current precision values are of 94.626% and 95.689%, tested over a 50,000 token corpus[1]. The difference between the two scores is due to the evaluation process: The first one refers to the evaluation of the entire tag —containing features like case, mode, function, etc.—, while the second one is an evaluation of only the first three elements of the tag (category, type and grade/person, in EAGLES format [8]). We have to note that the initial results of the trained POS-tagger are slightly below state-of-the-art, which is reported to be around 97% [1, 2].

Besides the POS-tagger, the correction procedure includes a compiler that generates parsers from different grammars [4]. The grammars are written in a specific formalism that allows to modify the linguistic information of the tokens (type, category, gender, number, etc.), and to establish dependencies between them [7].

Instead of using this formalism to generate syntactic parsers, it can be used to create correction rules that modify and/or add linguistic information, defining the syntactic pattern of each systematic error.

For example, let us see a rule modifying an odd tag assigned to token `a` when appears to the left of a masculine noun:

Example of a Correction Rule

```
Single: DET<token:[Aa]> [NOUN<gender:M>]
Corr: tag:PRP, type:P, lemma:a
%
```

"Single" stands for a single rule (not a dependency one). It means that the element outside the brackets is the one that will be corrected, being the brackets used to define the context of the rule. Here, if a DET (determiner), whose token is `a` (or `A`), occurs before a masculine noun, then its tag will be changed by PRP (preposition, or SPS00 in the format of FreeLing) and its lemma by `a` (once the lemma of the feminine determiner `a` is `o`). Thus, this rule could correct sentences like `a nível nacional` (with a <token lemma TAG> format), previously analyzed as follows:

[1] Extracted from Bosque 8.0: `http://www.linguateca.pt/Floresta/corpus.html#bosque`

```
a o DA0FS0                                  a a SPS00
nível nível NCMS000           to            nível nível NCMS000
nacional nacional AQ0CS0                    nacional nacional AQ0CS0
```

The formalism lets to specify larger application patterns, including optional elements, disjunctions, conjunctions, and other regular expressions. It also allows to establish previous dependencies to simplify some of the rules, enlarging the coverage of the corrector:

Example of a Dependency Rule

```
AdjunctLeft: ADJ NOUN
Agreement: gender, number
%
```

This rule establishes a dependency between an adjective and a noun agreeing in gender and number, being the head of the dependency at the right. By applying this rule before that shown above, it is possible to deal with <DET ADJ NOUN> structures, because the adjective is now related to the noun.

Let us note that the execution time of the system increases when the parser is applied. However, increase in time is not very significant, once the analysis speed only decreases from 11,500 to 8,500 tokens per second.

3 POS-tagger common errors

Table 1 shows the most common errors found on the evaluation tests of the POS-tagger. The annotation of `que` produced 141 mistaggings, mainly between CS (subordinate conjunction) and PR0CN000 (relative pronoun) tags (126), plus 15 errors in other contexts; the tagging of `a` yielded 153 errors (between determiner, personal and demonstrative pronoun, preposition and common noun), while `o` produced 67. The annotation of `um` and `uma` (numeral or determiner) was also one of the most common errors.

Times	Token	Correct Tag	Assigned Tag
86	que	CS	PR0CN000
81	a	SPS00	DA0FS0
41	um	Z	DI0MS0
40	a	DA0FS0	SPS00
38	que	PR0CN000	CS
37	uma	Z	DI0FS0
24	o	PD0MS000	DA0MS0

Table 1. Most common POS-tagging errors over a 50,000 token corpus

In the first analysis of the automatic tagged texts, we found several errors that seem to be easily corrected by (morpho)syntactic rules. The disagreement of number or gender features between determiners and nouns is an example of this kind of error. Thus, the annotation of specific tokens can be improved with some simple rules. By way of illustration, the example rule shown above increases the annotation accuracy of `a` as a preposition on 33.333% (corrects 29 of 87 errors), and assigns a wrong tag to 3 (of 46) tokens (decreasing the annotation of the determiner `a` on 6.521%). Clearly, this is a very strong pattern, so the rule produces few errors (some of them due to previous mistaggings on the annotation of nouns). Other contexts, such those between `que` as conjunction or as pronoun, need a deeper processing that requires more complex grammars.

In a shallow processing, the creation and enlargement of simple rules can help us to correct many of the most common and systematic errors produced by the POS-tagger. Up to now, we have evaluated some of our rules, in particular those dealing with `a`, `o` and `que` in their more frequent error contexts.

The rules are manually written based on the semi-automatic analysis of the POS-tagger errors. Before adding them to the grammar, the rules are tested over five 10,000 token corpora, verifying its performance in each run.

