A Neural Network Based Dutch Part of Speech Tagger

Mannes Poel Egwin Boschman Rieks op den Akker

University of Twente, Dept. Computer Science,
P.O. Box 217, 7500 AE Enschede, The Netherlands
{mpoel,infrieks}@cs.utwente.nl

Abstract

In this paper a Neural Network is designed for Part-of-Speech Tagging of Dutch text. Our approach uses the Corpus Gesproken Nederlands (CGN) consisting of almost 9 million transcribed words of spoken Dutch, divided into 15 different categories. The outcome of the design is a Neural Network with an input window of size 8 (4 words back and 3 words ahead) and a hidden layer of 370 neurons. The words ahead are coded based on the relative frequency of the tags in the training set for the word. Special attention is paid to unknown words (words not in the training set) for which such a relative frequency cannot be determined. Based on a 10-fold cross validation an approximation of the relative frequency of tags for unknown words is determined. The performance of the Neural Network is 97.35%, 97.88% on known words and 41.67% on unknown words. This is comparable to state of the art performances found in the literature. The special coding of unknown words resulted of an increase of almost 13% for the tagging of unknown words.

1 Introduction

A Part-of-Speech (PoS) Tagger is a system that assigns the correct Part-of-Speech or word class to each of the words in a document. Classical parts of speech are noun and verb, and a few others, but nowadays Part of Speech tag sets sub-divide these general word classes into smaller ones, such as proper names, singular nouns, mass nouns and plural nouns. Part of Speech tag sets typically contain from a little over twenty to more than a few hundred of different word classes. PoS tagging is a non-trivial task because most words are ambiguous: they can belong to more than one class, the actual class depends on the context of use. In this paper we approach the PoS tagging task using Neural Networks. For training and testing we use the Corpus Gesproken Nederlands (CGN - Spoken Dutch Corpus) a large morpho-syntactically annotated corpus. Figure 1 shows that the more common a word occurs in this corpus the more likely it is ambiguous. More details of this corpus are given in section 1.1. The data points are an average of the percentage of ambiguous words around that frequency. PoS tagging, or word class disambiguation, is the process of finding out the right word class for these ambiguous words. The result is then added as a label or 'tag' to the word. PoS tagging is often only one step in a text processing application. The tagged text can be used for deeper analysis, for example for chunk parsing or full parsing. Because the accuracy of the PoS tagging greatly influences the performance of the steps further in the pipeline [4], the accuracy of the PoS tagger is very important. In general PoS tagging can be seen as a sequential supervised learning problem [6] and various models for supervised machine learning have been applied to the problem of PoS tagging; memory based learning [5] , transformation rule based learning [3], (Hidden) Markov Models. Apart from the overall accuracy, relevant measures for PoS taggers concern the accuracy of handling unknown words, the amount of training data required (the learning rate), training time, tagging time, and the accuracy on different types of corpora. TnT, a trigram HMM tagger by Brants [1], has shown good results in both accuracy and training time as well as tagging time.

During the development of the CGN corpus, a number of methods for PoS tagging were compared and used for bootstrapping. Zavrel and Daelemans [13] report on a number of PoS taggers trained and tested

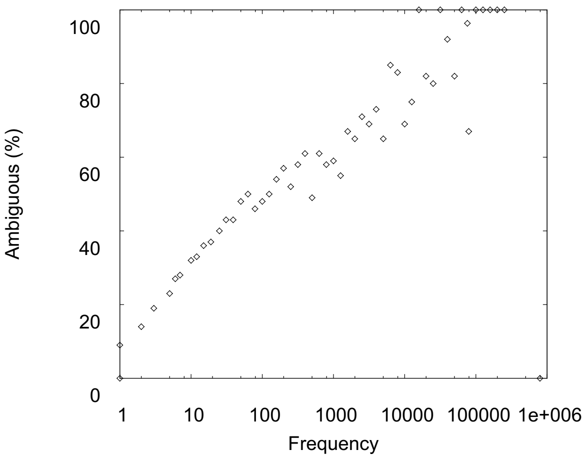

Figure 1: The average percentage of ambiguous words related to the number of occurrences in the CGN corpus.

on the CGN. The best overall performance reported in this study was 95.44%. Canisius and van den Bosch [4] used a subset of the CGN for constructing a memory based PoS tagger with a tagging performance of 95.96%.

The aim of this research is to find the appropriate ingredients for constructing a PoS tagger for spoken Dutch, using Neural Networks. One of the motivations is that, once trained, a Neural Network has efficient tagging performance. It should be remarked that in this study we focus on accuracy and handling of unknown words.

