INTERPOLATED PROBABILISTIC TAGGING MODEL OPTIMIZED WITH GENETIC ALGORITHM

FAI WONG¹, SAM CHAO¹, DONG-CHENG HU², YU-HANG MAO²

¹Faculty of Science and Technology of University of Macao, PO Box 3001, Macao SAR
²Speech and Language Processing Research Center, Tsinghua University 100084 Beijing
E-MAIL: {derekfw, lidiasc}@umac.mo, {hudc, myh-dau}@mail.tsinghua.edu.cn

Abstract:

In this paper, we present results of probabilistic tagging of Portuguese texts in order to show how these techniques work for one of the highly morphologically ambiguous inflective languages by using a limited corpus as the basic training source. In order to cope the ambiguities problem caused by the insufficient training data, especially the unknown words, we incorporate the lexical features into the probabilistic model. Different from other proposed tagging models, these features are introduced into the word probabilities by means of interpolation. A technique to determine the optimal set of interpolation parameters based on genetic algorithm is described. Our preliminary result shows that we can correctly tag 91.8% of the sentences based on our tagging model.

Keywords:
POS tagging; genetic algorithm; probabilistic model; Portuguese tagging

1. Introduction

Many words are ambiguous in their part of speech. For example, “acordo (agreement/wake up)” can be a noun or a verb. However, when a word appears in the context of other words, the ambiguity is often reduced: in “O acordo tem a duracao de 10 anos (The agreement has a duration of 10 years.)” the word "acordo (agreement)" can only be a noun. Tagging is the task of labeling (or tagging) each word in a sentence with its appropriate part of speech. Automatic text tagging is an important first step in discovering the linguistic structure of large text corpora. Part-of-speech information facilitates higher-level analysis, such as recognizing noun phrases and other patterns in text. It is the intermediate task within the natural language processing applications such language understanding, machine translation, etc.

However, part of speech tagging has many difficult problems to attack such as insufficient training data, inherent POS ambiguities, and most seriously unknown words. Unknown words are ubiquitous in any application and cause major tagging failures in many cases. By referring to any natural language processing conference proceedings and literature reports from 90s, we found that the majority of frequently processed languages are English, Chinese, French, German, Italian, and Spanish. And very few reports can found are related to Czech [1] and Swedish [2] in writing text tagging, and Portuguese [3] for spoken corpus tagging. There are many reasons but the key reason lies in the absence of the main resources - corpora for these languages.

2. Related work

Most of the recent corpus-based POS taggers in the literature are either statistically based, and use Markov Model [4] or statistical decision tree [5] techniques, or are primarily rule based, such as Brill's transformation based learner [6]. Each approach has some advantages: for stochastic techniques there exists a good theoretical framework, probabilities provide a straightforward way how to disambiguate tags for each word and probabilities can be acquired automatically from the data; for rule-based techniques the set of meaningful rules is automatically acquired and there exists an easy way how to find and implement improvements of the tagger. Small set of rules can be used, in contrast to the large statistical tables. Given the success of statistical methods in different areas, including text tagging, given the very positive results of English statistical taggers and given the fact that there existed no statistical tagger for Portuguese language (at least there is no any free applicable Portuguese POS tagger available from the internet), we wanted to apply statistical methods even for the Portuguese language although it exhibits a rich inflection accompanied by a high degree of ambiguity¹. Another consideration in our work is that

¹ Compare to English, Portuguese has very rich
language like Portuguese that, in contrast to languages like English, has limited resources in the form of digital corpora, computational lexicons, grammars or annotated Treebank.

In order to resolve the ambiguity and unknown words problem in our probability tagging model, lexical morphemes information are taken into consider for guessing the proper part of speech for a word within the context. In this paper, we introduce a probabilistic model by inducing the interpolation [7-8] to smooth the probability for unknown word by incorporating the lexical morphemes such as suffixes and prefixes into the word probability. In order to investigate and selected the most informative features of word that contribute to the performance of the tagger, a technique in finding the set of interpolation coefficient by using genetic algorithm is described, as well as experiment is made to demonstrate our proposed model in this paper.