In the case of `a`, the rule defined above was improved, enlarging its application context. More precisely, the determiner `a` will be tagged as preposition before plural or masculine nouns and adjectives, cardinal numbers followed by masculine adjectives and nouns, etc.

In a similar way, other rules correcting the tag of `o` and `que` were evaluated. Table 2 shows the results of the application of a set of rules for the three cited tokens.

		Errors			Errors		
Token	Correct Tag	Before	After	Correct Tag	Before	After	Improvement
a	SPS00:	87	39	DA0FS0:	46	27	50.376%
o	PD0MS000:	29	10	DA0MS0:	24	19	45.283%
que	CS:	87	59	PR0CN000:	39	33	26.984%

Table 2. Results of the applied rules

With these rules, the accuracy of the POS-tagger reaches 94.898% and 95.963% over the same 50,000 token corpus.

4 Discussion

Apart from the rules applied to correct the tagging of `a` as a preposition, (mainly with respect to the feature disagreement between the determiner and its head noun), other rules converting the preposition into a determiner were also created. Thus, doing such a change when `a` occurs before a singular feminine noun (and in other similar contexts), its annotation was improved on more than 55%, whereas the errors concerning the determiner `a` decreased about 41%.

The rules tested to correct the errors in the annotation of `o` mainly deal with structures other than <DET NOUN>, changing the tag from determiner to pronoun in some cases where the head noun is not filled (before a relative pronoun or the preposition `de`, for example). We also introduzed other rules changing the tag of `o` from relative pronoun to determiner before some interrogative contexts. With the rules applied to `o` as a determiner and as a demonstrative pronoun, the annotation accuracy of this token increased on 45.283%.

Finally, the case of `que` was more problematic, since it requires a deeper processing of the phrases in which it occurs. At present, we only have used some rules concerning the annotation of some common expressions, like `uma vez que`, `para que`, etc. Besides that, we have also tested other rules that change the tag of `que` from pronoun to conjunction in some comparative contexts (`melhor/pior do que...`) and in completive phrases following the preposition `de`. These rules increased the annotation of `que` as a conjunction on more than 32%, reducing the misstagings of the relative pronoun on 15%.

Although the results are still preliminary and not much statistically significant in the evaluation of the whole corpus, some of the most common errors produced by the POS-tagger were corrected with very basic and linguistically motivated rules.

5 Conclusions and further work

In this paper we have showed a method using a POS-tagger and a grammar compiler that allows to write correction rules to improve the accuracy of the morphosyntactic annotation, correcting errors founded in regular syntactic contexts.

Since some of the errors produced by the POS-tagger follow regular linguistic patterns, the application of these rules can correct many of the mistaggings, namely those that could be considered as linguistically significant.

At this moment, some rules were tested in order to correct the most common errors. In further development, error contexts should be analyzed into more detail to verify the existence of more specific regular patterns that will allow us to write new rules to improve the accuracy of the system. It could also be possible to automate the search for new rules, evaluating its precision before adding them to the grammar.

References

1. Bick, Eckhard: *The Parsing System PALAVRAS: Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework.* Ph.D. thesis, University of Aarhus, Denmark (2000).
2. Branco, António and João Ricardo Silva: Evaluating Solutions for the Rapid Development of State-of-the-Art POS-taggers for Portuguese. In: *Proceedings of the 4th Language Resources and Evaluation Conference (LREC)*, pp. 507–510 (2004).
3. Carreras, Xavier, Isaac Chao, Lluís Padró and Muntsa Padró: FreeLing: An Open-Source Suite of Language Analyzers. In: *Proceedings of the 4th Language Resources and Evaluation Conference (LREC)* (2004).

4. DepPattern (GPL License), `http://gramatica.usc.es/pln/tools/deppattern.html`
5. Finger, Marcelo: Técnicas de Otimização da Precisão Empregadas no Etiquetador Tycho-Brahe. In: Maria das Graças Volpe Nunes (ed.) *Proceedings of the 5th Encontro para o Processamento Computacional da Língua Portuguesa Escrita e Falada (PROPOR)*, pp. 141–154. ICNC/USP, São Paulo (2000).
6. FreeLing. An Open Source Suite of Language Analyzers, `http://www.lsi.upc.edu/~nlp/freeling/`
7. Gamallo Otero, Pablo and Isaac González: Una gramática de dependencias basada en patrones de etiquetas. In: XXV Congreso Anual de la Sociedad Española para el Procesamiento del Lenguaje Natural. Donostia (2009).
8. Leach, Geoffrey and Andrew Wilson: Recommendations for the Morphosyntactic Annotation of Corpora. Expert Advisory Group on Language Engineering Standard (EAGLES) (1996).