The methodology we use for designing a Neural Network based PoS tagger is as follows. First we determine which input features to use, this selection of input features is based on results found in the literature for different corpora. We will focus explicitly on the coding of unknown words.

Next we determine a small range of (near) optimal window sizes. Afterwards we evaluate different sizes for the hidden layer and select a small set of best ones, based on the performance. Finally we will evaluate the different combinations of the found parameters for window size and number of hidden neurons.

The Corpus Gesproken Nederlands (CGN - Spoken Dutch Corpus) will be introduced in Section 1.1. The detailed design approach for the Neural Network is explained in Section 2. In Section 3 the best found Neural Network will be evaluated. The final conclusions can be found in Section 4.

1.1 Spoken Dutch Corpus (CGN)

The Spoken Dutch Corpus (CGN) [8] is a database of contemporary spoken Dutch. It consists of almost 9 million transcribed words of spoken Dutch, divided into 15 different categories, cf. Table 1. Of these words, around two thirds originate from the Netherlands, the remaining one third from Flanders. The entire CGN is annotated with a large set of PoS tags. The full set consists of 316 different tags, which denote many different features of a word class. An example of such a tag is N(soort,ev,basis,zijd,stan) (noun, sort name, singular, basis (not diminutive), not neuter, standard case (not dative or genitive)). A full explanation of the features and their use can be found in [12]. Many of the pronouns contain even more features, up to nine. This subdivision is so fine-grained that many tags occur only a few times in the entire corpus. There are even 19 tags that occur only once. Although it is possible to discard all the subclasses and use only the main class, this would leave us with a set of only 12 tags (including LET and SPEC, for punctuation mark and special respectively). A tag set of this size is much smaller than what is commonly used in PoS tagging. Discarding all these features also reduces the value of the tagged data to further processing steps. To overcome this problem, the syntactic annotations use a different tag set, consisting of 72 tags. These tags are a reduced version of the full tag set. Only about ten percent of the corpus is tagged using these tags, but the tags can be automatically derived for the rest of the corpus using a set of simplification rules and the full tags. Table 2 shows an overview of this tag set. The challenges of using Neural Networks (NN's) on such a large corpus as the CGN is the size of the training set which requires a separation of the corpus, and to find a workable representation of the input.

Category	Type	Size in words
A	Face to face conversations	2626172
B	Interview with teacher Dutch	565433
C	Phone dialogue (recorded at platform)	1208633
D	Phone dialogue (recorded with mini disc)	853371
E	Business conversations	136461
F	Interviews and discussions recorded from radio	790269
G	Political debates, discussions and meetings	360328
H	Lectures	405409
I	Sport commentaries	208399
J	Discussions on current events	186072
K	News	368153
L	Commentaries on radio and TV	145553
M	Masses and ceremonies	18075
N	Lectures and discourses	140901
O	Text read aloud	903043

Table 1: The 15 different categories of the Dutch Spoken Corpus (CGN).

Tag numbers	Part-of-Speech Tag	Tags in the CGN corpus
$1 \ldots 8$	Noun	N1, N2, . . ., N8
$9 \ldots 21$	Verb	WW1, WW2, . . ., WW13
22	Article	LID
$23 \ldots 49$	Pronoun	VNW1, VNW2, . . ., VNW27
$50, 51$	Conjunction	VG1, VG2
52	Adverb	BW
53	Interjections	TSW
$54 \ldots 65$	Adjective	ADJ1, ADJ2, . . ., ADJ12
$66 \ldots 68$	Preposition	VZ1, VZ2, VZ3
$69, 70$	Numeral	TW1, TW2
71	Punctuation	LET
72	Special	SPEC

Table 2: Tags in the medium-sized tag set of size 72. The items in the second column are the main classes and correspond to the reduced tag set of 12 tags.

2 Design of ANN based tagger

In order to design a NN for PoS tagging several issues have to be resolved. First the input features for each word have to be determined and also the window size. In this paper we will focus on NN's with look ahead, that is features of words succeeding the word under consideration are used to classify the word.

2.1 Determining the input features

In conclusion, for every word we use at least the relative frequencies (prior distribution) of PoS tags for that word, of course these relative frequencies are based on the training set. Moreover in the training phase for the words in the window preceding the current word, also the actual PoS tag is used. Of course the test phase the predicted tag is used. For all preliminary tests in order to determine the parameters of the neural network we used set0, set1, set2 and set4 as training set and set10 as validation set. Each set consists of around 100,000 sentences and around 900,000 words. Each training and validation was repeated tree times.