3. The problem of tagging

In our work, we have used well-known probability models known as Hidden Markov Models; therefore, none of the background here is novel. If we want to determine the most likely syntactic part of speech or tag for each word in a sentence, we can formulate a probabilistic tagging model. Let us assume that we want to know the most likely tag sequence, \( T = \{t_1, t_2, \ldots, t_n\} \), given a particular word sequence, \( W = \{w_1, w_2, \ldots, w_n\} \). Using Bayes’ rule we have that:

\[
p(T | W) = \frac{p(W | T)p(T)}{p(W)} \tag{1.1}
\]

Where \( p(T) \) is the priori probability of tag sequence \( T \), \( p(W | T) \) is the conditional probability of word sequence \( W \) occurring given that a sequence of tags \( T \) occurred, and \( p(W) \) is the unconditioned probability of word sequence \( W \). Then, to consider all possible tag sequences, we need to evaluate \( p(T | W) \) of each, and choose the tag sequence \( T \) that is most likely, i.e., the sequence that maximizes \( p(T | W) \). Since \( W \) is the same for all hypothesized tag sequences, we can disregard \( p(W) \).

Now, the probability of each sequence can be rewritten as a product of the conditional probabilities of each word or tag given all of the previous tags:

\[
p(T | W) \approx \prod_{i=1}^{n} p(t_i | t_{i-1}, t_{i-2}, \ldots, t_1) p(w_i | t_1, \ldots, t_i, w_{i-1}, \ldots, w_1) \tag{1.2}
\]

Typically, two simplifying assumptions can be made to cut down on the number of probabilities to be estimated. Rather than assuming \( w_i \) depends on all previous words and all previous tags, one assumes \( w_i \) depends only on \( t_i \). Similarly, rather than assuming the tag \( t_i \) depends on the full sequence of previous tags, we can assume that local context is sufficient, that tag \( t_i \) depends only on previous tag \( t_{i-1} \). Then using a bi-gram model, we then have the following:

\[
\arg \max_T p(T | W) \approx \arg \max_T \prod_i p(t_i | t_{i-1}) p(w_i | t_i) \tag{1.3}
\]

If we have sufficient training data, we can estimate the tag n-gram sequence of probabilities and the probability of each word given a tag (lexical probabilities). Unfortunately, we do not have such kind of training corpus and we believe it is impossible to have it even in the future.

4. Resolving unknown words

Based on the training data, word emitting probabilities can be estimated by investigating the occurrences of words in the corpus. But many words in sentences that we want to tag will not be appeared in the training corpus. Some words will not even be in the dictionary, especially the conjugations of verb in highly inflective language like Portuguese. The occurrence of unknown words is very common in the tagging problem.

To resolving the unknown words problem in POS tagging, in rule-based approach, Brill [6] used the TBL algorithm, where the allowable templates were defined orthographically. The algorithm induced all the English inflectional features and many other derivational features for use in guessing the unseen words. In probabilistic method, Weischedel [9] developed a probabilistic model that takes into account features of the word in determining the likelihood of the word given a part of speech. In estimating the probability for unknown word, he treated the word as a set of independent features \( w = \{f_1, f_2, \ldots, f_j\} \). He used four specific kinds of orthographic features: inflectional endings, derivational endings, capitalization and hyphenation, and are expressed by the following equation to compute the likelihood of an unknown word:
\[
p(w_i | t_1) = p(\text{unknown word} | t_1) \times \frac{p(\text{capital} | t_1) \times p(\text{endings} | t_1)}{p(t_1)}
\] (1.4)

### 4.1. Features interpolation

In our model, we also consider the orthographic features of words for guessing the unknown words during the tagging process. Mainly we focus on the features of: lexical suffixes and the capitalization. Instead of carefully selected some specific endings for inflectional and derivational words as Weischedel [9], we take several suffixes with length of three to five characters respectively. The central idea behind our formulation is that, when we want to assign a tag \( t_i \) to a given word \( w_i \), if it is an unseen word to the model, we would like to guess the possible part of speech by investigating the endings of the word. Since believe this information is relating to the word itself. Therefore, in order to capture this behavior, we interpolate the word features into the word probability. Now, consider the following equation:

\[
p(w_i | t_1) = \frac{p(t_1 | w_i) p(w_i)}{p(t_1)}
\] (1.5)

The probabilities of this lexical information are interpolated in the word probability as expressed in the following equation:

\[
P_{\text{interp}}(w_i | t_1) \approx \frac{p(w_i)}{p(t_1)} \left[ \lambda_1 p(t_1 | w_i) + \lambda_2 p(t_1 | \text{capital}) + \sum_{f \in \text{suffixes}} \lambda_i p(t_1 | w_i \geq f) \right]
\] (1.6)

\[
P_{\text{interp}}(w_i | t_1) \approx \frac{p(w_i)}{p(t_1)} \left[ \lambda_1 p(t_1 | w_i) + \lambda_2 p(t_1 | \text{capital}) + \sum_{f \in \text{suffixes}} \lambda_i p(t_1 | w_i \geq f) \right]
\] (1.7)

Then, equation (1.7) is obtained after the simplification. Where \( \lambda_i (i = 1, \ldots, 5) \) are the interpolation coefficients and \( \sum_i \lambda_i = 1 \), the word probability \( p(w_i) \) is the uniformly distribution.

