# A POS-Tagger Generator for Unknown Languages

**Nuno C. Marques** y **Gabriel Pereira Lopes**
**Universidade Aberta**
**nmm@univ-ab.pt**

**Centria, DI-FCT/UNL**
**{nmm,gpl}@di.fct.unl.pt**

**Abstract** It is current belief that POS-taggers need huge amounts of hand tagged text for training (in the order of $10^5$ pretagged words). In this paper we show how to generate POS-taggers trained with no more than $10^4$ hand tagged words. These taggers achieve precision results that are as good as the best performant state-of-the-art POS-taggers. We overcome the huge training corpus problem by carefully combining a large lexicon with an efficient neural tagger. Experimental results are presented and discussed for the Susanne Corpus and three different Portuguese corpora. 96% precision rates are obtained when unknown words occur in the test set.

## 1 *Introduction*

We wanted a generic approach to tagging. This approach should be totally independent of the language or text genre we are dealing with. We should also answer questions such as what is the best way to start working when we have only a large untagged corpus available? Or, what is the minimal amount of work required from a linguist? Our goal is to be able to work in any language or text genre. Our system should be tuneable to any imaginary unknown Language with only a minimum amount of work. This way we should be able to use any tag set on any language, sublanguage or text genre we need. Moreover, our goal should be to tag with any tag set that is needed for any specific application. We should support either syntactically oriented or more semantically oriented tag sets.

The problem of the lack of training text is a universal one even for English: The huge amounts of pre-tagged texts available are only suitable for some domains. As a matter of fact taggers trained with the existing hand tagged corpora perform quite poorly in specific application areas. Moreover, the tag set to use is highly dependent on the kind of problem and text one wants to study. This is the reason why we think that the colossal efforts done either for hand tagging huge amounts of text ([MSM93]) or for hand building (or tuning) complex rule systems ([SV97]) are only of limited use. Since [Mer94], it has also been clear that unsupervised training methods for tagging (such as the one proposed in [CKPS92] or [Bri95]) derive their good performance from lexical information (word endings) and need background information, supplied to the tagger by a hand defined set of initial contextual tagging rules. In practice these systems are just hand built rule base taggers enriched with probability estimates collected from untagged text. Although these probability estimation improves slightly the results of pure rules, these taggers still need a huge amount of hand work on rule building: they are unsuitable for our generic approach to linguistics.

Our first experiments, using HMM models ([VMLV95]), were discouraging. We needed better learning devices. That's why we started using neuronal networks([ML96b]) and got promising results. The use of neural networks for tagging should be similar to other probability estimation methods also applied to tagging, such as the maximum entropy tagger ([Rat98]), having the advantage of allowing a clear and richer modulation of context. We will present some results showing that when we train our tagger above the 10 000 tagged words limit, any more extra tagged words are only improving lexicon quality[ML01]. Based on this we will show that, by using a bootstraping approach, lexica can be acquired with an effort smaller to the one required for tagging a 100 000 word corpora. Results will show that this technique can successfully be used for overcoming problems posed by huge training corpora dependency, language dependency and text genre dependency.

## 2 Modeling Lexical Contextual with Neural Networks

In [Mar00], [ML01] we have described a neural model for POS-tagging. This model is based on two assumptions. The first assumption is that any word can be represented by a probability vector $\overrightarrow{amb_w}$. We named this vector as the lexical probability vector. The lexical probability vector appears in a dictionary and contains the probabilities of all the tags for a given word:

$$\overrightarrow{amb_w} = [p(tag_1|w), ..., p(tag_N|w)]^T$$

where $w$ is the word under consideration and the set of tags used for tagging the corpus is $tag_1, ..., tag_N$. The second assumption is that the part-of-speech information could be unambiguously extracted from sequences of three words (trigrams). Since syntactical information tends to appear in a local context, this approach is usual in the part-of-speech tagging literature [Mer94]. In [Mar00] we justify why these two assumptions tend to be closer to reality than the ones presented in standard POS-tagging literature. The basic reason for this is that a trigram of lexical vectors can enclose more information than the one supplied by ambiguity classes ([CKPS92]) or by the previous state on a HMM tagger ([Mer94]).