2.1.1 Coding of unknown words

One problem with relative frequencies (prior distribution of tags) based on the training set is the occurrence of so-called unknown words (words that are in the test set but not in the training set). One option to overcome this problem is to use equal priors, meaning that each component of the feature vector gets the value $1/72$, since there are 72 possible tags. We used 10-fold cross validation on the training set to compute the relative frequencies and estimate prior probabilities of unknown words. The results can be found in Table 3. Given the fact that there are 63 tags with relative frequencies less than 2% we decided not to take into account these tags. Hence every unknown word during testing is coded by the normalized values determined by Table 3. It should be remarked that the tags N3 and N5 do not occur very often in the whole corpus, less than 2%, but they are important tags for unknown words.

Tag nr.	**Tag**	**Rel. Frequency**	**Tag nr.**	**Tag**	**Rel. Frequency**
1	N1	33.0%	15	WW7	2.5%
3	N3	14.0%	54	ADJ1	4.0%
5	N5	10.0%	62	ADJ9	2.4%
9	WW1	2.2%	72	SPEC	19.0%
12	WW4	2.3%			

Table 3: The relative frequency (prior probability) of unknown words based on a 10-fold cross validation. Only the relative frequencies with value at least 2% are listed.

2.2 Determining the window size

From the research of Marques and Lopes [7] and Schmid [10] one can conclude that a window size of 3 (tags of) words back and 2 words ahead could be reasonable for the input. We constructed and validated 7 Neural Networks with different window sizes, all with a hidden layer consisting of 50 neurons. The validation results can be found in Table 4. The average performances (over 3 runs) all vary around 96% on the validation set (set10). This set consists of 912,660 words (samples). This leads to an estimate of the 95% confidence interval of $\pm 0.022\%$. Hence we can conclude that a 3×2, a 3×3 and 4×3 ($b \times a$ means b words back and a words ahead) window perform the best.

	Window size						
	2x2	2x3	3x2	3x3	3x4	4x3	4x4
run1	96.52	96.66	96.53	96.64	96.59	96.62	96.52
run2	96.54	96.51	96.64	96.57	96.53	96.56	96.55
run3	96.45	96.61	96.66	96.55	96.56	96.65	96.63
average	**96.503**	**96.593**	**96.610**	**96.587**	**96.560**	**96.610**	**96.567**

Table 4: The performance in % for different window sizes, $b \times a$ means a window size of b tags back and a tags ahead. Each neural network has a hidden layer of 50 neurons.

2.3 Determining the size of the hidden layer

In order to determine the optimal number of hidden neurons we used the smallest window size, 3×2, of the previous subsection. Once again the different networks where trained 3 times on the union of set1, set2, set3 and set4 and validated on set10. The average results can be found in Table 5. From Table 5 we can conclude that the Neural Networks with 250 and 370 hidden neurons are the two best performing ones.

2.4 Determining the optimal configuration

In order to determine the best configuration we combine the results of the previous two subsections, window size 2×3 or 3×2 or 4×3 and number of hidden neurons 250 or 370. This results in 6 different configuration,

	Number of hidden neurons									
	50	90	130	170	210	250	290	330	370	410
Av. Perf.	96.600	96.653	96.697	96.680	96.777	96.787	96.693	96.767	96.803	96.707

Table 5: The performance for different sizes of the hidden layer for an input window of 3×2. Once again the 95% confidence interval is $\pm 0.022\%$.

cf. Table 6, which were compared using the same procedure as described above. Given the results of Table 6 we conclude that a Neural Network with a window of 4 words back, 3 words ahead and 370 hidden neurons is the winner of the test. This network will be used for the evaluation.

	Window size		
Hidden neurons	2x3	3x2	4x3
250	96.677	96.787	96.743
370	96.663	96.803	**96.830**

Table 6: The average performance of the different configurations over 3 runs. Once again the 95% confidence interval is $\pm 0.022\%$.

3 Evaluation

In the evaluation we used set1 up to and including set9 as training set. We will train the Neural Network configuration – window of 4 words back, 3 words ahead and 370 hidden neurons – several times and use set10 for validation. The best performing Neural Network on set10 will be used for testing on set0. Due to the enormous size of the training set, approximately 8 million words, batch learning is not feasible. Hence a different online learning approach is taken. In the first iteration a linear scan through the training set is done. In the following iteration a randomized walk through the training set is used. In every step of this walk a random next line, uniform between 1 and 8, for training is selected. This is repeated 800,000 times in one training epoch. The results on the validation set, set10, are depicted in Table 7. From this table it can be concluded that the Neural Network after the 4th epoch performed best. This neural network will be used for evaluation of the ultimate test set; set0.

