Part-of-Speech tagging based on artificial neural networks

Salvador Tortajada Velert, María José Castro Bleda, Ferran Pla Santamaría

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, 46022 Valencia, Spain
{stortajada,mcastro,fpla}@dsic.upv.es

Abstract

In this paper, we describe a Part-of-Speech tagging system based on connectionist models. A Multilayer Perceptron is used following corpus-based learning from contextual and lexical information. The Spanish corpus LexEsp has been used for the training and evaluation of the tagging system based on artificial neural networks. Different approaches have been compared and results with the Hidden Markov Model systems are also given. The results show that the connectionist approach is feasible.

1. Introduction

Part-of-Speech (PoS) tagging attempts to assign a PoS tag or category from a previously defined set to each word in a text. In this paper, we use the tag set Parole. PoS tagging is a very useful task in Natural Language Processing applications such as Information Retrieval, Machine Translation or Document Indexing.

Over the last few years, different approaches have been proposed for solving morpho-syntactic disambiguation. The most relevant ones are based on Hidden Markov Models (Brants, 2000; Pla and Molina, 2004), on transformation-based learning (Brill, 1995), on memory-based learning (Daelemans et al., 1996) and on the maximum entropy principle (Ratnaparkhi, 1996). Hybrid approaches which combine the power of rule-based and statistical PoS taggers can also be implemented (Padró, 1998). Moreover, some works combine the output of different taggers by means of some "voting" methods in order to improve their performance (Brill and Wu, 1998; Van Halteren et al., 2001).

Nevertheless, PoS tagging based on connectionist models has been less extensively investigated. For example, in (Martín Valdivia, 2004; Martín Valdivia et al., 2002), a Kohonen network is trained using the LVQ algorithm to increase accuracy in PoS tagging; in (Marqués and Pereira, 2001) feed-forward neural networks are used to generate tags for unknown languages; in (Benello et al., 1989) neural networks are also used for syntactic disambiguation; in (Pérez-Ortiz and Forcada, 2001) recurrent neural networks are used for PoS tagging; and in (Veronis and Ide, 1990) very large neural networks are used to solve this task.

This paper presents a way to generate PoS tags based on Multilayer Perceptrons, which use two variants of the error backpropagation algorithm and different topologies, depending on the contextual information being used. This system has been experimentally evaluated with the Spanish corpus LexEsp (Sebastián et al., 2000), and the tag set Parole. Different approaches have been compared and results with the Hidden Markov Model systems are also given.

2. Corpus LexEsp

The Spanish corpus LexEsp consists of a collection of texts divided into a training set and a test set. There follows an example of a fragment of the corpus with Parole categories:

Cuando *cuando* CS
escribo *escribir* VMI
esto *esto* PD
la *la* TD
Madre_Coraje *madre_coraje* NP

The first column refers to the word itself, the second column shows the lemma of the word and the last column shows the lexical category without any morphological information. There are 52 tags and the selection of ambiguous words is limited to their ambiguous appearance in the corpus. This means that a word will be ambiguous if and only if it presents different lexical functions throughout the entire global corpus. The LexEsp has 96 961 words, of which 25 538 (around 26%) are ambiguous and where 6 585 ambiguous words belong to the test set. The vocabulary size of ambiguous words in the corpus is 725.

In a first attempt to solve the problem of PoS tagging, we decided to simplify the complexity of the Parole tags and then extend the results obtained to solve the original problem. The simplified tags for the same fragment of the corpus are:

Cuando *cuando* C
escribo *escribir* V
esto *esto* P
la *la* T
Madre_Coraje *madre_coraje* N

There are 13 simplified tags: nouns, punctuation marks, prepositions, verbs, articles, adjectives, pronouns, conjunctions, adverbs, determiners, numbers, digits and others. The vocabulary of ambiguous words of the entire corpus is 594 and the total amount rises to 24 100 (around 25%), where 6 264 ambiguous words belong to the test set. Table 1 shows the absolute and relative frequency of

Tag	Words		Ambiguous	
Nouns	20 357	20.99%	2 009	8.34%
Punctuation	13 801	14.23%	0	0.00%
Prepositions	13 346	13.76%	1 878	7.79%
Verbs	12 875	13.28%	1 029	4.27%
Articles	10 753	11.09%	8 311	34.49%
Adjectives	6 561	6.77%	1 326	5.50%
Pronouns	5 466	5.64%	2 688	11.15%
Conjunctions	5 242	5.41%	4 059	16.84%
Adverbs	4 808	4.96%	1 011	4.20%
Determiners	2 793	2.88%	1 371	5.69%
Numerals	477	0.49%	386	1.60%
Ciphers	390	0.40%	6	0.02%
Others	92	0.10%	26	0.11%
TOTAL	96 961	100%	24 100	100%

Table 1: Frequency of total words and ambiguous words for simple categories.

each simplified tag and the number of ambiguous words existing in the corpus.

