Portuguese-specific Issues in the Rapid Development of State-of-the-art Taggers

António Branco and João Silva

Department of Informatics, University of Lisbon
Faculdade de Ciências, Campo Grande, 1749-016 Lisboa
{ahb,jsilva}@di.fc.ul.pt

Abstract

The application of general-purpose machine learning techniques to natural language part-of-speech tagging has matured to a point where it is now quite rapid to develop new taggers. In the present paper, we report on solutions we adopted for the specific issues that arise when developing a new automatic tagger for Portuguese and are generic enough to be further reused to develop other new taggers for this language, possibly by using other training data.

1. Introduction

Lexemes with the same syntactic distribution are grouped together and assigned the same part-of-speech (POS) tag (e.g. `Noun`, `Adjective`, `Preposition`, etc.). Many lexemes belong to more than one such distributional grouping thus implying that many lexeme-types bear more than one tag in the lexicon and that the correct tag for each of their lexeme-tokens has to be decided given the specific occurrence at stake.

From a computational point of view, the non trivial issue with respect POS tagging consists in deciding for each token of a lexeme in a text, and from the set of admissible POS tags for its type in the lexicon, which tag is the correct one to be assigned to that lexeme in that specific occurrence. Though apparently simple when synthetised under these terms, POS tagging is a very important step in natural language processing inasmuch as it permits to cut down a considerable amount of ambiguity present in natural language utterances at a quite early stage of processing, even before the subsequent, and computationally expensive stages of syntactic and semantic processing.

The application of general-purpose machine learning techniques to natural language POS tagging has matured to a point where it is now quite rapid to develop new taggers. As a matter of fact, when using the applications making use of such techniques to develop a new tagger, the time span needed to set it up is determined basically by the language-specific issues that have to be dealt with. Such issues are found in each of the three major steps involved in automatic tagging raw text: chunking, tokenizing and tagging stricto sensu.

In the present paper, we report on solutions we worked out for the specific issues that arise when developing a new automatic tagger for Portuguese and are generic enough to be further reused with benefit to develop other new taggers for this language from other training data.

2. Chunker

As in other languages with orthographic conventions similar to those adopted for Portuguese, designated punctuation symbols ('`.`', '`?`', '`!`',...) are used to mark the end of sentences. Most sentence boundaries can then be detected when these terminators precede sentence starters, i.e. designated orthographic clues marking the beginning of a subsequent sentence (viz. word beginning with a capital letter) — the expected abbreviation/period ambiguity of '`.`' can be addressed by means of the solutions proposed in the literature for other languages (Mikheev, 2002).

Conventions for sentence bounding that are specific to Portuguese, or at least not found in other close Romance languages or English under exactly the same format, involve the marking of paragraph (turn taking) and sentence boundaries in written dialogue.

The beginning of the first sentence containing a character's turn is easily handled as this starts with a dash ('–') immediately followed by the usual sentence starters.

```
<s> - Bom dia! </s>
```

Things get convoluted, however, when it comes to narrator's asides: the beginning of a narrator's aside is always indicated by a dash but its ending is also indicated by a dash only in the cases where the aside does not conclude the sentence.

```
<p><s> - Apetece-me ir ao cinema – anunciou ele. </s></p>
<p><s> - Eu cá - disse ela – também quero. </s></p>
```

Taking narrator's asides into account, it is worth noting that a character's sentence other than the first one in the current turn starts also with a dash exactly in the cases where such sentence follows a character's sentence ending with a narrator's aside.

```
<p><s> - Não - disse ela. </s><s> - Eu não. </s></p>
```

As for termination symbols of character's utterances, only those that are different from a period appear before the beginning of a narrator's aside.

```
<s> - Bom dia! - exclamou. </s>
```

Other hard cases involve the determination of sentence/paragraph boundaries indicated by starters of enumerated lists and quotation delimiters and by the starter/terminator ambiguity of ellipsis ('...').

These issues will be discussed in detail in the presentation and a systematic procedure to handle them will be outlined. For this procedure, we scored a recall of 99.94% and precision of 99.93% when tested on a 12 000 sentence corpus accurately hand tagged with respect to sentence and paragraph boundaries.

3. Tokenizer

For most tokens in a raw text, tokenization is a trivial procedure, consisting in detaching punctuation marks and taking advantage of the whitespace as a delimiter symbol. There are, however, a few non-trivial cases (complete list to be presented at the workshop) that involve tokenization-ambiguous strings, i.e. strings that can be tokenized in more than one way.

`deste -> |deste|` or `deste -> |de|este|`.

In a general setup like ours, where one counts on a tagger trained over previously annotated data, this inevitably introduces circularity that has to be resolved: Although all tagging decisions require previous tokenization, the tokenization of these ambiguous strings requires previous knowledge of the POS tag of the token(s) corresponding to the string. In the example above, we would tokenize `deste` as one token only if it had been tagged as a Verb, but for it to be tagged as a Verb it should have already been tokenized as one token.

To resolve these cases, we used a two-level approach to tokenization where tagging is interpolated into the tokenization process, which has now two stages, one before and another after the tagger has been applied. Accordingly, (i) a pre-tagging tokenizer definitely identifies every token except those related to ambiguous strings: These strings are provisionally identified as one token.

