Using Neural Nets for Portuguese Part-of-Speech Tagging[*†]

Nuno C. Marques[‡] **Gabriel Pereira Lopes**
(nmm@di.fct.unl.pt) **(gpl@di.fct.unl.pt)**
Universidade Nova de Lisboa - Faculdade de Ciências e Tecnologia
Departamento de Informática
2825 Monte da Caparica
Portugal

Abstract

We will describe the use of neural networks to perform part-of-speech (POS) disambiguation of textual *corpora*. Available part-of-speech taggers need huge amounts of hand tagged text, but for Portuguese there is no such *corpora* available. In this paper we propose a neural network that, apparently, is capable of overcoming the huge training corpus problem. Distinct network topologies are applied to the problem of learning the parameters of a part-of-speech tagger from a very small Portuguese training corpus. The experiments carried out are discussed. The results obtained point to a correction rate above the 96% using a hand tagged training *corpus* with approximately 15,000 words.

1 Introduction

The application potential of textual *corpora* increases, when the *corpora* is annotated. The first logical level of annotation is usually part-of-speech tagging. At an upper level the text is no longer seen as a mere sequence of strings and is taken as a sequence of linguistic entities with some meaning. The annotated text can then be used either for further introdution of new types of annotations (usually by means of syntactic parsing (Marcus et al., 1993), (de Marcken, 1990)), or for directly (or indirectly) collecting statistics for different kinds of applications. Working at the word tagging level enabled various kinds of applications namely speech synthesis (Church and Mercer, 1993), clustering (Fernando Pereira, 1993) computational lexicography (Manning, 1993), and many others.

The success of this kind of technique is certainly due to its intrinsic capability for assigning a sequence of part-of-speech tags to any sequence of words with high precision by using quite modest computer resources. Despite this, part-of-speech taggers are not yet as fully available as they should, especially when we are working with languages other than English. The main problem with currently available part-of-speech taggers is the lack of hand tagged *corpora* necessary for training the taggers. And almost every tagger needs huge amounts of hand tagged text[1]. Due to recent tagging efforts this is no longer a problem in languages such as English or French, but for Portuguese, that problem had not yet been solved.

In this paper we propose and discuss a method for overcoming the problem of small and scarce hand tagged *corpus* , having no need of any complex algorithms. For that it was used the Stuttgart Neural Network Simulator (Uni, 1994) [2], standard Unix tools (Church, 1994) and a small hand tagged corpus: the *Radio-Bras Corpus*[3]. Simple gawk[4] scripts have been implemented in order to achieve the results shown in this paper[5].

In this paper we will address the problem of part-of-speech tagging in order to obtain an answer for

[*] Published in Proceedings of the Fifth International Conference on Cognitive Science and Natural Language Processing, Dublin City University, Ireland, September 2-5 1996.

[†] Work partially supported by project Corpus funded by JNICT under contract number PLUS/C/LIN/805/93

[1] It is usually argued that the Markovian taggers can handle text without requiring annotated training text. However this is not the case, as it has been independently shown by Merialdo (Merialdo, 1994) and Elworthy (Elworthy, 1994).

[2] The SNNS package is fully available at the University of Stuttgart and can be obtained via anonymous ftp from host: *ftp.informatik.uni-stuttgart.de (129.69.211.2)* in the subdirectory */pub/SNNS*

[3] *http://www.mre.gov.br/sinopse.htm.* This bulletin is distributed in Internet through electronic mail messages by the science and technology editor from "Agência Brasil".

[4] Gnu AWK

[5] Please have a look at *http://www-ia.di.fct.unl.pt/~nmm/Software* for more information.

the following two questions. How large must a hand tagged *corpus* be for enabling the achievement of acceptable results (over 95% correctness on tagging new text)? How complex must neural net based tagger be?

For the first problem we have found that a training corpus with 3,362 hand tagged words enables precision results over 95% . For the second problem we have expermented three types of neural nets and the results obtained are reported in the next few sections.