A validation set was created by randomly selecting sentences out from the training set. This validation set is used to decide when to stop the training of the neural network. It usually stops when a minimum is reached on the error curve of the validation set. This is done in order to avoid over-fitting which can lead to poor generalisation due to the learning of specific training data rather than the learning of the general properties of the underlying linguistic properties.

Morpho-syntactic disambiguation is handled with context data, making use of right and left bigrams and/or trigrams. With left bigrams (B-), the category c_i of the ambiguous word w_i will only be related to the category c_{i-1} of the previous word w_{i-1}. In left trigrams (T-), the contextual information will be led by categories c_{i-2} and c_{i-1} from the previous words w_{i-2} and w_{i-2}. We will also take into account the use of the right context by means of the knowledge given by the categories c_{i+1} and c_{i+2} from the following words w_{i+1} and w_{i+2}.

3. Experiments with simplified tags

The use of simplified categories allows us to make a first attempt at solving the problem of PoS tagging by using Multilayer Perceptrons. By reducing the number of classes, i.e., categories, we also reduce the complexity of the problem, thus, we can determine if this approach is feasible or not.

As we have already mentioned, five different models that depend on the contextual information have been developed and designed. These approaches are: left bigrams (B-), left trigrams (T-), left-right bigrams (B-B), left trigrams and right bigrams (T-B), and left-right trigrams (T-T).

Figure 1 shows a scheme of these five models. For instance, the experiment with left trigrams and right bigrams (T-B) needs a Multilayer Perceptron with an input layer of 633 units: 594 units to represent the ambiguous word w_i and 13 units for each category c_{i-2}, c_{i-1} and c_{i+1}. This

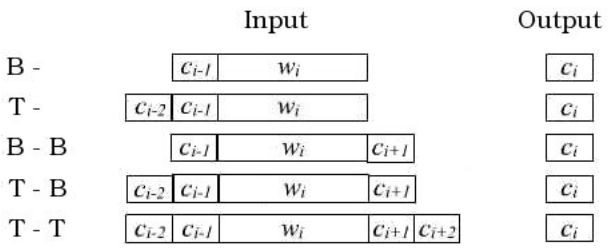

Figure 1: *Scheme of the different approaches.*

can be seen as a vector of bits where the activated units represent both the contextual information and the ambiguous word that we are dealing with.

We have made a series of experiments for each approach. The first goal was to select the best network topology from a collection of a priori proposed networks. Each one was trained through the standard error backpropagation algorithm (Rumelhart et al., 1986) with a small learning rate of 0.5. The training was stopped when the disambiguation error of the validation set was not improved in 100 epochs. The training and test evaluation were carried out using the Stuttgart Neural Network Simulator of the University of Stuttgart (Zell et al., 1998).

We chose the topology which showed the best behaviour for the validation patterns. Then, with this topology, we trained the networks with two algorithms (standard error backpropagation and error backpropagation with momentum) and different parameters for each one. In particular, the learning rate, ρ, varied from 0.01 to 0.9 and the momentum, μ, varied from 0.1 to 0.5. The trained network which presented the lowest disambiguation error for the validation set was evaluated using the test set in order to know its real accuracy (see Table 2). The best network topologies, algorithms and parameters used in each approach are shown below:

- **Left bigrams** (B-): one hidden layer of 8 units and error backpropagation algorithm with learning rate $\rho = 0.3$ and momentum term $\mu = 0.3$.
- **Left trigrams** (T-): one hidden layer with 32 units and standard backpropagation algorithm with learning rate $\rho = 0.1$.
- **Left-right bigrams** (B-B): one hidden layer with 16 units and standard backpropagation using $\rho = 0.1$ as the learning rate.
- **Left trigrams and right bigrams** (T-B): one hidden layer with 32 units and error backpropagation with learning rate $\rho = 0.05$ and momentum term $\mu = 0.4$.
- **Left-right trigrams** (T-T): 32 units in the hidden layer, trained with error backpropagation with learning rate $\rho = 0.1$ and momentum term $\mu = 0.1$.

Finally, we chose the network which offered the best performance (in this case, left trigrams) which obtained a disambiguation error of 7.85%. With this configuration, we trained a new network by combining the validation and

Strategy	Disambiguation error
B-	8.78%
T-	7.85%
B-B	8.11%
T-B	8.13%
T-T	7.95%
T- (TR+VA)	8.03%

Table 2: Disambiguation error for the test set with simplified tags.