	Training epoch								
	1	2	3	4	5	6	7	8	9
Performance on set10	97.13	97.25	97.22	**97.36**	97.34	97.33	97.35	97.33	97.33

Table 7: The performance of the Neural Network with a window of 4 words back, 3 words ahead and 370 hidden neurons for the different training epochs.

3.1 Overall performance on the test set

The trained Neural Network reached a performance of 97.35% on the test, compared to a performance of 91.86% of the baseline tagger. The baseline tagger assigns to every word the most likely class based on the class probabilities estimated from the data in the training set.

A detailed evaluation of the performance of the Neural Network can be found in Table 8. In this table the performance of the Neural Network on the different categories is compared with the performance of the baseline tagger. From this table it can be concluded that there is a correlation between the size of the category and the performance; the larger the size, the better the performance. Moreover the Neural Network outperforms the baseline tagger with at least 4.5% on each category.

Category	**Words**	**NN**	**Baseline Tagger**
Face to face conversations	82069	97.34%	92.36%
Interviews with teacher Dutch	58307	97.16%	91.65%
Phone dialogue (recorded at platform)	72443	97.93%	93.42%
Phone dialogue (recorded with mini disc)	75706	97.96%	93.31%
Business conversations	13409	97.79%	92.45%
Interview and discussions record from radio	78890	96.96%	90.28%
Political debates and discussions	35341	96.55%	89.52%
Lectures	41032	97.36%	91.40%
Sport comments	21816	97.24%	90.78%
Discussions on current events	18183	97.22%	90.45%
News	35819	96.81%	90.94%
Comments (radio, TV)	13988	96.35%	89.23%
Masses, ceremonies	1762	96.20%	89.10%
Lectures, discourses	13244	96.07%	89.14%
Texts, read aloud	89502	96.31%	89.39%
Average result		**97.35%**	**91.86%**

Table 8: The performance of the trained NN for the different categories of the test set; set0. For comparison the performance results for the baseline tagger are depicted in the last column.

True	**Predicted**	**Errors**	**Known words**	**Unknown words**
N3	N1	1177	37	1140
N5	N1	823	111	712
Spec	N1	901	109	792
N5	Spec	601	533	68
WW2	WW4	1458	1458	0
WW4	WW2	976	976	0
VZ1	VZ2	857	875	0
VZ2	VZ1	700	700	0

Table 9: The most significant absolute confusions between the different tags and the contributions from the known and unknown words.

Since we have 72 tags, we will not give the total confusion matrix but only the most significant confusions, cf. Table 9. It follows from Table 9 that there is a large confusion between N1, N3, N5 and Spec. Moreover most of these confusions are due to the unknown words. One of the reasons for this confusion for unknown words could be due to the coding of unknown words, cf. Table 3. The most likely tag of an unknown word is N1.

On the other hand the confusions for the other tags are due to the known words. For the known words there is a bilateral confusion between WW2 and WW4. These tags have very low relative frequencies for the unknown words. Hence these tags are almost never the desired tag of an unknown word. The overall performance of the designed PoS tagger on the known words in the test set is 97.88% and most errors are due to the confusion between the tags WW2 – WW4 (contribution of 0.27% to the error) and VZ1 – VZ2 (contribution of 0.17% to the error). On the unknown words the performance of the NN based PoST is 41.67% and the confusion between N3 and N1 contributes 13.29% to the error. The confusion N5 – N1 and Spec – N1 both contribute around 9% to the error. We also tested our approach with equal prior probabilities for the unknown words but this resulted in a performance of 28.51% on the unknown words and a similar performance on known words. Hence our approach of coding unknown words boosted the performance on unknown words by almost 13%.

3.2 Comparison

The following table, Table 10, gives a comparison of different taggers on the CNG corpus[1].

Technique	ALL	Known	Unknown	Comments
NN (this paper)	97.4	97.9	41.7	
SVM [9]	97.5	97.3	70.0	(a)
TnT [11]	97.3	97.5	96.0	(b)
Brill [11]	96.1	97.1	94.4	(c)
Zavrel [13]	95.4			(d)
Canisius [4]	91.9			(d)

Table 10: Performance of different PosT on the CNG

Some comments are in place, the * refer to entries in Table 10:

(a) The SVM tagger uses the same 72 tagset and the same training and test data. Moreover it uses compound analysis for unknown. We expect that NN outperforms SVM if it uses this also. A higher tagging speed makes NN more practical than the SVM tagger.

