A Suite of Shallow Processing Tools for Portuguese: LX-Suite

António Branco
Department of Informatics
University of Lisbon
ahb@di.fc.ul.pt

João Ricardo Silva
Department of Informatics
University of Lisbon
jsilva@di.fc.ul.pt

Abstract

In this paper we present LX-Suite, a set of tools for the shallow processing of Portuguese. This suite comprises several modules, namely: a sentence chunker, a tokenizer, a POS tagger, featurizers and lemmatizers.

1 Introduction

The purpose of this paper is to present LX-Suite, a set of tools for the shallow processing of Portuguese, developed by the NLX—Natural Language and Speech Group, at the Department of Informatics of the University of Lisbon, Faculty of Sciences.[1]

The tools included in this suite are a sentence chunker; a tokenizer; a POS tagger; a nominal featurizer; a nominal lemmatizer; and a verbal featurizer and lemmatizer.

These tools were implemented as autonomous modules. This option allows to easily replace any of the modules by an updated version or even by a third-party tool. It also allows to use any of these tools separately, outside the pipeline of the suite.

The evaluation results mentioned in the next sections have been obtained using an accurately hand-tagged $280,000$ token corpus composed of newspaper articles and short novels.

2 Sentence chunker

The sentence chunker is a finite state automaton (FSA), where the state transitions are triggered by specified character sequences in the input, and the emitted symbols correspond to sentence (`<s>`) and paragraph (`<p>`) boundaries. Within this setup, a transition rule could define, for example, that a period, when followed by a space and a capital letter, marks a sentence boundary:

"`...  A...`" → "`...</s><s>A...`"

Being a rule-based chunker, it was tailored to handle orthographic conventions that are specific to Portuguese, in particular those governing dialog excerpts. This allowed the tool to reach a very good performance, with values of 99.95% for recall and 99.92% for precision.[2]

3 Tokenizer

Tokenization is, for the most part, a simple task, as the whitespace character is used to mark most token boundaries. Most of other cases are also rather simple: Punctuation symbols are separated from words, contracted forms are expanded and clitics in enclisis or mesoclisis position are detached from verbs. It is worth noting that the first element of an expanded contraction is marked with a symbol (+) indicating that, originally, that token occurred as part of a contraction:[3]

`um, dois` → `|um|,|dois|`
`da` → `|de+|a|`
`viu-o` → `|viu|-o|`

In what concerns Portuguese, the non-trivial aspects of tokenization are found in the handling of ambiguous strings that, depending on their POS tag, may or may not be considered a contraction. For example, the word `deste` can be tokenized as the single token `|deste|` if it occurs as a verb (*Eng.: [you] gave*) or as the two tokens `|de+|este|` if it occurs as a contraction (*Eng.: of this*).

[1] `http://nlx.di.fc.ul.pt`

[2] For more details, see (Branco and Silva, 2004).

[3] In these examples the | symbol will be used to mark token boundaries more clearly.

It is worth noting that this problem is not a minor issue, as these strings amount to 2% of the corpus that was used and any tokenization error will have a considerable negative influence on the subsequent steps of processing, such as POS tagging.

To resolve the issue of ambiguous strings, a two-stage tokenization strategy is used, where the ambiguous strings are not immediately tokenized. Instead, the decision counts on the contribution of the POS tagger: The tagger must first be trained on a version of the corpus where the ambiguous strings are not tokenized, and are tagged with a composite tag when occurring as a contraction (for example `P+DEM` for a contraction of a preposition and a demonstrative). The tagger then runs over the text and assigns a simple or a composite tag to the ambiguous strings. A second pass with the tokenizer then looks for occurrences of tokens with a composite tag and splits them:

`deste/V` $\rightarrow$ `|deste/V|`
`deste/P+DEM` $\rightarrow$ `|de+/P|este/DEM|`

This approach allowed us to successfully resolve 99.4% of the ambiguous strings. This is a much better value than the baseline 78.20% obtained by always considering that the ambiguous strings are a contraction.[4]

4 POS tagger

For the POS tagging task we used Brant's TnT tagger (Brants, 2000), a very efficient statistical tagger based on Hidden Markov Models.

For training, we used 90% of a $280,000$ token corpus, accurately hand-tagged with a tagset of ca. 60 tags, with inflectional feature values left aside.

Evaluation showed an accuracy of 96.87% for this tool, obtained by averaging 10 test runs over different 10% contiguous portions of the corpus that were not used for training.

The POS tagger we developed is currently the fastest tagger for the Portuguese language, and it is in line with state-of-the-art taggers for other languages, as discussed in (Branco and Silva, 2004).

