A CONNECTIONIST APPROACH TO PART-OF-SPEECH TAGGING*

F. Zamora-Martínez[1], M. J. Castro-Bleda[2], S. España-Boquera[2], Salvador Tortajada[3] and P. Aibar[4]
[1]*Departamento de Ciencias Físicas, Matemáticas y de la Computación*
Universidad CEU-Cardenal Herrera, Alfara del Patriarca (Valencia), Spain

[2]*Departamento de Sistemas Informáticos y Computación*
Universidad Politécnica de Valencia, Valencia, Spain

[3]*IBIME, Instituto de Aplicaciones de Tecnologías de la Información y de las Comunicaciones Avanzadas*
Universidad Politécnica de Valencia, Valencia, Spain

[4]*Departamento de Lenguajes y Sistemas Informáticos*
Universitat Jaume I, Castellón, Spain

Keywords: Natural language processing, Part-Of-Speech tagging, Neural networks, Multilayer perceptron.

Abstract: In this paper, we describe a novel approach to Part-Of-Speech tagging based on neural networks. Multilayer perceptrons are used following corpus-based learning from contextual and lexical information. The Penn Treebank corpus has been used for the training and evaluation of the tagging system. The results show that the connectionist approach is feasible and comparable with other approaches.

1 INTRODUCTION

The major purpose on Natural Language Processing research is to parse and understand language. Before achieving this goal several techniques have to be developed focusing on intermediate tasks such as Part-Of-Speech (POS) tagging. POS tagging attempts to label each word in a sentence with its appropriate part of speech tag from a previously defined set of tags or categories. Thus, POS tagging helps in parsing the sentence, which is in turn useful in other Natural Language Processing tasks such as information retrieval, question answering or machine translation.

POS tagging can also be seen as a disambiguation task because the mapping between words and the tagspace is usually one-to-many. There are words that have more than one syntactic category. This POS tagging process tries to determine which of the tags from a finite set of categories is the most likely for a particular use of a word in a sentence.

In order to decide the correct POS tag of a word there are basically two possible sources of information. The first one is the contextual information. This information is based on the observation of the different sequences of tags, where some POS sequences are common, while others are unlikely or impossible; therefore, looking at the tags of each contextual word can give a significant amount of information for POS tagging. For instance, a personal pronoun is likely to be followed by a verb rather than by a noun. The second source of information is called the lexical information and it is based on the word itself and the crucial information it can give about the correct tag. For instance, the word "*object*" can be a noun or a verb, thus the set of possible tags is significantly reduced. In fact, Charniak (Charniak et al., 1993) showed that assigning simply the most common tag to each word can perform at a level of 90% correct tags. Currently, nearly all modern taggers make use of a combination of contextual and lexical information.

Different approaches have been proposed for solving POS tagging disambiguation. The most relevant ones are ruled-based tagging (Voutilainen, 1999), probabilistic models (Merialdo, 1994) or based on Hidden Markov Models (Brants, 2000; Pla and Molina, 2004), on memory-based learning (Daelemans et al., 1996) and on the maximum entropy principle (Ratnaparkhi, 1996). Hybrid approaches which

*This work has been partially supported by the Spanish Ministerio de Educación y Ciencia (TIN2006-12767, TIN2008-06856-C05-02, PTQ-08-01-06802).

Zamora-Martínez F., J. Castro-Bleda M., España-Boquera S., Tortajada S. and Aibar P. (2009).
A CONNECTIONIST APPROACH TO PART-OF-SPEECH TAGGING.
In *Proceedings of the International Joint Conference on Computational Intelligence*, pages 421-426
DOI: 10.5220/0002313004210426
Copyright © SciTePress

combine the power of ruled-based and statistical POS taggers have been developed, like transformation-based learning (Brill, 1995). Recently, support vector machines have also been developed for POS tagging with very good results (Giménez and Márquez, 2004).