Schmid (Schmid, 1994), presents a neural network approach capable of part-of-speech tagging. Although a 96.22% performance is reported, the neural network model presented there is more complex than the ones presented here and a corpus of 100,000 words was used to train the neural net. Schmid argues the neural net based tagging achieve better results when the size of the training corpus is small. In this paper we argue in the same direction. (Nakamura et al., 1990) has also done some related work, but their goal was the prediction of the next word to appear in the input text. It was not intended for *corpus* tagging (a precision of 86.9% was obtained). Other work on this area (Elenius, 1990) is also referred in (Dermatas and Kokkinakis, 1995), but we have not got it, yet. No other work we had access to addresses the kind of problem we are interested in.

2 System Description

2.1 Neural Network Models

The first topology we had used for solving the part-of-speech tagging problem, was a simple feed-forward neural net (Rumelhart and McClelland, 1986) (named 1L Net) having only input and output units. The input units were divided into two sets of context units. Each output unit represents one of the tags. A one-to-one relation is established between each value in the lexical probability vector (acquired from the lexicon) and each input unit in each set. A simple bigram model was implemented using these two sets: the first set receives the probability vector of the word we want to classify and the second set the vector of the next word in the sentence (the context word). The network is fully connected: each input unit is connected to all output units.

In a second topology (named HL Net) we tried to increase the neural network discriminative power. A layer of hidden neurons was added: all input units are connected to all hidden units. The hidden units are now the only connection to the output units.

Finally, in the third topology, an Elman neural network (Elman, 1990) was used (this topology was named Elman Net). We replaced the context word's input layer of the second network by a recurrent layer: each unit was connected to itself by a link of weight 1 (identity link) and to all hidden units. Each hidden unit was connected to one context unit using an identity link. The main goal to achieve with this topology is to supply the net with a short-term memory, so that the context of the last word seen can be parameterized by the learning process.

2.2 Word Coding

Any tagging problem can be seen as the maximization of the joint probability $P(W,T)$, where W stands for the input string of words and T for the ordered set of tags.

Each distinct word (w) in the *corpus* has a corresponding entry in the lexicon containing an estimation of the probability of w having a tag t:

$$\hat{k}(w|t) = \frac{freq(w,t)}{freq(t)} \tag{1}$$

So, each entry in our lexicon is associated with a m dimensional vector (one $\hat{k}(w|t)$ for each tag t)[6]. The basic idea in this approach was to supply the network with the lexical probabilities. Contextual probabilities are left for being learned from the training *corpus* . According to (Merialdo, 1994), the probability distribution $P(W,T)$, may be seen as:

$$p(W,T) = \prod_{i=1}^{n} p(w_i|w_1t_1\cdots w_{i-1}t_{i-1}t_i)p(t_i|w_1t_1\cdots w_{i-1}t_{i-1}) \tag{2}$$

The following hypothesis have been made for the proposed model (called a n-gram model or an HK model):

- The probability of a tag, given its past, depends only on the last n tags (in our work a bigram model ($n=1$) was assumed) :

$$p(t_i|w_1t_1\cdots w_{i-1}t_{i-1}) = h(t_i|t_{i-n}\cdots t_{i-1}) \tag{3}$$

  This distribution should be modelled by the network.

- The probability of a word, given its past, depends only on its tag:

$$p(w_i|w_1t_1\cdots w_{i-1}t_{i-1}t_i) = k(w_i|t_i) \tag{4}$$

[6] m is the number of tags used.

This value can be estimated using $\hat{k}(w|T)$ as we have seen in equation 1. $\hat{k}(w|T)$ is supplied by the lexicon.

In fact our neural nets can use more information than the one just described. The lexical probability vectors may be distinct for words that are represented by the same part-of-speech tag. The chosen representation for the words (associating a set of neurons with each word) reveals itself not only a very simple and elegant one but also a very expressive one.