The neural tagging model was implemented using a simple feed-forward neural net [Hay94] using only input and output units (we tested several networks in [ML96b], and the simplest network was the one with best performance). The basic idea is to associate each neural unit with a part-of-speech tag $t_i$: Three vectors of input neurons represent the word trigram $w_{i-1}, w_i, w_{i+1}$. These three sets directly receive the values taken from lexical vectors $\overrightarrow{amb_{i-1}}, \overrightarrow{amb_i}, \overrightarrow{amb_{i+1}}$. In order to determine these vectors we use an internal lexicon. We based ourselves on MLE estimators extracted from the training corpus to calculate these values. After counting the frequency of pairs (word, tag), we build the lexicon by estimating $\overrightarrow{amb_w}$ for each tag. The lexical vector is calculated by:

$$\overrightarrow{amb_w}^* = [\tfrac{freq(tag_1,word)}{freq(word)}, ..., \tfrac{freq(tag_N)}{freq(word)}]^T.$$

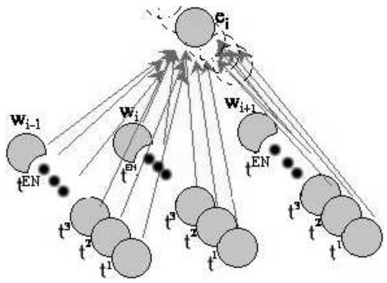

Figure 1: Diagram of the neuron used to identify a given POS tag

We will call this lexicon the internal tagging lexicon.

For training purposes the output units are assigned the values 1 or 0, according to the part-of-speech category they were tagged in the corpus: 1 if the neuron represents the tag assigned to the word and 0 otherwise. The network was trained first using the standard backpropagation algorithm and then momentum backpropagation algorithm. The standard backpropagation was used with $\eta = 1.0$. The momentum backpropagation algorithm was used with $\eta = 1.0$, $\mu = 0.7$ and $c = 0.1$ [Uni94].

Evaluation/tagging is similar: input values are acquired in the same way as during training. Then we propagate the input through the neural network and select the output unit with the larger activation value. We tag the word with the tag that is associated with the selected neuron.

We usually split our tagged corpus into a training corpus (used by the network training algorithm), an evaluation corpus (with 3 sentences, approximately 100 words[1]) and a test corpus (also with approximately 100 words). As usual in neural network training, to avoid over-fitting, we used the evaluation corpus to determine when we should stop the training process[2]. The test corpus is solely used to compute error rate.

[1] We have used more than the three sentences for test/evaluation, but results on the ten different runs showed that the use of just three sentences led to stable values

[2] That is, we minimized the learning error on another corpus different from the training corpus

## 3 The Influence of more tagged text: Results on Susanne Corpus

In [ML01], we have used the SUSANNE corpus[3] to separately measure the influence of contextual and lexical information in tagging precision. The Susanne corpus contains a total of 142524 tagged words (divided by 4200 sentences[4]). Without loss of generality, we have remapped the 426 tags presented in the original SUSANNE tag set into a smaller tag set of 37 POS tags. The 37 POS set represents a fairly standard tag set in tagging literature [MSM93][5]. This has also been done in order to increase the number of occurrences of each tag (some of the original tags occurred only once in corpus). For instance distinct unambiguous tags, such as tags MCn and MCr, denoting an Arabic numeral or a Roman numeral have been joined into the same POS class, numeral.

For evaluating the importance of the size of corpus into global precision we have selected several subsets of the corpus. Each subset had a different size (i.e. different percentages from the original corpus). At each subset we have used two distinct methods for building the internal tagging lexicon. The normal method used only the text available in the current subset for building the internal tagging lexicon. We used 10-fold cross-validation to measure the precision results presented in line TrigNorm of figure 2. In the other method we used the entire text in SUSANNE corpus, excepting the one selected for testing (that is our text was always unseen text, having unknown words) for building the internal tagging lexicon. As a result we have also measured how the tagger behaves if we use a not so good context (the same ones that were used for line TrigNorm) and a very good lexicon (that is a lexicon extracted from approximately 130 000 words, systematically excluding the test set). These results are presented under line TrigFull in figure 2.