In the last few years, artificial neural network approach to POS tagging has been increasingly investigated due to its ability to learn the associations between words and tags and to generalize to unseen examples from a representative training data set. In (Schmid, 1994), a connectionist approach called *Net-Tagger* performed considerably well compared to statistical approaches; in (Benello et al., 1989) neural networks were used for syntactic disambiguation; in (Martín Valdivia, 2004), a Kohonen network was trained using the LVQ algorithm to increase accuracy in POS tagging; in (Marques and Pereira, 2001), feed-forward neural networks were used to generate tags for unknown languages; recurrent neural networks were also used in (Pérez-Ortiz and Forcada, 2001) for this task; other examples are (Ahmed et al., 2002; Tortajada Velert et al., 2005).

In the following section, the classical probabilistic model is explained in order to establish a comparison with the connectionist model, which is explained in Section 3. The corpus used for training and testing the neural POS taggers was the well-known Penn Treebank Corpus (Marcus et al., 1993). We explain its characteristics in Section 4. Training and performance of the connectionist systems are described in Section 5. Finally, some conclusions are remarked in Section 6.

2 PROBABILISTIC MODEL

One of the main approaches for POS tagging tasks is based on stochastic models (Jurafsky and Martin, 2000). From this point of view, POS tagging can be defined as a maximization problem. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_k\}$ be a set of POS tags and let $\mathcal{W} = \{w_1, w_2, \ldots, w_m\}$ be the vocabulary of the application. The goal is to find the sequence of POS tags that maximizes the probability associated to a sentence $w_1^n = w_1 w_2 \ldots w_n$, i.e.:

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n). \tag{1}$$

Using Bayes' Theorem, equation (1) turns into equation (2):

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(w_1^n | t_1^n) P(t_1^n). \tag{2}$$

The estimation of these parameters are time consuming and some assumptions are needed in order to simplify the computation of the expression (2). For these models, it is assumed that words are independent of each other and a word's identity only depends on its tag, thus we obtain the *lexical* probabilities,

$$P(w_1^n | t_1^n) \approx \prod_{i=1}^{n} P(w_i | t_i). \tag{3}$$

Another one establishes that the probability of one tag to appear only depends on its predecessor tag,

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1}). \tag{4}$$

This is called a bigram class, which is useful to obtain the *contextual* probability. If a trigram class is used the the expression is

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i | t_{i-1}, t_{i-2}). \tag{5}$$

This represents the probability of having the i-th POS tag, t_i, given that the two preceding tags are t_{i-1} and t_{i-2}.

With these assumptions, a typical probabilistic model following equations (2), (3) and (4) is expressed as:

$$\begin{aligned} \hat{t}_1^n &= \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n) \\ &\approx \underset{t_1^n}{\operatorname{argmax}} \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1}), \end{aligned} \tag{6}$$

where $\hat{t}_1^n$ is the best estimation of POS tags for the given sentence $w_1^n = w_1 w_2 \ldots w_n$ and considering that $P(t_1 | t_0) = 1$.

The probabilistic model has some limitations: it does not model long-distance relationships and the contextual information takes into account the context on the left while the context on the right is not considered. Both limitations can be overwhelmed using articial neural networks models, although in this paper we just considered to exploit the contextual information on the right side of the ambiguous word.

3 CONNECTIONIST MODEL

A connectionist model for POS tagging based on a multilayer perceptron network trained with the error backpropagation algorithm was presented in a previous work (Tortajada Velert et al., 2005). In that model, both the tag-level contextual information and the word-level information were used to predict the POS tag of the ambiguous input word. The main difference between these models and the classical probabilistic models is that future context, i.e. the context on the right, is taken into account. Thus, the network input consists in the past and future tag context of the ambiguous word and the word itself. The output of the network is the corresponding tag for the ambiguous input word. Therefore, the network learns a mapping between ambiguous words and tags as:

$$F(w_i, context) = t_i, \qquad (7)$$

where context refers to the group of tags $t_{i-p}, t_{i-(p-1)}, \ldots, t_{i-1}, t_{i+1}, \ldots, t_{i+(f-1)}, t_{i+f}$, being p the size of the left (past) context, and f the size of the right (future) context. The ambiguous input word w_i is locally codified, i.e. the unit representing the word is activated while the others are not. The weights of the multilayer perceptron are the parameters of the function F.