(b) The TnT tagger uses the same 72 tagset and the same training and test data as the NN and SVM taggers. Build with the TnT tagger of T. Brants, [2].

(c) Uses the Brill tagger, [3]. Same medium sized tag set but trained on 100.000 sentences only. Training on the full training set is not doable.

(d) These results on the CGN corpus are hardly comparable because of different training size and different tag sets.

4 Conclusions

In this paper we designed a Neural Network for Part-of-Speech tagging on Dutch corpora. More specifically we used the Corpus Gesproken Nederlands (CNG) for the design of the Neural Network. The Neural Network uses a sliding window of 4 words back and 3 words ahead. The hidden layer consists of 370 neurons. The input features for the Neural Network are based on a literature study and resulted in a relative frequency coding for the word under consideration and words ahead and the Part-of-Speech tag for the words back.

[1] We are grateful to Herman Stehouwer for his contribution in comparing the results of the NN tagger with those obtained with the TnT and Brill tagger.

Special attention is paid to the coding of unknown words; this is based on the relative frequencies determined by a 10-fold cross validation on the training set.

This design resulted in a performance of 97.35% (Table 8) with a 95% confidence interval of ±0.02%. A more detailed analysis showed a performance of 97.88% on known words and 41.67% on unknown words. This performance is comparable to state of the art PoST on the CNG corpus, cf. Section 3.2.

From the analysis of the confusion matrix it followed that large confusions are either totally due to the known words or totally due to the unknown words (Table 9). The developed coding of unknown words based on the relative frequencies for unknown words based on a 10-fold cross validation shows an improvement of almost 13% with respect to a coding with equal priors.

Since many words in Dutch are compounds it is our opinion that still more improvement can be gained by a finer coding on unknown words based on the compound analysis of the unknown word.

References

[1] T. Brants. TnT – A statistical part-of-speech tagger. In *Proceedings of the 6th Applied NLP Conference (ANLP-2000)*, 2000.

[2] Thorsten Brants. Tnt: A statistical part of speech tagger. In *Proceedings of the 6th applied NLP conference, ANLP-2000*, 2000.

[3] Eric Brill. Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[4] S. Canisius and A. van den Bosch. A memory-based shallow parser for spoken dutch. ILK/Computational Linguistics and AI, Tilburg University, 2004.

[5] Walter Daelemans, Jakub Zavrel, Peter Berck, and Steven Gillis. Mbt: A memory-based part of speech tagger-generator. In *Proceedings of the 4th Workshop on Very Large Corpora, ACL SIGDAT*, 2000.

[6] T. G. Dietterich. Machine learning for sequential data: A review. In T. Caelli, A. Amin, R. P. W. Duin, M. S. Kamel, and D. de Ridder, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 2396 of *Lecture Notes in Computer Science*, pages 15–30. Springer, 2002.

[7] N.C. Marques and G.P. Lopes. Tagging with small training corpora. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis*, volume 2189 of *Lecture Notes in Computer Science*, pages 63 – 72, 2001.

[8] N. Oostdijk, W. Goedertier, F. Van Eynde, L. Boves, J.-P. Martens, M. Moortgat, and H. Baayen. Experiences from the spoken dutch corpus project. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*, pages 340 – 347, 2002.

[9] M. Poel, L. Stegeman, and H.J.A. op den Akker. A support vector machine approach to dutch part-of-speech tagging. In M.R. Berthold, J. Shawe-Taylor, and N. Lavrac, editors, *Advances in Intelligent Data Analysis VII. Proceedings of the 7th International Symposium on Intelligent Data Analysis, IDA 2007*, volume 4723 of *Lecture Notes in Computer Science*, pages 274–283. Springer Verlag, 2007.

[10] H. Schmid. Part-of-speech tagging with neural networks. In *Proceedings of the 15th Conference on Computer Linguistics*, volume 1, pages 172 – 176, 1994.

[11] J.H. Stehouwer. Comparing a tbl tagger with an hmm tagger: time efficiency, accuracy, unknown words. Internal report, Dept. Computer Science, University of Twente, 2006.

[12] F. van Eynde. Part of speech tagging en lemmatisering. Technical report, Centrum voor Computerlinguïstiek, K.U. Leuven, 2000.

[13] J. Zavrel and W. Daelemans. Bootstrapping a tagged corpus through combination of existing heterogeneous taggers. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*, 2002.