Application of Artificial Neural Networks in Morphological Tagging of Czech

Petr Němec [*]
nemec@ufal.mff.cuni.cz

Abstract: Morphological tagging is an important problem in the area of computational linguistics as it underlies other crucial tasks such as syntactic parsing and machine translation. Nowadays, the problem is being most commonly solved by a statistical approach. Artificial neural networks (ANN) represent another promising approach to this kind of problems for which the exact algorithmic solution is unknown or not efficient enough. In this paper we present the results obtained by application of the well-known backpropagation (BP) neural network in several types of experiments. We have focused on the Czech morphology, because its morphological system is very rich and no experiments concerning application of artificial neural networks have been carried out for this language. First, we have verified on a set of preliminary experiments that the neural network is capable of capturing the Czech morphology, which, secondly, served also for determination of appropriate network and context parameters. Thirdly, we have used neural networks for a voting experiment. The aim of the voting experiment was to select the correct tag (if present) from the outputs of two statistical taggers, the Markov model tagger and the Feature-based tagger. In this experiment, BP showed higher tagging precision (93,56%) than any of the input statistical methods (92,74%, 92,58%) and exceeded even the currently best available statistical result (93,47%). BP has proved to be a worthy post-processing tool that is able to perform implicit evaluation on complementary aspects of different statistical approaches.

Keywords: neural networks, tagging, morphology

1 Introduction

Morphological tagging is an important problem in the area of computational linguistics as it underlies other crucial tasks such as syntactic parsing and machine translation. The task is especially interesting for highly inflectional languages whose morphological system is very rich. An excellent example are Slavonic languages, namely Czech, which is one of the morphologically most complex languages. Nowadays, the problem is being most commonly solved by a statistical approach.

Artificial neural networks(ANN) represent another promising approach to this kind of problems for which the exact algorithmic solution is unknown or not efficient enough. When presented a sufficiently representative training set of problem instances, they are able to adapt in such way that they produce correct results even for the instances that were not trained, i.e. they are able to generalize over the trained data.

Many statistical methods have already dealt with morphological tagging and a relatively high tagging precision has been obtained. On the contrary, there were just a few attempts to apply ANN for this task.

[*] Center for Computational Linguistics, MFF UK, Malostranské náměstí 25, 11800 Praha

For the first time for Czech, we describe in this paper experiments that test the neural networks approach performance on morphological tagging for this language.

Section 2 introduces Czech morphology and explains the terms used in this paper. Section 3 describes the experiment data and the electronic sources we used. Section 4 lists the results of statistical approaches and explains their role in our experiments. Section 5 presents the background of ANN experiments for this area and specifies our approach. Section 6 describes our preliminary experiments whose aim was to develop a problem representation model and to determine the optimal BP ANN configuration to be used in the final voting experiment, which is described in Section 7. Section 8 summarizes the achieved results and Section 9 points out possible future work.

2 On Czech morphology

Czech language recognizes ten part-of-speech types, five of which are inflective: nouns, adjectives, pronouns, numerals, and verbs. For each of them there is usually a large set of inflectional and conjugational patterns. There are common morphological categories present: gender, number, case, person, tense, grade, voice. This information including other special attributes form a complex system whose configuration for a given word form is expressed by a positional 15 character[1] morphological tag.

E.g., `NNFP1-----A----` stands for plural (position 4) feminine (position 3) noun (position 1) in nominative (position 5). There are more than 2000 possible tags for Czech word forms. A morphological analysis tool that determines for (almost) each Czech word a set of possible morphological tags is available, so the morphological tagging device - a tagger - only selects the proper tag of the word from this restricted set. On detailed description of Czech morphological tags together with the morphological analysis tool see [2].

3 Experiment data

All the experiments have been performed on the Prague Dependency Treebank (PDT) morphological data [2] containing both the result of the morphological analysis and the tagging results of two statistical taggers (Feature-based tagger and Markov model tagger, see Section 4).

The training set contains approximately 1.5 million words. Two disjoint testing sets (both containing approximately 100 000 words) are used: The development testing set serves the purpose of experiment parameters fine-tuning - we attempt to obtain the optimal tagging performance on this set. The corresponding experiment configuration is then tested on the evaluation testing set and the obtained figures are then reported, i.e., it is not allowed to optimize the results directly for the evaluation set. This way the relevance of the reported figures is ensured.

Because the tagging result is always a single tag, we measure the performance of a tagger by the tagging accuracy defined as the number of correctly tagged words divided by the number of total tagged words.

[1] The tag represents 13 categories - positions 13 and 14 are unused.