Another characteristic of the chosen architecture is the fact that the classification is performed only locally and never at the sentence level. This has some relation with the maximum likelihood tagging that Merialdo refers in his article (Merialdo, 1994). The great majority of the statistical taggers (using Markov models) works at sentence level. This is done by performing a global optimization algorithm of the possible bigram or trigram tag sets used in a sentence level: the Viterbi tagging. Although this is linguistically more plausible, this approach has as a drawback the propagation of the tagging errors done at the start of a sentence through out the whole sentence. Conversely, the local approach used here, has the disadvantage of allowing quite implausible combinations of tags, unacceptable to models working at the sentence level. According to experimental evidence reported in (Merialdo, 1994), these effects do not seem to affect seriously the tagging performance.

2.3 Training and Evaluation

The three networks were trained using the standard back-propagation algorithm with and without momentum term. The training of the Elman Net is done without using the recurrent links. The recurrent links are only used after a full iteration of the learning algorithm, when we are updating the value of the contextual units. This way the standard back-propagation and momentum back-propagation can be applied.

All networks were trained using $\eta = 1.0$. When used, the momentum term, was set to $\mu = 0.7$ and the flat spot elimination parameter (c) was set to 0.1 . Training was performed until an acceptable convergence was found (usually between 5000 and 10000 iterations). For more information about the training algorithms please see (Uni, 1994).

Each unit is associated with a part-of-speech tag T_i. During the training phase the value $p(w_k|T_i)$, acquired from the lexical probability vector of each word w_k, supplied by the dictionary, is assigned to the associated set of input units.

For training purposes, the output units are assigned the value 1 or the value 0, according to the part-of-speech category they were tagged in the corpus: 1 if the neuron is related to the tag assigned to the word and 0 otherwise.

During the evaluation, the vectors from the lexicon entries that are addressed by the word we are tagging and by the word following it are respectively assigned to the first and the second training sets. After this we propagate the input units to the output units and select the tag related to the output unit with a larger value.

2.4 The Tagging Process

Using a set of tagged texts, it was possible to build a lexicon that contains, for each word its possible associated tags and its occurrence frequencies. A simple script was built returning for each entry in the lexicon its associated lexical probabilities vector (see section 2.2). Three training sets and four test sets pattern files[7] were randomly generated.

The generated pattern files were compatible with a general purpose neural network simulator: the *Stuttgart Neural Network Simulator* version 3.2 (Uni, 1994). Each training set was used for training the three tested models (see section 3). Each test file was then used to evaluate the trained tagger. This way the test sets were tagged and so the precision of the tagger could be found, simply by comparing the taggers output with the corpus tags.

In Figure 1 we illustrate the tagging process.

3 Results

As stated earlier, the *Radio-bras Corpus* was used to compare the used network topologies based on their tagging precision. 652 sentences from that *corpus* were hand tagged using 35 distinct tags. The resulting *corpus* has 19141 tagged words (Villavicencio et al., 1995), (Villavicencio, 1995) .

The acquired lexicon has $4,009$ entries. From these 93.60% are non ambiguous, 5.64% have two alternative tags, 0.67% three alternative tags, 3 entries (0.07% from the total) have 4 possible tags and finally 1 entry has 5 possible tags (0.02% from the total).

In order to build a training and a test sets, the main *corpus* was randomly split in three (non disjunct) training sets and in four disjunct test sets. After this, the training sets and three of the test sets were tagged using the dictionary extracted from the main corpus. The fourth test set was tagged with a lexicon extracted from the training set. Using this

[7] The file containing the set of activation patterns of all input and output units of a neural network