5 Nominal featurizer

This tool assigns feature value tags for inflection (Gender and Number) and degree (Diminutive, Superlative and Comparative) to words from nominal morphosyntactic categories.

[4]For further details see (Branco and Silva, 2003).

Such tagging is typically done by a POS tagger, by using a tagset where the base POS tags have been extended with feature values. However, this increase in the number of tags leads to a lower tagging accuracy due to the data-sparseness problem. With our tool, we explored what could be gained by having a dedicated tool for the task of nominal featurization.

We tried several approaches to nominal featurization. Here we report on the rule-based approach which is the one that better highlights the difficulties in this task.

For this tool, we built on morphological regularities and used a set of rules that, depending on the word termination, assign default feature values to words. Naturally, these rules were supplemented by a list of exceptions, which was collected by using an machine readable dictionary (MRD) that allowed us to search words by termination.

Nevertheless, this procedure is still not enough to assign a feature value to every token. The most direct reason is due to the so-called invariant words, which are lexically ambiguous with respect to feature values. For example, the Common Noun `ermita` (*Eng.: hermit*) can be masculine or feminine, depending on the occurrence. By simply using termination rules supplemented with exceptions, such words will always be tagged with underspecified feature values:[5]

`ermita/?S`

To handle such cases the featurizer makes use of feature propagation. With this mechanism, words from closed classes, for which we know their feature values, propagate their values to the words from open classes following them. These words, in turn, propagate those features to other words:

`o/MS ermita/MS humilde/MS`
Eng.: the-MS humble-MS hermit-MS
but
`a/FS ermita/FS humilde/FS`
Eng.: the-FS humble-FS hermit-FS

Special care must be taken to avoid that feature propagation reaches outside NP boundaries. For this purpose, some sequences of POS categories block feature propagation. In the example below, a PP inside an NP context, `azul` (an "invariant"

[5]Values: `M`:masculine, `F`:feminine, `S`:singular, `P`:plural and `?`:undefined.

adjective) might agree with faca or with the preceding word, aço. To prevent mistakes, propagation from aço to azul should be blocked.

faca/FS de aço/MS azul/FS
Eng.: blue (steel knife)
or
faca/FS de aço/MS azul/MS
Eng.: (blue steel) knife

For the sake of comparability with other possible similar tools, we evaluated the featurizer only over Adjectives and Common Nouns: It has 95.05% recall (leaving ca. 5% of the tokens with underspecified tags) and 99.05% precision.[6]

6 Nominal lemmatizer

Nominal lemmatization consists in assigning to Adjectives and Common Nouns a normalized form, typically the masculine singular if available.

Our approach uses a list of transformation rules that helps changing the termination of the words. For example, one states that any word ending in ta should have that ending transformed into to:

gata (*[female] cat*)
→ gato (*[male] cat*)

There are, however, exceptions that must be accounted for. The word porta, for example, is a feminine common noun, and its lemma is porta:

porta (*door*, feminine common noun)
→ porta

Relevant exceptions like the one above were collected by resorting to a MRD that allowed to search words on the basis of their termination. Being that dictionaries only list lemmas (and not inflected forms), it is possible to search for words with terminations matching the termination of inflected words (for example, words ending in ta). Any word found by the search can thus be considered as an exception.

A major difficulty in this task lies in the listing of exceptions when non-inflectional affixes are taken into account. As an example, lets consider again the word porta. This word is an exception to the rule that transforms ta into to. As expected, this word can occur prefixed, as in superporta. Therefore, this derived word should also appear in the list of exceptions to prevent it from being lemmatized into superporto by the rule. However, proceeding like this for every possible prefix leads to an explosion in the number of exceptions. To avoid this, a mechanism was used that progressively strips prefixes from words while checking the resulting word forms against the list of exceptions:

supergata
-----gata (apply rule)
→ supergato
but
superporta
-----porta (exception)
→ superporta

A similar problem arises when tackling words with suffixes. For instance, the suffix -zinho and its inflected forms (-zinha, -zinhos and -zinhas) are used as diminutives. These suffixes should be removed by the lemmatization process. However, there are exceptions, such as the word vizinho (*Eng.: neighbor*) which is not a diminutive. This word has to be listed as an exception, together with its inflected forms (vizinha, vizinhos and vizinhas), which again leads to a great increase in the number of exceptions. To avoid this, only vizinho is explicitly listed as an exception and the inflected forms of the diminutive are progressively undone while looking for an exception:

vizinhas (feminine plural)
vizinha (feminine singular)
vizinho (exception)
→ vizinho

To ensure that exceptions will not be overlooked, when both these mechanisms work in parallel one must follow all possible paths of affix removal. An heuristic chooses the lemma as being the result found in the least number of steps.[7]

To illustrate this, consider the word antena (*Eng.: antenna*). Figure 1 shows the paths followed by the lemmatization algorithm when it is faced with antenazinha (*Eng.: [small] antenna*). Both ante- and -zinha are possible affixes. In a first step, two search branches are opened, the first where ante- is removed and the second where -zinha is transformed into

[6] For a much more extensive analysis, including a comparison with other approaches, see (Branco and Silva, 2005a).