Strategy	Disambiguation error
B-	9.96%
T-	10.89%
B-B	11.33%
T-B	9.40%
T-T	9.67%
T-B (TR+VA)	9.25%

Table 3: Disambiguation error for the test set with Parole tags.

training sets into one training set without validation. Thus, a net with topology 620-32-13 was trained with the standard error backpropagation algorithm with learning rate $\rho = 0.1$ until the disambiguation error of the training set itself was lower than the error reached by the validation set in the original experiment. This result can be seen in the last line of Table 2.

4. Experiments with Parole tags

In order to get a system capable of disambiguating words using more complex categories, we have conducted new experiments from the original corpus tagged with Parole categories. For this new experiment, we had 52 Parole tags in all.

The immediate consequence was a greater complexity of the problem. The networks have more input and output units as there are 52 different tags, which also increases the number of connections between the different layers of neurons. This leads to a greater computational cost. For this reason, instead of doing a thorough proof scanning, we used the best results achieved in the previous experiments with simplified tags. The network topologies, the algorithms and their specific parameters are those cited above. The results of the trained networks for the test set evaluation can be seen in Table 3.

Similarly to the previous experiment with simplified tags, we made a final proof with the approach which achieved the best performance, by training with the validation set attached to the training set. Thus, the learning phase lasted until the error was lower than the disambiguation error reached for the validation set in the original experiment. The last line of Table 3 shows the performance of this system.

Tags	Disambiguation error
Simplified	9.26%
Parole	8.85%

Table 4: Disambiguation error for the test set with simplified and Parole tags using HMMs.

5. Comparison with other approaches

In order to compare the results presented in this work with other related approaches, we used the PoS tagger based on Hidden Markov Models introduced in (Pla and Molina, 2004; Pla et al., 2001). The contextual model is represented by means of trigrams smoothed with linear interpolation. Lexical probabilities are estimated by maximum likelihood from the annotated data. A smoothing method was also applied to handle unknown words. A Spanish morphological analyser, MACO (Acebo et al., 1994; Carmona et al., 1998), was included to provide the possible grammatical categories for each word. The results obtained for both tag sets are shown in Table 4.

Table 4 shows that the system based on neural networks achieved a better result than the one based on HMMs when the number of tags is small. As the size of the tag set increases, the results of the connectionist approach become worse. This may be because neural networks need more data for their correct training. Nevertheless, we have to take into account that the Parole tag set experiment has been evaluated without performing a complete scanning to establish the best parameters.

6. Conclusions

The goal of this paper was to prove that neural networks are a feasible approach for PoS tagging tasks. The results show that this is true. The presented system based on feed-forward neural networks demonstrates a good behaviour with simplified tag sets. In general, many Natural Language Processing applications require very detailed categories, but there are others, like Word Sense Disambiguation where access to the semantic net WordNet only needs to know if a word is a noun, a verb, an adjective or an adverb. Despite the satisfactory results it is worth commenting on some features in order to know its limitations and its possible future improvement.

A major downside has to do with the selection and treatment of ambiguous words. These words have been chosen as ambiguous considering only the corpus without any other linguistic criterion. This makes our approach a “closed problem”, because it is limited to the linguistic information handled in the initial process and during the training and evaluation phases. Consequently, if we try to disambiguate a text whose linguistic field is different from the one we learnt, some ambiguous words would not be resolved, thus, worsening the accuracy of the system.

A possible solution to this problem could be the a priori use of a morphological analyser to point out which words are ambiguous. So, whatever the context may be, every ambiguous word could be coded and then learnt by the neural network.

Another choice would be a word-independent system which would return a solution according to the contextual information only. When analyzing the training, validation and test patterns, we found that certain words were not represented in each set. The words that did not appear in the training set, but did appear in the test set, were observed to prove how the neural network was classifying them. As these samples did not exist in the training set, the network was not able to learn them and, we can say, the network was unaware of them. Therefore, when the network attempts to disambiguate the test set, the only information provided is the contextual one. Thus, the syntactic structure of the sentence controls the decision of tagging an ambiguous word with a specific category.

In (Hanson and Kegl, 1987), the ability to absorb at least some syntactic knowledge simply from exposure to samples of natural language text was demonstrated. These results show that this system can infer some syntactic structures from natural language sentences. This feature suggests a new approach to the problem, focusing the system by learning contextual information only and forgetting the morphologic information of the word.

Another limitation of our approach to the tagging problem is related to a typical characteristic of language: the lack of proportion between the use of some linguistic categories and the use of others. For instance, in a text, we find more nouns or prepositions than numbers or foreign expressions. This is not a real problem in the linguistic field, but it is a handicap for training neural networks since the ideal training must be done using a high number of samples for each category.