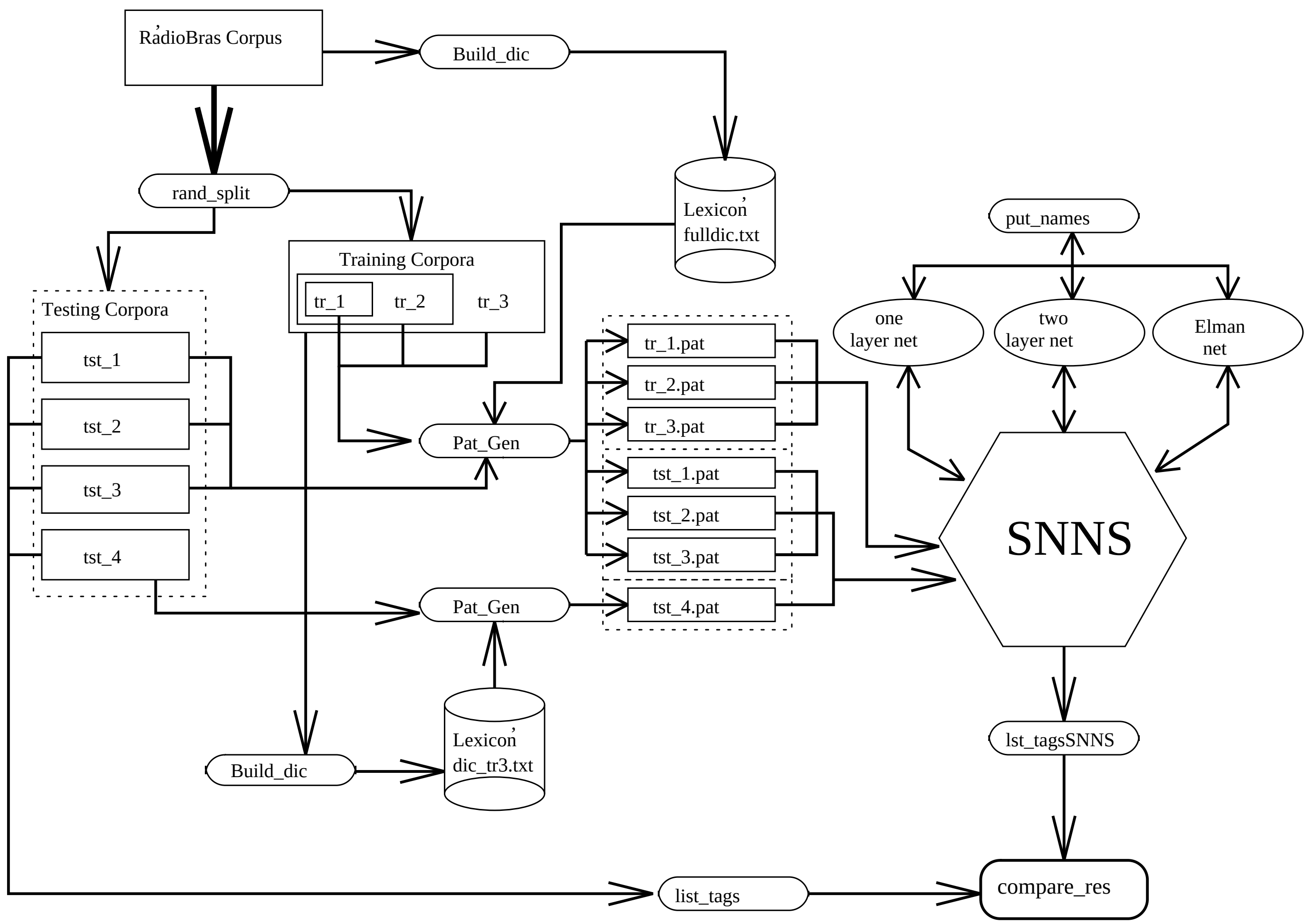

Figure 1: Diagram illustrating the work of the Part-of-Speech tagger

File	Type	#Sentences	#Tags	%Ambiguity					
				2 Tags	3 Tags	4 Tags	5 Tags	Unknown	Tot.
Corpus	Crp	652	19141	17.9%	7.7%	3.0%	0.3%	—	28.8%
tr_1	Trn	22	662	16.8%	6.8%	2.7%	0.2%	—	26.4%
tr_2	Trn	112	3362	17.8%	7.4%	3.2%	0.2%	—	28.6%
tr_3	Trn	538	15861	17.5%	7.7%	3.0%	0.2%	—	28.4%
tst1	Tst	33	945	18.8%	8.7%	3.2%	0.3%	—	31.0%
tst2	Tst	21	593	19.7%	6.4%	4.1%	0.3%	—	30.5%
tst3	Tst	25	691	21.0%	5.0%	2.8%	0.6%	—	29.4%
tst4	Tst	35	1051	19.5%	6.0%	3.5%	0.0%	12.9%	41.9%

Table 1: Data from the several files used to train and to test the neural net.

process we have created three test sets (tst1, tst2 e tst3) without unknown words[8] and one training set (tst4) with unknown words [9]. Table 1 shows the number of sentences, words and ambiguities of each of these sets. The column "type" has the value "Trn" for the training sets and "Tst" for the test sets.