4 Statistical approach

Statistics represent nowadays a classic approach to the morphological tagging problem. The tagging accuracy of the currently best statistical taggers for Czech (the Feature-based tagger, the Markov model tagger, and an advanced statistical tagger CZ031219 [1]) is listed in Table 1. The CZ031219 tagger is the currently best statistical tagger trained on PDT data[2]. We list the tagging accuracy for the selected three morphological categories (gender, number, case), because they show the lowest success rates and therefore substantially determine the overall tagging accuracy[3]. Apart from the entire tag determination we will also focus on the sole determination of these categories.

The output of these statistical taggers plays a significant role in our experiments. Not only we compare the achieved results to them, but their output also serves as a ”context provider” as described in Section 6.1 and is used for voting experiment in Section 7. We aim to make use of these methods and improve their tagging accuracy.

Tagger	Total	Gender	Number	Case
CZ031219	93.47	97.82	98.00	95.37
Feature-based	92.74	97.55	97.62	94.67
Markov model	92.58	97.62	97.86	94.41

Tab. 1: Tagging accuracy of statistical taggers

5 ANN approach

Compared to other approaches, there are not many current experiments where ANN are applied to Natural Language Processing (NLP) tasks. However, ANN experiments regarding morphological tagging have been performed for English. Schmid [5] trained recurrent multilayer perceptron network as part-of-speech morphological tagger. Using wide left and right context together with suffix information, he was able to obtain results similar to those reached by statistical methods (Hidden Markov model systems). Nakamura et al. [4] used a massive feed-forward network to predict the part-of-speech of a word on the basis of its left context. Also in this case, the tagging precision was approximately the same as that of a trigram-based predictor. To our best knowledge, no same neither similar ANN experiments have been carried out for Czech.

Towards our ANN experiments, we have also considered using BP-SOM architecture [7], because BP-SOM learning algorithm showed very good performance over some classical computational problems. For example, it outperformed the classical BP learning algorithm in the binary vector parity classification task and monks tasks [6]. BP-SOM network can be used only as a classifier (each output vector represents a class), where the number of classes need not be very high as each class maps to one neuron in the output

[2] These figures are obtained when the taggers are trained on a subset of the entire training set. As the precise results for the entire set are yet unavailable, we could not compare our results to them nor could we use them in our experiments to improve their performance.

[3] Tagging accuracy for the remaining 10 categories is close to 100%.

layer). For the purpose of experimenting with Czech morphology there is a need of statistical merging (see Section 6.3), which causes that the resulting output vectors do not represent a class anymore, but they form a potentially infinite set of general real number vectors. Without statistical merging, the training set would be not only inconsistent, but also very large. We have therefore decided to abandon BP-SOM approach.

A classical option that overcomes the output vector limitation is a feed-forward network trained using the well known BP learning algorithm, see e.g. [3]. We have therefore tested its capabilities on the morphological tagging task.

6 Preliminary experiments

The purpose of our preliminary experiments is to build a problem representation model for BP ANN and to test the capability of the network to determine a correct tag (or value of a particular category) of a word. Furthermore, the designed model is then used in the voting experiment, see Section 7.

6.1 Experiment model

The experiment model is the standard n-gram model. BP ANN has to determine the correct tag (or category value) of a word on the basis of its suffix and $n-1$ immediate preceding tags, where n is a fixed constant). Formally, let t_i be a tag of a word w_i. In each sentence

$$s = (w_1, w_2, ..., w_k)$$

and $n \leq k$ we consider

$$(t_1, t_2, ..., t_{n-1})$$
$$(t_2, t_3, ..., t_n)$$
$$...$$
$$(t_{k-n+1}, t_{k-n+2}, ..., t_{k-1})$$

as (left) contexts of $w_n, w_{n+1}, ..., w_k$, respectively[4].

We suggest two types of preliminary experiments. First, we set the preceding tags to be the correct ones (from PDT, i.e. the left context is always disambiguated[5]) Second, we set them to be the tags produced beforehand by a statistical tagger. Using an output of such tagger, which in this way becomes a context provider, removes the mentioned limitation on cost of the context reliability. We have used Feature-based tagger as the context provider, because its tagging accuracy is higher than that of Markov model tagger[6].

The output is the correct category value of the tagged words $w_n, w_{n+1}, ..., w_k$ respectively.

BP ANN is trained on a set of input/output vector pairs (I, O). Now, the purpose is to construct a suitable representation of these vectors taking into account also the suffix[7] of the word whose tag is to be determined. We have tried several ways of coding of I and O. We present here only the one that was shown to be optimal.

[4] We extend the sentence by adding virtual empty tags at its beginning in order to guarantee adequate contexts for every word in the sentence.