Baseline precision was also calculated by computing unigram precision (measuring how far can we go without using any word context). The correspondent lines in figure 2 are *UnigNorm* (for the normal internal lexicon) and UnigFull (for the lexicon extracted from approximately 130 000 tagged words).

The analysis of graphic gives rise to an outstanding conclusion for most of the tagging research community. When using a sufficiently good statistical estimator, trigram contextual information can be extracted from as little as 5 700 words (92.6% ± 0.5 precision). Moreover top performance results (93.6% ± 0.5%) are acquired with only 22803 tagged words (that is a rather conservative engineered estimate since we achieve 93.2% ± 0.5 precision with only 11402 tagged words).

This conclusion justified the need of better dictionaries for tagging. Next experiment shows how this results can be used in a real life tagging problem.

## 4 Tagging Portuguese Text: Using the POLARIS lexicon

We have also applied this tagger generator to Portuguese text. When we started our work in Portuguese there wasn't tagged corpora at all to train our system (this work started 7 years ago with lexical development for Portuguese and at that time there wasn't not even an untagged Portuguese corpus). So we used a very small hand tagged corpus with only $5,000$ words. With a so small tagged corpus, the internal tagging lexicons that could be learned from text were also very small and incomplete. This fact resulted in a huge number of unknown words [ML96a] when tagging new text. Of course if we used the closed lexicon assumption we could achieve correction rates over 97%. But when, in a more realistic approach, we used an open lexicon our system's accuracy dropped to a modest 88% precision.

So we started working on using a general lexicon, the POLARIS system lexicon ([LMR94]). This system has a lexical database with more than $100,000$ base word forms, together with morphological inflection rules, as well as word derivation (suffixation

[3] The SUSANNE Corpus is a freely available, English annotated subset of the Brown corpus (ftp://ota.ox.ac.uk/pub/ota/public/susanne). This corpus contains a total of 4200 sentences, or 142524 tagged words and is supplied by the University of Sussex.

[4] We have measured this by counting the number of end-of-sentence marks that appeared in the corpus.

[5] Since very few tagging efforts are made public, no "standard tag set" for comparison purposes has yet emerged. Probably the problem is that every human tagger – we included – has her/his particular view of what tags she/he should/ *has to* use in its work. Despite this we have tried to adapt Susanne tags to what is fairly common in literature.

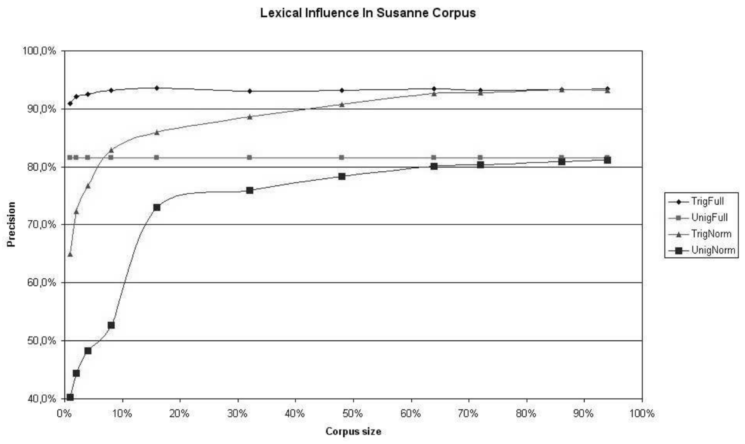

Figure 2: Learning speed with a good dictionary and different sizes of corpora. Each dot represents a subset of the corpus. Unigram lines represent results without any context and trigram lines represent results with context. The -norm suffix indicates that the internal tagging lexicon was extracted only from that subset of corpus. -full suffix indicates that the full corpus was used for building the internal tagging lexicon.

and prefixation) rules and irregular word formation rules ([ML94]). By using the POLARIS morphological engine and morphological rules our system is capable of recognizing more than $10^6$ words.