In our first approach to connectionist POS tagging (Tortajada Velert et al., 2005), a typical multilayer perceptron with one hidden layer was used. In this work, we have added to the net a new hidden layer that performs a projection of the locally codified word to a more compact distributed codified word (Zamora-Martínez et al., 2009). This projection layer was required because the size of the vocabulary of ambiguous words in the Penn Treebank Corpus labeling task is larger than in our previous experiments (Tortajada Velert et al., 2005).

Besides, as pointed out at the introduction, another useful source of information has been used: every possible tag with which the target word is labeled in the training corpus was added to the input of the network. Thus, expression (7) is better expressed for this model as:

$$F(w_i, T_i, context) = t_i, \qquad (8)$$

where T_i is the set of POS tags that have been found to be related to the ambiguous input word w_i in the training corpus.

When evaluating the model, there are words that have never been seen during training; therefore, they do not belong neither to the vocabulary of known ambiguous words nor to the vocabulary of known non-ambiguous words. These words are called "unknown words". In order to tag these unknown words the network uses an additional input unit. Figure 1 represents a connectionist model with all of these characteristics. We will refer to this system as MLP_{All}. When an unknown word is to be tagged, every tag is activated at the input.

Section 5 shows that unknown words present the hardest problem for the network to tag correctly. A way to avoid this handicap is to combine two multilayer perceptrons in a single system, where the first one is mainly dedicated to the known ambiguous words and the second one is specialized in unknown words. A scheme of a multilayer perceptron dedicated to tag unknown words is illustrated in Figure 2. The whole POS tagging system with the two multilayer perceptrons is shown in Figure 3. We will refer to this system as $MLP_{Combined}$, the multilayer perceptron specialized in unknown words will be MLP_{Unk} and the one for ambiguous known words, MLP_{Know}.

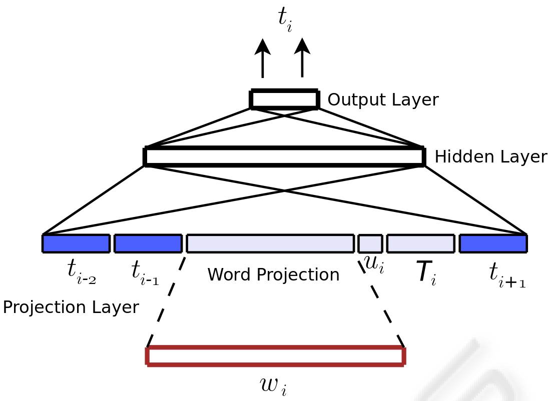

Figure 1: MLP_{All}: POS tagging system with a multilayer perceptron for tagging known ambiguous words and unknown words. The known ambiguous input word w_i is locally codified at the input of the projection layer. Unknown words are codified as an additional input unit u_i. In this case, two labels of past context and one label of future context are used. T_i is the set of POS tags that have been found to be related to the ambiguous input word w_i. When the input is an unknown word, every tag in T_i is activated.

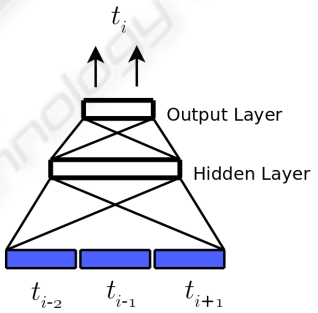

Figure 2: MLP_{Unk}: A multilayer perceptron dedicated to POS tag unknown words. The input to the multilayer perceptron is the context of the unknown word at time i.