[5] This way, of course, it is not possible to tag a continual text directly in a practical application, because we cannot guarantee correct preceding tags, so a single mistake in tagging would lead to context information collapse.

[6] CZ031219 could not be used as its tagging results on the training set are not available.

[7] In Czech it is the suffix that bears substantial morphological information

6.2 Representation of input

Let $n \geq 1$, $m \geq 0$ be fixed integers (experiment parameters), let k be the number of characters (length) of w_n. Let w_n be preceded by $n-1$ known tags $T = (t_1, t_2, ..., t_{n-1})$. Let $S = (l_n^1, l_n^2, ..., l_n^r)$ be the suffix of w_n, where, with respect to w_n, l_n^1 represents its last character, l_n^2 its second character to the last, and so forth. (If $r < m$ for a suffix length m put $l_n^i = \lambda$ (empty word) for $i > r$). We define the context of w_n to be the concatenation of $(t_1, t_2, ..., t_{n-1})$ and $(l_n^1, l_n^2, ..., l_n^m)$. Therefore, the context of w_n is

$$I = (t_1, t_2, ..., t_{n-1}, l_n^1, l_n^2, ..., l_n^m).$$

Each single tag from T (or single character from S) is coded separately with a fixed code width. The linear string of these codes will form the code for T (or S) as shown in Table 2.

Code of T				**Code of S**			
$C_t(t_1)$	$C_t(t_2)$	...	$C_t(t_{n-1})$	$C_l(l_n^1)$	$C_l(l_n^2)$	...	$C_l(l_n^m)$

Tab. 2: Input vector code

Every character appearing in the corpus is assigned an ordinal number index. The code $C_l(l_i)$ of the character l_i is this index. In a similar manner, we code the tag as a concatenation of the codes of its 13 morphological positions: the code $C_c(a_j)$ for a particular category value a_j is its ordinal index in the set of all values for that category[8], see Table 3.

Tag code C_t			
$C_c(a_1)$	$C_c(a_2)$	...	$C_c(a_{13})$

Tab. 3: Tag code

In this way we assure that each index resides in a part of the vector of fixed width (length) equal for all indices and as long as necessary to store the highest index of the given set. The indices are binary coded. The coding functions are bijective, hence individual network weights can be trained within their specific positions.

6.3 Representation of output

The output vector O includes possible (category value) candidate outputs for I as well as their frequencies (see below).

The simplest way to code a single instance of an output for a specific category in a given context (therefore, pointing out a value from a list of possible values) is to use "one from n" coding. The vector length equals to the number of possible output values,

[8] Experiments show that the exact linear order of the respective components is not very important.

where, in its basic variant, all vector components are set to 0, except for the position of the correct output, which is set to 1.

For the cases when there are more instances with equal context but different output, it is possible to merge all such training instances into a single output vector, where the position of the value with maximal frequency of occurrence is set to 1, while all other vector positions are set to

$$\Theta + \frac{(1 - \Theta) * V_j}{V_{max}}$$

where V_j is the frequency of occurrence of the value corresponding to the output vector position j, V_{max} is the frequency of the winning value (maximal), and Θ is a reliability threshold constant.

This approach makes the training set consistent and also substantially reduces the training set size.

6.4 Results

The best preliminary results (as listed in Table 4) were obtained using a left context of length 2 and suffix length 4. BP ANN showed optimal performance with hidden layer of 300 neurons, learning rate of 0.2 and momentum of 0.7. These parameters were tuned by performing number of experiments on the development testing set (see Section 2). However, the result values showed to be stable and small changes of these parameters did not have any significant effect on them. It has been observed in almost all experiments that the performance on the testing files usually reached its maximum very soon, before 200-th cycle. Although the tagging accurancy of the preliminary experiments is lower than that of the statistics[9], it stays high above the random baselines (91,71%;85,4%;74,33% for the three categories respectively) encouraging us to perform the voting experiment described in the next section.

Context	Total	Gender	Number	Case
Disambiguated	89,22	94.11	96.67	92.71
Statistical	88,71	94.51	97.12	94.46

Tab. 4: Results of preliminary experiments

7 Voting experiment

Having verified that BP ANN is capable of capturing Czech morphological information and having determined suitable BP ANN parameters, the aim of the voting experiment will be a determination of the correct tag from outputs of existent statistical taggers. The voting has been done on two statistical taggers, the Feature-based tagger (FB) and the Markov model tagger (MM). The complementary rates of these taggers are CR(FB,MM) = 41,86% and CR(MM,FB) = 43,06%.