Unfortunaly the huge training times didn't allow the use of better evaluation methods, such as obtaining randomly independent training and test sets from the hand tagged *corpora* (a number closed to 30 would work, but this would require a 4 month workload for training). Despite this, the precision variation among the several test sets is usually less than 1%. So we think that the results obtained are representative of the tagging accuracy.

In figure 2 we summarize the results obtained.

In line 1, we only used 120 iterations of the back-propagation algorithm, and, in line 2, 10000 iterations were used. The precision gain acquired by the performing more iterations decreases with the increase of the training *corpus* size.

The precision gain obtained by augmenting the size of the training corpus above a certain threshold (3200 words for 1L net) is rather limited.

These results become less visible when we increase the number of training iterations. We only get a 2% gain in precision when we have passed from training sets tr_2 to tr_3. The presented results seem to suggest that, at least in what refers to the 1L network, an acceptable precision can be acquired using only 3362 tagged words. Line 3 in this figure, although having less detail, confirm this pattern of growth for Elman net.

3.1 Results for the Used Models

In table 2, we compare the performance of the networks experimented for the training set tr_3.

Nets 1L and HL give the best precision results. Network 1L seems to be the one which gives better results even with set tst_4, where unknown words are present[10]. The Elman Network precision is significantly lower when tagging with a closed lexicon (tst_1, tst_2, tst_3), but seems to give average results when exposed to unknown words (in tst_4). Better results may be obtained if more iterations of the training algorithm are performed on this network.

From these results, it seems that, for the volume of data used, no significant advantage was acquired by increasing the network complexity. Despite this fact, we must say that our training data was very small. So, we still don't know if the use of more complex networks (such as HL or Elman nets), can become profitable when larger data sets are available.

The results presented seem to indicate that a neural network tagger can supply better results than other techniques, when the size of the training *corpus* is smaller than 20,000 words. However it is difficult to compare these results with the results obtained by Markovian methods, because we are working on a different language (Portuguese, not English), with different tag sets.

These results seem to be significantly better than the ones obtained in previous joint work (Villavicencio et al., 1995) or (Villavicencio, 1995). Using the same *corpus* and a Hidden Markov Based tagger (Villavicencio et al., 1995) and a general purpose dictionary, 84.5% of precision was reported. This number is considerably lower than the one presented in this work for network 1L *without no* general purpose dictionary. An 88.0% precision was obtained by (Villavicencio, 1995) using a closed dictionary (assuming that every word is known), the same *corpus* and a Hidden Markov tagger. This is also considerably lower than the results reported here.

3.2 Handling unknown words

Whenever the test set tst_4 was used the precisions droped to 87% accuracy. This is certainly due to nonexistence of unknown words in the training set and also to the way how unknown words were handled: whenever an unknown word occured in the text, a default probability vector was assigned to it. Ulterior results, based on the use of a standard dictionary to deal with unknown words, did reveal that the tagging precision can be raised up to 91.0%.

We hope to increase these values not only by giving the proper treatment to the Portuguese morphology[11], but also by using a standard lexicon for Portuguese language (Lopes et al., 1994), (Marques and Rocio, 1994) including Portuguese suffix and prefix information (Marques and Lopes, 1994).

[8] Slightly biased results may be acquired by using these test sets but they are the only way to test the usual closed lexicon assumption ((Villavicencio, 1995) or (Brill, 1995)).

[9] Remember that the dictionary used for tagging the test set tst4 was obtained from the training set. This means that probably in tst4 there are words that didn't show up in the training sets, hence the *unknown word*

[10] More iterations were performed with Elman and HL nets. However worst results were obtained, probably due to overtraining effects on these nets. This problem needs further research.