[7] This can be seen as following a rationale similar to Karlsson's (1990) local disambiguation procedure.

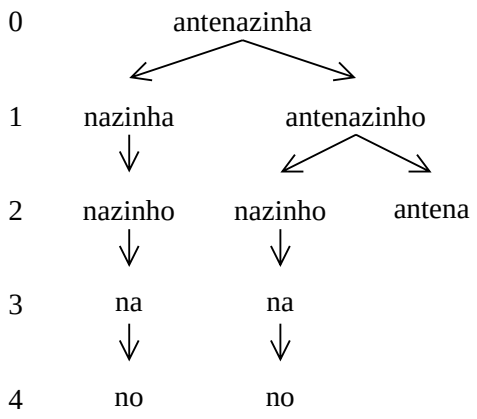

Figure 1: Lemmatization of `antenazinha`

`-zinho`. The search proceeds under each branch until no transformation is possible, or an exception has been found. The end result is the "leaf node" with the shortest depth which, in this example, is `antena` (an exception).

This branching might seem to lead to a great performance penalty, but only a few words have affixes, and most of them have only one, in which case there is no branching at all.

This tool evaluates to an accuracy of 94.75%.[8]

7 Verbal featurizer and lemmatizer

To each verbal token, this tool assigns the corresponding lemma and tag with feature values for Mood, Tense, Person and Number.

The tool uses a list of rules that, depending on the termination of the word, assign all possible lemma-feature pairs. The word `diria`, for example, is assigned the following lemma-feature pairs:

`diria`
→ ⟨`dizer,Cond-1ps`⟩
→ ⟨`dizer,Cond-3ps`⟩
→ ⟨`diriar,PresInd-3ps`⟩
→ ⟨`diriar,ImpAfirm-2ps`⟩

Currently, this tool does not attempt to disambiguate among the proposed lemma-feature pairs. So, each verbal token will be tagged with all its possible lemma-feature pairs.

The tool was evaluated over a list with ca. $800,000$ verbal forms. It achieves 100% precision, but at 50% recall, as half of those forms are ambiguous and receive more than one lemma-feature pair.

[8]For further details, see (Branco and Silva, 2005b).

8 Final Remarks

So far, LX-Suite has mostly been used in-house for projects being developed by the NLX Group.

It is being used in the GramaXing project, where a computational core grammar for deep linguistic processing of Portuguese is being developed under the Delphin initiative.[9]

In collaboration with CLUL,[10] and under the TagShare project, LX-Suite is being used to help in the building of a corpus of 1 million accurately hand-tagged tokens, by providing an initial, high-quality tagging which is then manually corrected.

It is also used for the QueXting project, whose aim is to make available a question answering system on the Portuguese Web.

There is an on-line demo of LX-Suite located at `http://lxsuite.di.fc.ul.pt`. This on-line version of the suite is a partial demo, as it currently only includes the modules up to the POS tagger. By the end of the TagShare project (mid-2006), all the other modules described in this paper are planned to have been included. Additionally, the verbal featurizer and lemmatizer can be tested as a standalone tool at `http://lxlemmatizer.di.fc.ul.pt`.

Future work will be focused on extending the suite with new tools, such as a named-entity recognizer and a phrase chunker.

References

Branco, António and João Ricardo Silva. 2003. *Contractions: breaking the tokenization-tagging circularity*. LNAI 2721. pp. 167–170.

Branco, António and João Ricardo Silva. 2004. *Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese*. In Proc. of the 4th LREC. pp. 507–510.

Branco, António and João Ricardo Silva. 2005a. *Dedicated Nominal Featurization in Portuguese*. ms.

Branco, António and João Ricardo Silva. 2005b. *Nominal Lemmatization with Minimal Word List*. ms.

Brants, Thorsten. 2000. *TnT - A Statistical Part-of-Speech Tagger*. In Proc. of the 6th ANLP.

Karlsson, Fred 1990. *Constraint Grammar as a Framework for Parsing Running Text*. In Proc. of the 13th COLING.

[9]`http://www.delph-in.net`

[10]Linguistics Center of the University of Lisbon