4 THE PENN TREEBANK CORPUS

The corpus used in the experiments was the well-known part of the Wall Street Journal that had been processed in the Penn Treebank Project (Marcus et al., 1993). This corpus consists of a set of English texts from the Wall Street Journal distributed in 25 directories containing 100 files with several sentences each

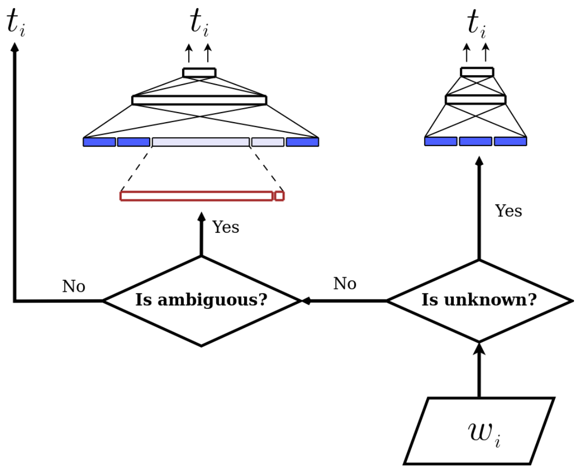

Figure 3: *MLP$_{Combined}$*: POS tagging system where two multilayer perceptrons are combined. The model on the left (*MLP$_{Know}$*) is dedicated to known ambiguous words, i.e., ambiguous words included in the training vocabulary, and the model on the right (*MLP$_{Unk}$*) is specialized in unknown words.

one. The total number of words is about one million, being 49 000 different. The whole corpus was automatically labeled with a POS tag and a syntactic labeling. The POS tag labeling consists of a set of 45 different categories. One more tag was added to take into account the beginning of a sentence (the ending of a sentence is in the original set of tags), thus resulting in a total amount of 46 different POS tags.

The corpus was divided in three sets: training, tuning and test. The main characteristics of these partitions are described in Table 1.

Considering only the ambiguous words of the training set we obtain a vocabulary of more than 6 000 words, which is a prohibitive amount if we codify each word as a unit of the neural network input. In order to reduce the dimensions of the vocabulary and to obtain some samples of unknown words for training, a cut-off was set to an absolute frequency of 10 repetitions. If a word had a frequency smaller than 10, then it was treated as an unknown word. Moreover, POS tags appearing in a word 90% less than the most repeated tag in such word, and less than 3 times in an absolute frequency, were also eliminated, because these tags are mainly erroneous tags.

For example, the word "*are*" is ambiguous in the corpus because it appears 3 639 times like a VBP and one time like NN, NNP and IN. The NN, NNP and IN tags are errors, as shown in the sentence: "*Because many of these subskills –the symmetry of geometrical figures, metric measurement of volume, or pie and bar graphs, for example–* ***are*** *only a small part of the total fifth-grade curriculum, Mr. Kaminski says, the preparation kits wouldn't replicate too many, if their real intent was general instruction or even general familiarization with test procedures.*", where "*are*" is labeled like NN.

With these assumptions we finally got a vocabulary of 2 563 ambiguous words for training. The number of words that are codified with the unknown word symbol in the training corpus are 2 826.

Table 2 shows the total number of ambiguous and unambiguous words for each partition, along with the unknown words after this cut-off preprocessing.

5 THE CONNECTIONIST POS TAGGERS

Different multilayer perceptrons were trained for the two POS tagging systems. The *MLP$_{All}$* model was trained with known ambiguous words and unknown words. The *MLP$_{Combined}$* system is composed by two independent multilayer perceptrons (see Figure 3), one multilayer perceptron for known ambiguous words (*MLP$_{Know}$*) and other multilayer perceptron for unknown words (*MLP$_{Unk}$*).

The multilayer perceptron networks to be used as POS classifiers were trained with the error backpropagation algorithm (Rumelhart et al., 1986). The topology and parameters of multilayer perceptrons in the trainings are shown in Table 3, and they were selected in previous experimentation. For the experiments we have used a toolkit for pattern recognition tasks developed by our research group (España et al., 2007).

Then, the next step is to evaluate the impact of the amount of contextual information in the accuracy of the model. In these experiments, the multilayer perceptrons are trained like a classifier. Under this assumption, when a multilayer perceptron is tagging a word, the past and future context are extracted from the correct tags. The exploration results of the different combination of contextual information for the *MLP$_{All}$* and *MLP$_{Combined}$* systems are shown in Table 4. The results are calculated for the tuning set of the corpus.