Unfortunately the POLARIS lexicon is a dictionary-extracted lexicon. And besides some inevitable noise, it only contains the list of possible parts-of-speech for each word. No tag probabilities are supplied. The main problem was then how to convert this ambiguity tag class into a lexical probability vector. The chosen approach solved the problem by using the following intuitive rules:

- **DictionaryRule**: If the word's vector is known in the internal tagging lexicon (the one extracted during training) then use it.
- **PBWRule**[6]: If a word has the same base word form as one whose probability vector is known in the internal tagging lexicon then use that probability vector.
- **PolarisRule**: If the word doesn't have any related probability vector in the internal tagging lexicon then use POLARIS lexicon. Since Polaris only returns an ambiguity class, we have to use the training corpus to build the ambiguity class probability vector[7].
- **UnknownRule**: If the word is unknown in both lexicons return a default unknown word probability vector, also measured from corpus.

The default ambiguity class probability vector is calculated based on the words that occur only once in the training corpus and have the given ambiguity class (ambiguity classes were first suggested in [CKPS92]). The unknown words probability class is based on the words that are not on the POLARIS lexicon (mainly proper names), and occur only once in the training Corpus (The reason behind estimating the probabilities of the unknown words based solely on the probabilities of the words that occur only once in the training corpus is justified in [BS96]. According to the studies performed by these authors in English, the probability of a given feature of the unseen words is best estimated based on the probability of the words that occur only once in the corpus, the *hapax legomena*).

## 5 *Portuguese Precision Results*

We have available two distinct Portuguese corpora: The Lusa corpus with internal news bulletins from the Portuguese news agency and the PGR corpus related to case-law texts. Since Lusa corpus is composed by news bulletins internally sent between the media, these bulletins sometimes have misspelled

[6] From Polaris Base Word.

[7] After using Polaris lexicon to convert each word to its ambiguity class, we estimate a dictionary of ambiguity classes by the same process we used to build the internal tagging lexicon

| | Dictionary | PBW | Polaris | None |
|---|---|---|---|---|
| Lusa | $1.7938 \pm 9 \times 10^{-4}$ | $1.18 \pm 2 \times 10^{-2}$ | $6.58 \pm 10^{-2}$ | $10.821 \pm 8 \times 10^{-3}$ |
| Lusa(+PGR) | $1.956 \pm 10^{-3}$ | $1.14 \pm 10^{-2}$ | $10.62 \pm 2 \times 10^{-2}$ | $18.857 \pm 6 \times 10^{-3}$ |
| PGR | $1.6683 \pm 4 \times 10^{-4}$ | $1.17 \pm 10^{-2}$ | $9.00 \pm 2 \times 10^{-2}$ | $18.09 \pm 10^{-2}$ |

| | Total |
|---|---|
| Lusa | $2.748 \pm 2 \times 10^{-3}$ |
| Lusa(+PGR) | $3.627 \pm 3 \times 10^{-3}$ |
| PGR | $2.316 \pm 10^{-3}$ |

Table 1: Medium ambiguity for each rule.

| | Dictionary | PBW | Polaris | None |
|---|---|---|---|---|
| Lusa | 83.4% | 1.5% | 11.2% | 3.9% |
| Lusa(+PGR) | 83.6% | 2.5% | 9.9% | 4% |
| PGR | 92.6% | 1.3% | 1.7% | 4.4% |

Table 2: Relative weight for each rule.

words, namely missing diacritics. Compared with PGR corpus the Lusa corpus is a very noisy one. The Lusa corpus has 5 400 hand tagged words (with $2.748 \pm 2 \times 10^{-3}$ tags per word) and the PGR corpus has 18675 hand corrected tagged words (with $2.316 \pm 10^{-3}$ tags per word). If we join both corpora we will acquire a more generic corpus (that we will call Lusa(+PGR) from now on) with $3.627 \pm 3 \times 10^{-3}$ tags per word. In table 2, we have also counted how many times each of previous section rules were applied in the test corpus, when we use the training corpus to build the internal tagging dictionary. For example, in the Lusa texts 83.4% of words were present in the internal tagging dictionary. 1.5% of words were morphologically related to the words from the training corpus; 11.2% only occur in the POLARIS lexicon; 3.9% words were unknown.