### 4.2. Features for part-of-speech tagging

In the exploitation of other features of word and its context to improve the lexical probability estimates for unknown words, we can use morphological and other clues to make inferences about a word’s possible parts of speech. Current technology depends on handcrafted linguistic and domain knowledge. For example, Weischedel [9] used four specific kinds of lexical features: 3 inflectional endings (-ed, -s, -ing), 32 derivational endings (including -ion, -al, -ive, -ed), 4 values of capitalization and hyphenation. Cha [10], in their Korean tagging system, applied the morpheme pattern dictionary that covers all necessary syllable patterns for unknown morphemes including common nouns, proper nouns, adnominal, adverbs and etc., even the special symbols for foreign words. These lexical patterns for morphemes are carefully studied and collected.

In our system, the model generates the space of features by scanning each pair \((t_i, f_i)\) in the training data with the specified feature templates, i.e. suffixes with specific length of three to five and the capitalization of prefix, where \( f_i \in (\text{capital}, \text{suffixes}) \). And the probabilities of \( p(t_i | f_i) \) can be estimated for the features space, where \( p(t_i | f_i) \) is the conditional probability of possible tag \( t_i \) occurring given that a feature \( f_i \) of word is observed.

### 4.3. Finding the most informative features

Interpolated model allows arbitrary features on the context, so it can use additional features that may correctly tag the words. Since such features typically occur infrequently, the training set consistency must be good enough to yield reliable statistics. Otherwise particular feature will model noise and perform poorly on the test data. In order that informative features of word can be more contributed towards probability of unknown word for guessing the correct tag, it is possible to search for the interpolation parameters \( \lambda_i \) using the deleted interpolation [11] technique where different parts of the training data rotate in training for \( \lambda_i \), the results are then averaged.

Here, we use the genetic algorithm (GA) [12] to achieve the same goal. Since the search of an optimal solution in genetic algorithm is heuristic by its nature. Possible solutions are suggested and fitness values obtained for the solutions. Then GA, through generations of evolution, provides the possible optimal solutions. Here, the search is formulated as optimization problem:
\[
\begin{align*}
\min \text{TagError}(\lambda_1, \lambda_2, \ldots, \lambda_i) \\
\text{subject to} \\
\sum_{i} \lambda_i &\leq 1 \\
\lambda_i &\geq 0, \quad i = 1\ldots5
\end{align*}
\] (1.8)

Where \( \text{TagError}(\lambda_1, \lambda_2, \ldots, \lambda_i) \) is the objective function that counts the number of mis-tagged words under the set of interpolation parameters. The set of model parameters \( \lambda_i \) are coded as a chromosome in GA.

In order to yield meaningful results, the data used to estimate the parameters \( \lambda_i \) need to be different from the data used to train the model. Therefore, in our searching method, a section of the training data is reserved for this purpose. In finding the optimal set of parameters (or set of related most informative features), we run the genetic algorithm with 30 generations under the configurations of: population size is 50, probability of crossover is 0.3, probability of mutation is 0.2, and the parameter \( a \) in the rank-based evaluation function is 0.05.

5. Experiments

For the experiments described herein, we used the Tycho Brahe (Parsed Corpus of Historical Portuguese) [13] described in Galves [14]. It contains 44,416 sentences (about one million words) from the Institute of Mathematics and Statistics of the University of São Paulo. The sentences have been tagged manually at the University of Campinas in the lines of the Penn-Helsinki Parsed Corpus of Middle English. In the Tycho Brahe corpus, 154 different tags are used. These tags were projected on a smaller system of 41 tags by maintaining the first level of tags. The results quoted in this paper all refer to this smaller system. Table 1 shows part of the tags used in the Tycho Brahe corpus.