This problem has been alleviated by putting together some Parole categories into new ones. For example, the tags I (interjections), X (residuals), W (dates), E0 (foreign expressions) and Y (abbreviations) were grouped in one category; the tags VAG (auxiliar gerund verbs), VAM (auxiliar imperative) and VAC (auxiliar conditional) were grouped in another one; and the tags Fcs, Ftp, Fs and Fch (corresponding to the punctuation symbols "?", "%", "/" and "]" respectively) were grouped in a third category. This is why the total number of PoS tags in this work was 52 and not 62.

To conclude, the right contextual information, i.e., right bigrams and trigrams, is a drawback at the moment of tagging a text in a non-supervised way. Suppose that we have a text which has not been previously PoS tagged. If we find a sequence of words $w_{i-2}, w_{i-1}, w_i, w_{i+1}$, where w_i is ambiguous, we can assume that we will have categories c_{i-2} and c_{i-1}. But if the word w_{i+1} is also ambiguous, we cannot know the category c_{i+1}, which is absolutely necessary in right-context approaches. This problem can be solved in several ways. For instance, by using a new input unit in the Multilayer Perceptron which would explicitly point out if the right context has an ambiguous word or not. Another way to solve the problem is to perform the classification by means of a left-context neural network before doing a right-context disambiguation. We can also use some more sophisticated methods like a combination of neural networks which could use a left-context network in cases similar to the one mentioned above and a two-sided-context network when the word w_{i+1} is not ambiguous. There are many possibilities for this situation and each one would require a complete research.

7. Acknowledgements

This work has been partially supported by the Spanish CICYT under contract TIC2003-07158-C04-03.

8. References

Acebo, S. et al., 1994. MACO, Morphologic Analyzer Corpus Oriented. *Acquilex II, WP 31*.

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

Brants, Thorsten, 2000. TnT – A Statistical Part-of-Speech Tagger. In *Proceedings of the 6th Applied Natural Language Processing (ANLP)*. Seattle, Washington.

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.

Brill, Eric and Jun Wu, 1998. Classifier Combination for Improved Lexical Disambiguation. In *Proceedings of the Joint 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*. Montréal, Canada.

Carmona, J. et al., 1998. An Environment for Morphosyntactic Processing of Unrestricted Spanish Text. In *Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC)*. Granada, Spain.

Daelemans, W. et al., 1996. MBT: A Memory–Based Part–of–speech Tagger Generator. In *Proceedings of the 4th Workshop on Very Large Corpora*. Copenhagen, Denmark.

Hanson, S. J. and J. Kegl, 1987. PARSNIP: A Connectionist Network that Learns Natural Language Grammar from Exposure to Natural Language Sentences. In *Proceedings of the Ninth Anual Cognitive Science Society Meeting*. Hillsdale, New Jersey.

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

Martín Valdivia, M. T., 2004. *Algoritmo LVQ aplicado a tareas de Procesamiento del Lenguaje Natural*. Ph.D. thesis, Departamento de Lenguajes y Ciencias de la Computación. Universidad de Málaga.

Martín Valdivia, M. T., L. A. Ureña, and M. García, 2002. Resolución de la ambigüedad mediante redes neuronales. *Procesamiento del Lenguaje Natural*, 29:39–45.

Padró, Lluís, 1998. *A Hybrid Environment for Syntax–Semantic Tagging*. Ph.D. thesis, Departament de Llenguatges i Sistemes Informàtics. Universitat Politècnica de Catalunya.

Pérez-Ortiz, J. A. and M. L. Forcada, 2001. Part-of-speech tagging with recurrent neural networks. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*.

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

Pla, F., A. Molina, and N. Prieto, 2001. Evaluación de un etiquetador morfosintáctico basado en bigramas especializados para el castellano. *Procesamiento del Lenguaje Natural*, 27:215–225.

Ratnaparkhi, A., 1996. A Maximum Entropy Part–of–speech Tagger. In *Proceedings of the 1st Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1986. *PDP: Computational models of cognition and perception, I*, chapter Learning internal representations by error propagation. MIT Press, pages 319–362.

Sebastián, N. et al., 2000. LexEsp: Léxico informatizado del español. Technical report, Universitat de Barcelona.

Van Halteren, Hans, Jorn Zavrel, and Walter Daelemans, 2001. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics*, 27(2):199–229.

Veronis, J. and N. M. Ide, 1990. Word Sense Disambiguation with Very Large Neural Networks Extracted from Machine Readable Dictionaries. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING)*.

Zell, A. et al., 1998. *SNNS: Stuttgart Neural Network Simulator. User Manual, Version 4.2*. Institute for Parallel and Distributed High Performance Systems, University of Stuttgart, Germany.