[9] Our method uses less information that statistical taggers as it does not consider the right context which statistics does implicitly by using Viterby algorithm.

7.1 Experiment model

The input of the voting experiment (coded as described earlier in this paper) consists, in its basic variant, of: left statistical context, (two) candidate statistical tags and suffix of for the given word. Additionally, a right statistical context (tags appearing right of the word determined beforehand by the statistical tagger) was added. Right context tags were coded in the same way as the left context tags. The order of these elements in the input vector has been: Left context — Right context — Candidate tags — Suffix[10]

The size of the output vector has been set equal to the number of candidate tags (i.e. 2 in our case) and each component represents whether the corresponding candidate tag is the correct tag. If so, it is set to 1, otherwise it is set to 0. This raw training set is then statistically merged as described in the previous section, so the output vector values represent the frequency value of the given statistical tag to be the correct tag in the given context.

7.2 Results

Firstly, we have measured the baseline results obtained by selecting a random statistical output. To assure higher baseline objectivity, the random test has been run 20 times and the obtained values have been averaged. The obtained baseline value of 92.69% is less than the sole output of the better of the two statistical methods (see Table 1).

Left Context	**Right Context**	**Suffix Length**	**Precision**
1	0	0	**93.56**
1	0	2	93.52
1	0	4	93,48
1	1	4	93,51
2	1	4	93,47

Tab. 5: Voting experiment results

As in the case of preliminary experiments, we have selected the (better) Feature-based tagger to be the context provider. Therefore, the left and the right contexts were created from the Feature-based tagger output.

The neural network parameters were set with respect to the values described in the previous section, i.e. the learning rate takes values from the interval [0.1, 0.2], and the momentum takes values from the interval [0.6, 0.8]. The exact value of these parameters was set random for each experiment, and each experiment ran at least 500 cycles within a single iteration. The best obtained results for various context and suffix lengths are listed in Table 5. The number units for left and right contexts represent number of tags, while the number units for suffix length represent number of characters. It is surprising that the experiment that took into account the least information showed the highest tagging accuracy. This may be due to the excessive size of the input vector if more information is provided or perhaps this information is not so relevant for determining the complementary aspects of the two taggers.

[10] Again, the exact linear order of these components does not seem to be important.

8 Conclusion

We have used the BP ANN in several types of experiments. When determining the correct morphological tag directly, we have learned that the neural network is basically capable to handle the problem. We have managed to determine appropriate network and context parameters, which we have used in a voting experiment. For the voting experiment, the BP ANN showed higher tagging precision (93,56%) than any of the input statistical methods (92.74%, 92.58%). Our tagging precision is even higher that the best available statistical result (93.47%). [11] The presented results show that the union of the statistical and neural network approach is very promising and that it is worth to perform various ANN experiments, especially for the purpose of complementary connections of various taggers. Therefore BP ANN is also suitable for contrastive evaluation of different taggers.

9 Future work

The necessary task is to perform the described tests with statistical parsers trained on the entire training set and to compare the results.

We believe that there are also other experiments worth performing. For instance, recurrent neural networks performance on the morphological tagging task should be tested. As a part of other methods for the purpose of implementation of dynamical contexts, neural networks could be trained to determine the optimal context parameters for the tagging task.

10 Acknowledgement

This work was supported by the Center of Excellence, Center for Computational Linguistics, project number LN00A063 of the Czech Ministry of Education.

Bibliography

1. Jan Hajič. *Disambiguation of Rich Inflection (Computational Morphology of Czech).* MFF UK, 2002, 334.
2. Jan Hajič, Eva Hajičová, Petr Pajas, Jarmila Panevová, Petr Sgall, Barbora Vidová-Hladká. *Prague Dependency Treebank 1.0. CDROM.* CAT:LDC2001T0., ISBN 1-58563-212-0, 2001.
3. T. Mitchell. *Machine learning.* McGraw-Hill, 1997.
4. M. Nakamura, K. Maruyama, T.Kawabata, K. Shikano. *Neural network approach to word category prediction for English texts.* Proceeding of COLING-90, 1990, 213-218.
5. Helmut Schmid. *Part-of-Speech Tagging with Neural Networks.* Proceeding of COLING-94, 1994, 172-176.
6. S.B. Thrun. *The MONK's Problems: a performance comparison of different learning algorithms.* Technical Report CMU-CS-91-197, 1991, Carnegie Mellon University.
7. Ton Weijters. *The BP-SOM architecture and learning rule.* Neural Processing Letters, 1995, 13-16.

[11] We were unable to perform the tests with statistical parsers trained on the entire training set as these data were not available at the time this paper is written.