[11] Some ideas on how to improve the dictionary by using morphological information can be acquired in (Levinger et al., 1995)

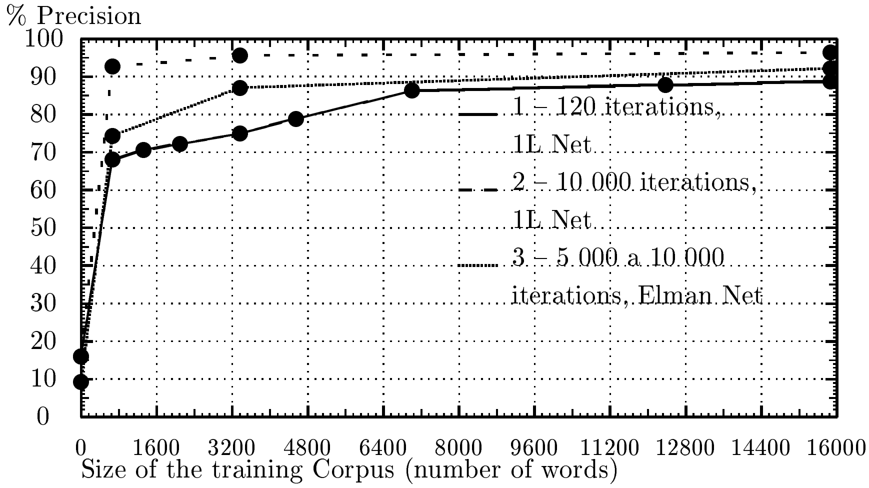

Figure 2: Variation of the precision with the training corpus size.

File	1L Net			HL Net		Elman Net	
	Iterations			Iterations		Iterations	
	4322	6550	10 000	2626	3021	4461	5165
tst_1	96.6 %	96.6 %	96.6 %	96.4 %	96.4 %	91.7 %	92.7 %
tst_2	95.1 %	95.4 %	95.8 %	95.8 %	95.8 %	90.9 %	91.5 %
tst_3	96.1 %	96.5 %	96.5 %	96.8 %	96.6 %	92.6 %	92.2 %
pond. aver.	96.0 %	96.2 %	96.4 %	96.4 %	96.3 %	91.8 %	92.2 %
tst_4	87.8 %	87.6 %	87.5 %	82.2 %	82.3 %	85.9 %	86.3 %

Table 2: Precision results obtained by the used Neural Networks for the training set tr_3

4 Conclusions

In this work we presented a novel neural network approach to the problem of part-of-speech tagging. The presented system has three main advantages over other neural net systems: its simplicity, its implementation (available through Internet) and its capacity to accurately tag new text needing only a very small hand tagged *corpus*. In (Schmid, 1994) a neural network approach is also presented. However his recurrent net, according to the presented results, needs a considerably larger hand tagged corpus to achieve the same tagging accuracy.

The performance obtained is comparable to the state-of-the-art part-of-speech taggers. It is true, that recent work addresses the problem of training a tagger using unanotated text (Brill, 1995), (Merialdo, 1994), (Merialdo,), (Elworthy, 1994) and (Cutting et al., 1992). However these approaches require both more complex methods than the one presented here and a pre-existing lexicon. In all these approaches the used lexicon is extracted from a pre-existing tagged corpus, and not from a general purpose dictionary. There are several problems with the extraction of a lexicon from a general purpose dictionary. To name just a few: the tag set used is normally much smaller than the one we intend to use; there is no way to find out which of the part-of-speech tags presented in an entry is more probable than the others; some entries are incomplete (don't list every possible use of that word (Manning, 1993)); and lastly (but probably the most significant) a machine readable dictionary is needed.

The main drawback of neural networks is the huge training times required for training a network. Training set tr_3 , although small when compared with other *corpora* used in experiments with Markov Models, has $15,870$ words. Each of these words is represented by a 35 dimension vector. Using our simpler net, each word implies the training of $35 \times 35 \times 2$ weights per word. So we need to perform $15,870 \times 35 \times 35 \times 2 = 38,881,500$ updates for each iteration. For the training *corpus* tr_3 the one layer *feed-forward* neural network took 4 days for training with $10,000$ iterations. Preliminary results with this method showed that, using the *momentum* back-propagation, may become a good solution for addressing this problem by allowing faster convergences.