We built a dictionary that indicates the list of possible tags for each word, by taking all the words that occur in this text and, for each word, all the tags that are assigned to it somewhere in the text. In some sense, this is an optimal dictionary for this data, since a word will not have all its possible tags (in the language), but only the tags that it actually had within the text.

We separated this data into two parts: 1) a set of 40,417 tagged sentences (90% of whole), the training data, which is used to build the models; and 2) another set of 4,000 tagged sentences (10%, around 89,000 words), the test data, which is used to test the quality of the models.

5.1. Experiment one

In the first experiment, we equally assign the interpolation coefficients \( \lambda_i = \frac{1}{5}, i = (1..5) \), in our model. Then we extracted N percentage of tagged sentences from the training data. We then compute the relative frequencies on these sentences and built an interpolated model as described previously. This model was then used to tag the 4,000 test sentences. We experimented with different values of N, for each of which we indicate the number and percentage of correctly tagged words. Results are indicated in Table 2. As expected, as the size of the training increases, the quality of the tagging improves.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>TAG</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SER</td>
<td>to be - individual level)</td>
<td>SR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SR-F</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SR-I</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SR-P</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SR-SP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SR-D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SR-R</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SR-SR</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SR-G</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SR-PP</td>
</tr>
</tbody>
</table>
Table 2. Training on % sentences, $\lambda_i = \frac{1}{5}, i = (1..5)$

<table>
<thead>
<tr>
<th>Training data (%)</th>
<th>Number of error (words)</th>
<th>Correct tag (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>16058</td>
<td>82.13</td>
</tr>
<tr>
<td>40</td>
<td>13559</td>
<td>84.91</td>
</tr>
<tr>
<td>50</td>
<td>12794</td>
<td>85.76</td>
</tr>
<tr>
<td>60</td>
<td>12761</td>
<td>85.80</td>
</tr>
<tr>
<td>70</td>
<td>12498</td>
<td>86.09</td>
</tr>
<tr>
<td>80</td>
<td>11655</td>
<td>87.03</td>
</tr>
<tr>
<td>90</td>
<td>11655</td>
<td>87.03</td>
</tr>
</tbody>
</table>

Table 3. Training on % sentences, model parameters optimized by genetic algorithm

<table>
<thead>
<tr>
<th>Training data (%)</th>
<th>Number of error (words)</th>
<th>Correct tag (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>8537</td>
<td>90.50%</td>
</tr>
<tr>
<td>40</td>
<td>8237</td>
<td>90.83%</td>
</tr>
<tr>
<td>50</td>
<td>8015</td>
<td>91.08%</td>
</tr>
<tr>
<td>60</td>
<td>8027</td>
<td>91.07%</td>
</tr>
<tr>
<td>70</td>
<td>8106</td>
<td>90.98%</td>
</tr>
<tr>
<td>80</td>
<td>8041</td>
<td>91.05%</td>
</tr>
<tr>
<td>90</td>
<td>8046</td>
<td>91.04%</td>
</tr>
</tbody>
</table>

5.2. Experiment two

In order to test our interpolated probabilistic model, the experiment is carried out by using the same set of training data that have the same size. Different from previous experiment, the model parameters are searched by using the previously described technique based on genetic algorithm. In our finding, one of the possible optimal parameters found by the algorithm is $\lambda_1 = 9.474e^{-1}$, $\lambda_2 = 5.0e^{-6}$, $\lambda_3 = 1.68e^{-3}$, $\lambda_4 = 2.353e^{-3}$ and $\lambda_5 = 7.0e^{-6}$. We repeated the experiment with different values of N, for each of which we indicate the number and percentage of correctly tagged words. Results are indicated in Table 3, and the chart representation of the results of experiments one and two is shown in Figure 1. From the result of the second experiment, we found that there is no great improvement as expected as that of in the first experiment where the quality of the tagging improves as the size of the training data increases. But compare with the first model, the optimized model reduces the error rate of 4.5%.

6. Conclusion

The results presented in this paper show that interpolating the features statistics into the word probabilities can efficiency guessing the correct tags for unknown words. Compare with other models, the lexical morphemes incorporated in our tagging system are acquired automatically from the training data other than manually constructed. Thus, the tagging accuracy of 91.08% is comparable for the results for Czech [1] and Swedish [2] tagging models. Another remarkable result shows that estimating the interpolation parameters of the model by using the genetic algorithm proves to be an alternative efficient technique to find the possible optimal values for the model coefficients.

Reference