In table 3 we present global precision values for our three corpora split accordingly with previous section rules. In last column we present the global average precision. As previously we have made ten runs with different training and test data over the same corpus, and established confidence intervals using a t-student distribution with $\alpha = 0.05$. Table 4 presents the same data without using any contextual information (unigram tagger).

Once again we see the importance of using a large lexicon. The known word precision in corpus is good (95.5% for Lusa and 97.3% for PGR[8]) but those numbers are worst when the word is unknown in the training corpus. However, when we use contextual information, our external dictionary does indeed provide some help: pure unknown words (the unknown column) are the most difficult ones, but (as you can see in table 2), only approximately 4% of the words fall into that case. All other words were covered by one of the POLARIS rules. There, despite some precision degradation, results are significantly better. Recall also that, although PBWRule may seem a little odd in some cases, on average it seems to give best results than not using it at all.

Of course the importance of using an external dictionary rises with the decrease in the size of the training corpus. In Lusa 12.7% of all the words (that is 1.5% + 11.2%, according to columns PBW and POLARIS in table 2) are treated by using POLARIS external dictionary, while in PGR (the 18675 word corpus) only 3.0% (1.3% + 1.7% from table 2) of words are treated.

We should also stress that merging different corpus is usually a bad idea. In line Lusa(+PGR) from table 3, we see a systematic degradation of tagging performance, due to the increased entropy effect of joining several distinct corpora. But, from table 2, we see that this entropy increase is not supported by a significant reduction in the number of unknown words. Neither joining corpora does reduce external dictionary dependence (most of rare words in a corpus are characteristic of that genre and kind of text, and so probably non existent in the other corpus). In the end we have increased hand tagged training corpus size but decreased global precision.

[8] Since Lusa corpus is a noisiest one, as it should be expected, precision degradation in Lusa is higher than in PGR.

|  | Dictionary | PBW | Polaris | None | Total |
|---|---|---|---|---|---|
| Lusa | 95.5 ± 0.2% | 78 ± 4% | 77 ± 2% | 61 ± 3% | 91.9 ± 0.3% |
| Lusa(+PGR) | 94.7 ± 0.3% | 90 ± 3% | 73 ± 3% | 57 ± 4% | 91.0 ± 0.3% |
| PGR | 97.30 ± 0.09% | 95.1 ± 0.8% | 91.1 ± 0.7% | 56 ± 2% | 96.3 ± 0.2% |

Table 3: Tagging precision in Portuguese.

|  | Dictionary | PBW | Polaris | None | Total |
|---|---|---|---|---|---|
| Lusa | 84.5 ± 0.5% | 36 ± 4% | 77 ± 2% | 67 ± 3% | 81.1 ± 0.5% |
| Lusa(+PGR) | 84.0 ± 0.5% | 40 ± 3% | 70 ± 1% | 73 ± 4% | 81.1 ± 0.5% |
| PGR | 86.4 ± 0.2% | 34 ± 2% | 89.5 ± 0.8% | 30 ± 2% | 84.9 ± 0.2% |

Table 4: Unigram precision.

## 6 *The Unknown Language Lexicon*

We observed that when we start with a large lexicon, a training corpus with only 10 000 words contains most of contextual information needed to estimate tagging probabilities for the neural network.

Until now we had available a huge lexicon, however sometimes, this is not possible for a generic unknown language, such as the one we aim at treating. Our main principle to solve these problems is that when there is no dictionary, it's a lot easier to build a tagging dictionary than to repeatedly tag the same words in a corpus. Indeed this methodology has already been applied successfully to the tagging of a corpus of medieval Portuguese texts[9]. Medieval Portuguese has two main problems illustrative of the importance of our tagger capabilities: non-normalized orthography and a changing lexicon (texts rage from XIII century to XIIX century).