The best combination of contextual information is achieved with two labels in the past context, and a future context of just one label for both systems. The best performance for known ambiguous words is a 5.7% of POS tagging error rate, achieved for the *MLP$_{Combined}$* system. For the case of unknown words, the best performance is also obtained with the *MLP$_{Combined}$* system, a 36.8% of wrong classified unknown words. Computing the total POS tagging error rate, the *MLP$_{Combined}$* systems obtains a 4.2%, that is,

Table 1: Partitions from the Penn Treebank corpus for training, tuning and testing. The total number of sentences and words are the sum of the different sets, while the total vocabulary size is the cardinal number of the intersection of each partition.

Dataset	Directory	Num. of sentences	Num. of words	Vocabulary size
Training	00-18	38 219	912 344	34 064
Tuning	19-21	5 527	131 768	12 389
Test	22-24	5 462	129 654	11 548
Total	00-24	49 208	1 173 766	38 452

Table 2: Number of unambiguous, ambiguous and unknown words in each partition from the Penn Treebank corpus after preprocessing (cut-off). The total number of words are the sum of the different sets. The vocabulary of ambiguous words for training is 2 563.

Dataset	Num. of words	Unambiguous	Ambiguous	Unknown
Training	912 344	484 622	378 898	48 824
Tuning	131 768	65 552	53 956	6 733
Test	129 654	63 679	54 607	5 906
Total	1 173 766	613 853	487 461	61 463

Table 3: Parameters of the MLP_{All} and $MLP_{Combined}$, where p is the size of the left (past) context, and f is the size of the right (future) context. The size of the vocabulary of known ambiguous words is $|\Omega| = 2563$ and there are $|T| = 46$ POS tags.

		$MLP_{Combined}$	
Parameter	MLP_{All}	MLP_{Know}	MLP_{Unk}
Input layer size:	$\|T\|(p+f+1)+\|\Omega\|+1$	$\|T\|(p+f+1)+\|\Omega\|$	$\|T\|(p+f)$
Output layer size:	$\|T\|$	$\|T\|$	$\|T\|$
Projection layer size:	128	128	–
Hidden layer size:	100	100	100
Hidden layer activation function:	Hyperbolic Tangent		
Output layer activation function:	Softmax		
Learning rate:	0.005		
Momentum:	0.001		
Weight decay:	0.0000001		

Table 4: MLP_{All} and $MLP_{Combined}$: POS tagging error rate for the tuning set varying the context (p is the past context, and f is the future context). **Known** refers to the disambiguation error for known ambiguous words. **Unk** refers to the POS tag error for unknown words. **Total** is the total POS tag error, with ambiguous, non-ambiguous, and unknown words.

Model	p f	Known	Unk	Total
MLP_{All}	2 1	5.8%	37.4%	4.3%
	3 1	5.9%	37.2%	4.3%
	4 1	6.1%	38.3%	4.5%
$MLP_{Combined}$	2 1	**5.7%**	**36.8%**	**4.2%**
	3 1	5.9%	37.5%	4.3%
	4 1	6.0%	38.9%	4.4%

the total disambiguation error, with ambiguous, non-ambiguous and unknown words.

Table 5: POS tagging error rate for the tuning and test sets with the $MLP_{Combined}$ model. **Known** refers to the disambiguation error for known ambiguous words. **Unk** refers to the POS tag error for unknown words. **Total** is the total POS tag error, with ambiguous, non-ambiguous, and unknown words.

Partition	p f	Known	Unk	Total
Tuning	2 1	5.7%	36.8%	4.2%
Test	2 1	6.1%	36.7%	4.3%

Performance for the best system (the $MLP_{Combined}$ system with a past context of two labels and a future context of one label) with the tuning and test sets are shown in Table 5. A total disambiguation error - ambiguous, non-ambiguous, and unknown words- of 4.3% was achieved for the test set.

