Dedicated Nominal Featurization of Portuguese

António Branco and João Ricardo Silva

University of Lisbon, Department of Informatics
NLX—Natural Language Group
{ahb,jsilva}@di.fc.ul.pt

Abstract. A widespread assumption about the analysis of inflection features is that this task is to be performed by a tagger with an extended tagset. This typically leads to a POS precision drop due to the data-sparseness problem. In this paper we tackle this problem by addressing inflection tagging as a dedicated task, separated from that of POS tagging. More specifically, this paper describes and evaluates a rule-based approach to the tagging of Gender, Number and Degree inflection of open nominal morphosyntactic categories. This approach achieves a better F-measure than the typical approach of inflection analysis via stochastic state-of-the-art tagging.

1 Introduction

Inflective languages pose a problem for current stochastic taggers as besides the usual POS tags, tokens need also to be tagged with a variety of inflection information, such as the values for the features of Gender, Number, Degree, Case, Mood, Tense, etc. This requires an extended tagset [3] which usually leads to a lower tagging precision due to the data-sparseness problem.

In this paper, we address this problem by studying what can be gained when inflection tagging is envisaged as an autonomous task, separated from POS tagging. Nominal featurization is thus circumscribed as the task of assigning feature values for inflection (Gender and Number) and, if applicable, degree (Diminutive, Superlative and Comparative) to words from the nominal morphosyntactic open classes (Adjective and Common Noun).

In Section 2, the algorithm is outlined and evaluated. In Section 3, concluding remarks are presented.

2 Algorithm

The morphologic regularities found in Portuguese suggest a straightforward rule-based algorithm for the autonomous assignment of inflection feature values given that word terminations are typically associated with a default feature value. For example, most words ending in `-ção`, like `canção` (*Eng.: song*) are feminine singular. Any exceptions to this can then be easily found by searches in machine-readable dictionaries (MRD): The exceptions are words with the designed termination but with inflection features that do not match the default one. For instance, coração (*Eng.: heart*) is masculine singular.

Assigning inflection features can thus be done by simply searching for a suitable termination rule and assigning the corresponding default inflection tag if the input token is not one of the exceptions to that rule.

However, using rules and exceptions is not enough to ensure that every token receives an inflection tag. The main reason for this is the existence of "uniform" words, which are lexically ambiguous with respect to inflection feature values. For example, ermita (*Eng.: hermit*), depending on its specific occurrence, can be tagged as masculine or feminine. By using nothing more than rules and exceptions, the output produced by the rule-based featurizer would always be ermita/?S (singular, but with an underspecified value for Gender).

To handle these cases, an algorithm can be envisaged that builds upon the fact that there is Gender and Number agreement in Portuguese, including within NPs. All words from the closed classes that have inflection features (Demonstrative, Determiner, Quantifier, etc.) are collected together with their corresponding inflection tags. During inflection analysis of a text, the inflection tags assigned to words from these closed classes are "propagated" to the words from open classes (Adjective and Common Noun) that immediately follow them. These may, in turn, propagate the received tags to other words.

The example below illustrates how tag propagation disambiguates an occurrence of ermita based on the Definite Article that precedes it. The tag given to ermita can then be propagated to the adjective humilde (*Eng.: humble*), which is also a uniform word.

o/MS → ermita/MS → humilde/MS   a/FS → ermita/FS → humilde/FS
Eng.: the [masculine] humble hermit   *Eng.: the [feminine] humble hermit*

In order to make a sensible use of this idea, one just has to ensure that tag propagation occurs only within NP boundaries. For that effect, some patterns of tokens and POS tags are defined such that, when they are found, tag propagation is prevented from taking place.

For example, propagation may be prevented from crossing the conjunction e (*Eng.: and*) or punctuation symbols such as the comma. This allows to properly handle propagation over an enumeration of NPs and other similar structures:

$$\overbrace{\texttt{cão/MS branco/MS}}^{\texttt{NP}}\ ,\ \overbrace{\texttt{gatas/FP pretas/FP}}^{\texttt{NP}}\ \texttt{e}\ \overbrace{\texttt{peixe/MS azul/MS}}^{\texttt{NP}}$$

Eng.: white dog, black cats and blue fish

Note that by using this featurization algorithm it is still possible for a token to be tagged with an underspecified inflection tag. This happens not only due to some propagations being blocked (the blocking patterns have a "defensive" design, preventing some correct propagations to avoid tagging in error), but also due to the so-called bare NPs, which do not have a specifier preceding the head Noun, as in `Eu detesto ermitas/?P` (*Eng.: I hate hermits*). It also occurs in non-bare NPs, provided that the specifier is itself a uniform word, such as `cada/?S` (*Eng.: each*), which is lexically ambiguous with respect to its Gender.