(ii) Subsequently, the tagger assigns a composite or a simple tag to every ambiguous string depending on it being a contracted or a non-contracted form, respectively: The tagger has been trained over a corpus where ambiguous strings are always tokenized as a single token and annotated with single or composite tags.

(iii) Finally, a post-tagging tokenizer handles only ambiguous strings, breaking those that are tagged with a composite tag into two tokens and the corresponding tags.

In our corpus, the ambiguous strings amount to 2% of the tokens. This two-level tokenization approach permitted to successfully resolve 99.4% of these ambiguous cases, against a baseline of 78.2% of success, which is obtained by tokenizing every such ambiguous string as two tokens in every occurrence (as 78.2% of the ambiguous strings were contractions in our text corpus).

4. Tagger

For the development of the Portuguese tagger strictu sensu, we used the TnT software, a Hidden Markov model based application developed and kindly granted to us by Thorsten Brants (Brants, 2000). When using a machine-learning tool like this to develop a new tagger, the critical issues are to be found in the gathering of appropriate training data. Assuming that the consistency and accuracy of the annotation of the general purpose training corpus used as a starting point is ensured, the main concern is directed towards manipulating and relabeling it in accordance with the tag set that needs to be opted for. The design of the latter turns out thus to be the non-trivial aspect that calls to be addressed.

In this respect, one finds the usual tension between increasing the discriminative power of the tagger — by using more tags — and minimizing the data sparseness — by using fewer tags. Looking for the best performance of a POS tagger supported by a suitably tuned balance of these two attractors cannot be reduced, however, to arbitrarily playing around with the number and the assignment of tags: By definition, a syntactic category identifies, under the same tag, tokens with identical syntactic distribution, i.e. tokens that, in any occurrence receiving that tag, can replace each other while preserving the grammaticality of the linguistic construction, modulo the adoption of suitable subcategorisation constraints impinging over them. If the goal is the development of a top-accuracy tagger that optimally supports subsequent syntactic parsing, this is the criterion that we cannot lose sight of in the choice of the tag set.

Accordingly, there are possible "candidate" categories or subcategories that can or should be excluded:

(i) Tags not justified by distributional facts, e.g. those indicating the degree of an adjective (example: `alto_ADJNORM`, `altíssimo_ADJSUP`);

(ii) Tags that tough conveying some distribution-related information can be unequivocally inferred from the form of the token, e.g. those indicating the polarity of an adverb (example: `sim_ADVPOS`; `nem_ADVNEG`), or inferred from its suffixes (example: `alto_ADJMascSing`, `altas_ADJFemPlu`);

(iii) Tags indicating the constituency status of the containing phrase but not a difference in syntactic distribution, e.g. the category of "indefinite pronouns/adjectives" used to mark articles,

demonstratives and other prenominals in headless Noun Phrases (example: `li [aquele_DEM livro]`$_{NP}$; `li [aquele_INDPRON Ø]`$_{NP}$) — note that a tag `IN` (Indefinite Nominals) for single word NPs like `tudo` was kept in the tag set.

This rationale, followed to circumscribe the tag set, not only helped to exclude possible tags, but also to isolate and include categories that are usually not taken into account in a more traditional perspective. Though being verbal forms, gerund, past participle and infinitive forms each have a distribution of its own: The tags `GER`, `PTP` and `INF` were thus included in the tag set.

Other non-canonical tags were also included: These may be less interesting from a general linguistic point of view but they are important to enhance the contribution of the tagger for subsequent processing stages, e.g. named entity recognition. We isolated social titles (`Pres.`, `Drª.`, `prof.`,...), part of addresses (`Rua`, `Av.`, `Rot.`,...), months, week days, measurement units (`km`, `kg`, `b.p.m.`,...), etc. as distinct syntactic classes. Our tag set includes also specific tags for roman numerals, denominators of fractions (`meio`, `terço`, `décimo`, `%`,...), and letters.

With the tag set defined (the complete list will be presented at the workshop), we prepared a training corpus by converting and adjusting the initial tagged corpus, a 230 Ktoken, hand tagged corpus kindly granted by CLUL.

With these data and the help of the TnT tool, a tagger for Portuguese was developed with 97.2% accuracy — a value obtained with one run test over a held out evaluation corpus with the 10% not used for training. This result is in line with the state-of-the-art performance obtained for German (96.7%) or English (96.7%) with the same tool over, respectively, the NEGRA Corpus (320 Ktokens) and the Penn Treebank (1.2 Mtokens) corpora, and an accuracy measurement averaged over 10 test runs (Brants, 2000).

5. References

Brants, T., 2000, "TnT-A Statistical Part-of-speech Tagger". *Proc. of ANLP2000*, 224-231.

Brill, E., 1995, "Transformation-based Error-driven Learning and Natural Language Processing: A case study in part-of-speech tagging". *Computational Linguistics*, 21, 543-565

Mikheev, A., 2002, "Periods, Capitalized Words, etc." *Computational Linguistics* 28(3), 289-318.

Rathnaparkhi, A., 1996, "A Maximum Entropy Part-of-speech Tagger". *Proc. EMNLP'96*, 133-142.

Samuelsson, C. and A. Voutilainen, 1997, "Comparing a Linguistic and a Stochastic Tagger". *Proc. ACL'97*, 246-253.