When there is no high-coverage lexicon available, the following bootstrapping approach should be used: Let us start with a minimal tagged corpus (for instance the 5000 tagged words from LUSA corpus). After that, due to the Zipf distribution of words in text, only very few words are unknown. According with table 2 we have 20% of unknown words in the 5000 word Lusa Corpus. That way to build a dictionary covering a corpus of $20,000$ words, we will have to tag more $3,000$ unknown words. Once again by table 2, we are in the conditions of the PGR corpus: 8% ambiguity. If we go directly to a dictionary equivalent to the one extracted from a 70 000 Corpus, we should only need to tag more 4 000 unknown words. That is, only 12 000 words have been tagged by the linguist, for achieving the conditions we had to make the SUSANNE corpus experiment.

Indeed, this was the case on Medieval Portuguese texts. When we started our work, there was neither a starting tagged text, nor a tag set to start from. So the work was started by hand tagging 10 000 words. This work supported the initial analysis of the text and was also needed to help linguists to define what where the best tag set for the corpus. Based on this small corpus we trained a neural network and built the first Medieval Portuguese lexicon. Then we have taken a sample of about 50,000 words. From this sample, we extracted all the unknown word forms (that is words not present in the lexicon extracted from the 10 000 word corpus). This resulted in a list of approximately 11,000 words that were also hand tagged and used to expand our initial lexicon.

Given the rather good results acquired, automatic tagging is still being applied to various texts on different centuries. Accordingly to our linguistic colleges the use of the neuronal tagger did significantly reduce the manual work to be done in order to collect data for different kinds of linguistic studies, namely: lexical, terminological and grammatical, and studies in other domains such as History, Culture and Literature.

Another way to avoid using an external lexicon could be achieved by using word endings. Several authors have already used this approach (for example [CKPS92] or [Sch94]). In fact the analyses of the errors introduced by unknown words is by itself a field of research, see for example [Fra96]). By table 3 we see that there is still some room for future work: with an optimistic increase of tagging on unknown words up to 90%, we could get a final

[9] Project JNIC-FCSH/C/LIN/931/95.

precision increase up to 4% × 30% = 1.2%, solely by considering unknown words. However this approach is particularly interesting if we add this information to the ambiguity vector. Indeed, in future work we intend to evaluate the effect of adding word endings to our system. The neuronal network is particularly well suited for this type of changes. For instance, we could add several neurons to represent the main word endings in a given language.

## 7 *Conclusions*

Probably the most significant conclusion of this work is that when using a neuronal network classifier for tagging, poor lexicons, instead of large corpora, are the major cause for tagging errors.

In our tagging system we have used neural networks to represent our tagging probabilities. Although this is just one from many possible parameter estimation techniques, this technique was particularly well suited for an efficient learning from our data. Also neural networks have the advantage of allowing richer models for context, namely if we use the word endings proposed in previous section. The presented results show that we have achieved our goal of using only minimal linguistic work for POS-Tagging. With this tagger it is no longer needed to be the human user to provide a so specific, interdependent and text dependent set of rules as the ones needed to disambiguate Part-of-speech information. Moreover, linguists now do the interesting and productive work, leaving the repetitive work for the computer.

Finally let us stress the importance of being able to train a tagger with such a small size of tagged corpora. Even in a studied language such as English, when we start treating new types and genres of text, not only it is frequently required a new tagger, but also it is frequently needed a completely different tag set. Indeed, standard tag sets are useful for comparing works (as we have made here), and do have some interest for ulterior parsing, but that doesn't exclude all the possibilities. For instance, more semantically inspired tag sets are often more suited for machine translation. Since it isn't the human user to provide a so specific, interdependent and text dependent set of rules as the ones needed to disambiguate Part-of-speech information, particular knowledge of the domain is not really needed. By using a learning-efficient tagger as the one described, we are able to change the tag set or even the language easily while continuing to achieve very good tagging performances.

## 8 *Acknowledgments*

The authors would like to thank FCSH/UNL historical linguistic research people, a group headed by Prof. M. Francisca Xavier: the results in the Medieval Portuguese would not be possible without our excellent cooperation.

## *References*

[Bri95] Eric Brill. Unsupervised learning of disambiguation rules for part of speech tagging. In *Proceedings of the Very Large Corpora Workshop*, 1995.

[BS96] H. Baayen and Richard Sproat. Estimating lexical priors for low-frequency morphologically ambiguous forms. *Computational Linguistics*, 22(2):155–166, 1996.