6 CONCLUSIONS

To evaluate the system a comparison with other approaches is necessary. Several works have used the

SciTePress
Science and Technology Publications

same corpus, and the same partitions. Our best result is a 4.3% of total error with the $MLP_{Combined}$ tagging system. This result is worse than the best, achieved with SVMs (Giménez and Márquez, 2004), a 2.8% tagging error. But if we focus our attention in the error in known ambiguous words, our model is comparable to SVMs (Giménez and Márquez, 2004) (they obtained a 6.1% POS tagging error rate). The major difference is that the adjustment of the unknown word classification is more accurate in the referenced works than in our approach.

In this line, our inmediate goal is to improve the performance of the MLP_{Unk} network. When dealing with unknown words, introducing relevant morphological information related to the unknown input word can be useful for POS tagging. Other approaches also use this kind of information (as in (Giménez and Márquez, 2004; Gascó and Sánchez, 2007)).

REFERENCES

Ahmed, Raju, S., Chandrasekhar, P., and Prasad, M. (2002). Application of multilayer perceptron network for tagging parts-of-speech. In *Proc. Language Engineering Conference*, pp. 57–63.

Benello, J., Mackie, A., and Anderson, J. (1989). Syntactic category disambiguation with neural networks. *Computer Speech and Language*, 3:203–217.

Brants, T. (2000). TnT: a statistical part-of-speech tagger. In *Proc. 6th conference on Applied Natural Language Processing*, pp. 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(4):543–565.

Charniak, E., Hendrickson, C., Jacobson, N., and Perkowitz, M. (1993). Equations for part-of-speech tagging. In *Proc. National Conference on Artificial Intelligence*, pp. 784–789.

Daelemans, W., Zavrel, J., Berck, P., and Gillis, S. (1996). MBT: A Memory-Based Part-of-Speech Tagger Generator. In *Proc. 4th Workshop on Very Large Corpora*, pp. 14–27.

España, S., Zamora, F., Castro, M.-J., and Gorbe, J. (2007). Efficient BP Algorithms for General Feedforward Neural Networks. In vol. 4527 of *LNCS*, pp. 327–336. Springer.

Gascó, G. and Sánchez, J. (2007). Part-of-speech tagging based on machine translation techniques. In *Patt. Recog. and Image Anal.*, pp. 257–264.

Giménez, J. and Márquez, L. (2004). SVMTool: A general pos tagger generator based on support vector machines. In *Proc. 4th Conf. on LREC*.

Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing*. Prentice Hall.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Marques, N. and Pereira, G. (2001). A POS-Tagger generator for Unknown Languages. *Procesamiento del Lenguaje Natural*, 27:199–207.

Martín Valdivia, M. (2004). *Algoritmo LVQ aplicado a tareas de Procesamiento del Lenguaje Natural*. PhD thesis, Universidad de Málaga.

Merialdo, B. (1994). Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2):155–171.

Pérez-Ortiz, J. and Forcada, M. (2001). Part-of-speech tagging with recurrent neural networks. In *Proc. IJCNN*, pp. 1588–1592.

Pla, F. and Molina, A. (2004). Improving Part-of-Speech Tagging using Lexicalized HMMs. *Natural Language Engineering*, 10(2):167–189.

Ratnaparkhi, A. (1996). A Maximum Entropy Part-Of-Speech Tagger. In *Proc. 1st Conference on EMNLP*, pp. 133–142.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). *Parallel distributed processing: explorations in the microstructure of cognition*, chap. Learning internal representations by error propagation, pp. 318–362. MIT Press.

Schmid, H. (1994). Part-of-Speech tagging with neural networks. In *Proc. International COLING*, pp. 172–176.

Tortajada Velert, S., Castro Bleda, M. J., and Pla Santamaría, F. (2005). Part-of-Speech tagging based on artificial neural networks. In *Proc. 2nd Language & Technology Conference*, pp. 414–418.

Voutilainen, A. (1999). *Syntactic Wordclass Tagging*, chapter Handcrafted rules, pp. 217–246. H. van Halteren.

Zamora-Martínez, F., Castro-Bleda, M., and España-Boquera, S. (2009). Fast evaluation of connectionist language models. In vol. 4507 of *LNCS*, pp. 144–151. Springer.