The underspecified inflection tags that still remain after this propagation algorithm has been run cannot be accurately resolved at this stage. At this point, one can take the view that it is preferable to refrain from tagging than to tag incorrectly, and not attempt to resolve the underspecified tags without certainty. The resolution of these tokens can be left to the subsequent phase of syntactic processing which, taking advantage of NP-external agreement,[1] may resolve some of these cases (more in Section 3).

2.1 Implementation

The list of ca. 200 terminations and corresponding default inflection values was built from a reverse dictionary. The exceptions to these rules were gathered by resorting to a MRD and finding entries with each one of the 200 terminations but with inflectional features that differ from the default. This lead to a list of ca. $9,500$ exceptions, with an average of 47.5 exceptions for each inflection rule.

To implement the propagation procedure described above, a lexicon with ca. $1,000$ words from closed classes and respective inflection features was collected by searches in MRDs for entries with the relevant POS categories. Additionally, 9 patterns for blocking feature propagation were required.

The algorithm was implemented using Flex,[2] which provides an easy way for patterns (defined by regular expressions) in the input to trigger actions.

2.2 Evaluation

The rule-based featurizer does not necessarily assign a fully specified feature tag to every token. Following [4, pp. 268–269] for the measures of recall and precision,[3] it is important to note that, by virtue of the design of the algorithm, a precision score of 100% can in principle be reached provided that the list of exceptions to termination rules is exhaustive. However, in our experiment, some errors were found as the MRD used to collect exceptions is not large enough. These missing entries were, however, not added to the exceptions list, as this provides a way to replicate our results with the MRD that was used.

The rule-based featurizer was evaluated over a corpus with ca. $41,000$ tokens, where ca. $8,750$ where Adjectives and Common Nouns.

Firstly, the featurizer was evaluated over an accurately POS-tagged corpus. In this way, POS tagging mistakes will not negatively influence the outcome of the featurizer. The evaluation was then repeated but now over an automatically POS-tagged corpus.[4]

[1] Agreement holding between Subject and Verb, Subject and predicative complement in copular constructions with *be*-like verbs, etc.

[2] Flex—Fast lexical analyzer generator: `http://www.gnu.org/software/flex`.

[3] Recall is “the proportion of the target items that the system selected” and precision is “the proportion of selected items that the system got right”

[4] The POS tagger used was TnT [2], with 96.87% accuracy. [1]

	Correct POS			Automatic POS		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Stochastic	n/a	n/a	n/a	92.90%	100.00%	96.32%
Featurizer	99.05%	95.09%	97.03%	93.23%	95.06%	94.14%
w/ agreement	98.85%	99.88%	99.36%	93.55%	99.85%	96.60%

Table 1. Evaluation

When running over a correctly POS-tagged corpus, the featurizer is highly precise (99.05%). However, most application cases also require POS tagging to be done automatically. As the propagation mechanism is sensitive to POS errors, precision drops (to 93.23%) when running over an automatically POS-tagged corpus. These results are shown in Table 1.

3 Concluding Remarks

The main weakness of this rule-based approach is its recall score (around 95.1%), caused by the featurizer abstaining from tagging some uniform words in the hope that subsequent syntactic processing will solve those cases. An examination of a sample of 113 such cases showed that 97 of them could be resolved syntactically, leaving only ca. 16% of the underspecified tags still unresolved. Extrapolation from this result indicates that using syntactic processing to handle underspecified tags could lead to a great increase in recall. More specifically, taking the case of the featurizer running over automatically POS-tagged text, if only 16% of the underspecified tags were left unresolved, precision would be 93.55% and recall would increase to 99.85%, for an F-measure of 96.60%.

When compared with a typical stochastic approach, this featurization strategy turns out to be a better solution with better scores. In order to develop a tagger to assign POS tags extended with inflection values, we trained a tagger with TnT, that implements a HMM approach with back off and suffix analysis, and offers top scoring results for Portuguese tagging [1]. The resulting tool for inflection analysis presents an F-measure of only 96.32%. The results obtained are summarized in Table 1.

References

1. António Branco and João Silva. 2004. *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). 507–510.
2. Thorsten Brants. 2000. *TnT—A Statistical Part-of-Speech Tagger.* In Proceedings of the 6th Applied Natural Language Conference (ANLP). 224–231.
3. Jan Hajič and Barbora Hladká. 1997. *Probabilistic and Rule-based Tagger of an Inflective Language: A Comparison.* In Proceedings of the 5th ANLP. 111–118.
4. Christopher Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing.* The MIT Press.