The proposed approach, seems to be an appealing alternative to the use of unannotated trained text taggers when no text is available. The small tagged corpus that can be used in this approach, does not seem to be much of a problem: the best way to familiarize yourself with a *corpus* and tuning the intended tag set to the used *corpus* , is to tag some text. Essentially when you know that even with a text as little as 100 sentences (approximately $3,000$ words), the tagger can immediately start supplying very good results.

This technique seems to be well suited for a *bootstrapping* approach to the tagging problem. The results also seem to be at least as good as for the standard hidden Markov models presented in the literature for larger training sets. For hand tagged *corpora* with the size of the one available for Portuguese, the results show a much better performance for the neural network approach.

Finally, further enhancements to the system just presented have started to be implemented. Namely, we are starting to deal with composed words, locutionary phrases, contractions, numbers and some other textual information, that usually decreases tagging accuracy. This can be achieved by preprocessing the text before tagging. Another enhancement that can be done is related to trying to use unsupervised learning on new, untagged text, acting similarly to the Baum-Welch reestimation procedure ((Rabiner, 1989)).

5 Acknowledgments

We would like to acknowledge Aline Villavicencio and Fábio Villavicencio for helping in the results obtained with the Hidden Markov Model Tagger, suppling us with the tagged *corpus* and for the many fruitful discussions through electronic mail. We also would like to send our thanks to the people involved with the SNNS project for making their software freely available.

References

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

Kenneth W. Church and Robert L. Mercer. 1993. Introduction to the special issue on computational linguistics using large corpora. *Computacional Linguistics*, 19(1):1–25, March.

Kenneth W. Church. 1994. Unix for poets. Notes of a course from the European Summer School on Language and Speech Communication, Corpus Based Methods, July.

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

Carl G. de Marcken. 1990. Parsing the lob corpus. In *Proceedings of the 28th Annual Meeting of ACL*, pages 243–251.

Evangelos Dermatas and George Kokkinakis. 1995. Automatic stochastic tagging of natural language texts. *Computacional Linguistics*, 21(2):137–162.

K. Elenius. 1990. Comparing a connectionist and rule based model for assignment parts-of-speech. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pages 597 – 600.

J. L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14:179–211.

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

Lillian Lee Fernando Pereira, Naftali Tishby. 1993. Distributional clustering of english words. In *Proceedings of the 31st Annual Meeting of ACL.*

Moshe Levinger, Uzzi Ornan, and Alon Itai. 1995. Learning morpho-lexical probabilities from an untagged corpus with an application to hebrew. *Computacional Linguistics*, 21(3):382–404.

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

Cristopher Manning. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting of ACL*, pages 235–242.

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

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

Nuno Cavalheiro Marques and Vitor Ramos Rocio. 1994. Polaris, a POrtuguese Lexicon Acquisition d Retrieval Interactive System. Technical report, UNINOVA, Centro de Inteligência Artificial (CRIA).

B. Merialdo. Tagging english text with a probabilistic model. *Computacional Linguistics: To Appear.*

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

Masami Nakamura, Katsuteru Maruyama, Takeshi Kawabata, and Kiyohiro Shikano. 1990. Neural network approach to word category prediction for english texts. In *Proceedings of the International Conference on Computational Linguistics — COLING.*

L. Rabiner. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE*, 2(77):257 – 286. Reprinted in W&L, 267 – 296.

D. E. Rumelhart and J.L. McClelland. 1986. *Parallel Distributed Processing*, volume 1. MIT Press.

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

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

Aline Villavicencio, Nuno C. Marques, José Gabriel Lopes, and Fábio Villavicencio. 1995. 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. Springer Verlag. (10 pages).

Aline Villavicencio. 1995. Assessing a part-of-speech tagger for portuguese. Master's thesis, Universidade Federal de Rio Grande do Sul, Porto Alegre, Brasil, September. In Portuguese.