[CKPS92] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Proceedings of the third ACL Conference on Applied Natural Language Processing*, pages 133 – 140, Trento, Italy, 1992.

[Elw94] David Elworthy. Does baum-welch re-estimation help taggers. In *Proceedings of the Fourth Conference on Applied Natural Language Processing*, 1994.

[Fra96] Alexander Franz. *Automatic Ambiguity Resolution in Natural Language Processing*, volume 1171 of *LNAI Series*. Springer, 1996.

[Hay94] Simon Haykin. *Neural Networks: A comprehensive Foundation.* Macmillan College Publishing Company, Inc., 1994.

[HD00] V. Hoste and W. Daelemans. Comparing bagging and boosting for natural language processing tasks: a typically approach. In Bernard Lang, editor, *BENELEARN 2000: proceedings of the Tenth Belgian-Dutch Conference* *on Machine Learning*, pages 101–109, Tilburg University, 2000, 2000.

[LMR94] José Gabriel Lopes, Nuno Cavalheiro Marques, and Vitor Ramos Rocio. Polaris, a POrtuguese Lexicon Acquisition and Retrieval Interactive System. In *Proceedings of the conference on Pratical Applications of PROLOG* , 1994.

[Mar00] Nuno Cavalheiro Marques. *Uma Metodologia Estatística para a Modelação da Subcategorização Verbal.* PhD thesis , Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, 2000.

[Mer94] Bernard Merialdo. Tagging english text with a probabilistic model. *Computacional Linguistics*, 20(2):155–171, 1994.

[ML94] Nuno Cavalheiro Marques and José Gabriel Lopes. Reconhecimento de neologismos. In *Proceedings of the IBERAMIA Conference* , 1994.

[ML96a] Nuno C. Marques and José Gabriel Lopes. A neural network approach to part-of-speech tagging. In *Proceedings of the Second Workshop on Computational Processing of Written and Spoken Portuguese* , pages 1–9, Curitiba, Brazil, October 21-22 1996.

[ML96b] Nuno C. Marques and José Gabriel Lopes. Using neural networks for portuguese part-of-speech tagging. In *Proceedings of the Fifth International Conference on Cognitive Science and Natural Language Processing* , Dublin City University, Ireland , September 2-5 1996.

[ML97] Nuno Cavalheiro Marques and José Gabriel Lopes. Neural networks, part-of-speech tagging and lexicon. Technical report , Departamento de Informática, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Febuary 1997.

[ML01] Nuno Cavalheiro Marques and José Gabriel Lopes. Tagging With Small Training Corpora. In *Advances in Intelligent Data Analysis, Fourth International Symposium, IDA 01.* To be published in Springer's LNCS Series. Cascais, September 2001. Portugal.

[MSM93] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computacional Linguistics*, 19(2):313–329, 1993.

[Rat98] Adwait Ratnaparkhi. *Maximum Entropy Models for Natural Language Ambiguity Resolution.* PhD thesis, University of Pennsylvania, 1998.

[Sch94] Helmut Schmid. Part-of-speech tagging with neural networks. In *Proceedings of the International Conference on Computational Linguistics*, Kyoto, Japan, 1994.

[SV97] Christer Samuelsson and Atro Voutilainen. Tagging french - comparing a statistical and a constraint-based method. In *Proceedings of the European Chapter of the Annual Meeting of ACL*, 1997.

[Uni94] University of Stuttgard – Institute for Parallel and Distributed High Performance Systems (IPVR). *User Manual of the Stuttgard Neural Network Simulator*, 1994. Report No. 3//94.

[VMLV95] Aline Villavicencio, Nuno C. Marques, José Gabriel Lopes, and Fábio Villavicencio. Part-of-speech tagging for portuguese texts. In Jacques Wainer and Ariadne Carvalho, editors, *Advances in Artificial Intelligence: Proceedings of the XII Brazilian Symposium on Artificial Intelligence*, Lecture Notes in Artificial Intelligence 991, pages 323–332, Campinas, October 11-13 1995. Springer Verlag. (10 pages).