

# A Neural Network Based Morphological Analyser of the Natural Language

Piotr Jędrzejowicz<sup>1</sup> and Jakub Strychowski<sup>2</sup>

<sup>1</sup> Chair of Information Systems, Gdynia Maritime University, Gdynia, Poland

<sup>2</sup> Rodan Systems S.A., Sopot, Poland

**Abstract.** The paper proposes a morphological analyser supported by a neural network to inflect words written in Polish. The approach can be also applied to other languages. The main task of the analyser is to create base forms from the analysed words' forms. Other objective is to provide grammatical information for the analysed form. Computational experiment results confirm that both objectives are fulfilled by the proposed neural network based morphological analyser. The common words are inflected with a very high quality of nearly 99.9%. Other words like geographical names and people's names, thanks to the incorporation of neural network, inflect with a quality reaching 93.3%.

## 1 Introduction

A natural language processing consists of succeeding steps of a speech and text analysis. Every step of this process (i.e. speech recognition, tokenization, morphological analysis, parsing, semantic knowledge mining) causes some errors. A quality of a final product depends on cumulative influence of all analysis elements, therefore it is very important to make every step of the NLP as reliable and effective as possible.

A morphological analysis, one of the NLP elements, means processing word forms without considering context and has fundamentally two objectives [7]:

- to obtain the grammatical information from a single word form,
- to assign the word form to its base form.

The quality of morphology analysis depends on the processed language [4]. For example, the morphology analysis of words in English is not very difficult, because English words contain little grammatical information. Unfortunately, morphology analysis must be expanded to the part-of-speech tagging process for this language. POS (Part Of Speech) tagger determines a grammatical function of the word through analysing the context of the word in the sentence [1]. Other languages may have rich inflection, and a morphological analyser development could be very difficult in such cases. Reward for this is the grammatical information obtained during morphology analysis process. For example in the Polish language, by examining only a single form of the word, one can determine tense, part of speech, case, number etc.

Such information could be very helpful in the successive steps of the natural language processing task [4].

Morphological analysers are usually constructed using one of the following approaches [15]:

- A dictionary approach, where the main part of the analyser is a dictionary containing all words' forms in a given language. Analysis of a given word is reduced to the searching for the matching form in the dictionary, and to reading grammatical information from this dictionary [16] [2] [11].
- An algorithmic approach, where analyser contains a set of the inflection rules. Appropriate rules are being applied to a given word to obtain its base form. Grammatical information depends on the chosen rules [5] [14] [3] [10].

The first approach ensures high quality of the analysis, but only for the forms which exists in the dictionary. Its main weaknesses include a substantial memory requirements and labour-consuming need for the dictionary development. The most important advantage of the second approach is its ability to analyse forms which do not occur in the dictionary. This is an advantage for the Polish language which could have geographical names, and peoples' names written in many inflected forms. Weakness of this approach are lower quality and a problem with developing a good set of rules [8].

The paper proposes a new solution integrating both of the above approaches. A full dictionary of the Polish language is used as a training set. During the training process applicable inflection patterns (similar to the inflection rules) are developed. A decision tree helps to assign appropriate inflection pattern to the given word's form. The tree is developed through the affixes analysis during the training. The focus of the research was to show that an artificial neural network can increase analyser quality through reinforcing abilities of the decision tree. While it is possible to achieve near 99.9% accuracy using only inflection patterns and a small dictionary containing all base forms of the words in a given language, for the unknown words a decision tree and a neural network should be used to gain high quality.

The following section describes an architecture, training process, and working rules of the morphological analyser. In the next section a neural network approach, improving analyser abilities, is suggested. To verify the approach a computational experiment was carried. Its results are, subsequently, presented and discussed. The final section contains conclusions and ideas for future research.

## 2 The Morphological Analyser

One of the possible methods of carrying word morphology analysis is an assignment of a given word's form to a valid inflection pattern. An inflection pattern consists of the set of affixes describing how to create specified form from a word's root. An affix is a chunk of a word which can occur before (prefixes), after (suffixes) and even inside (infixes) word's root. Each word can be constructed from a single root and from many affixes. To make things simplified, a group of affixes constructing specified form is called a *supplement* in this article. So, the inflection pattern consists of the supplements.

For example Polish word *dziadek* (*grandfather* in English) can be written in the following forms:

dziade**ek**, dziad**ka**, dziad**kowi**, dziad**kiem**, dziad**ku**, dziad**ki**, dziad**ków**,  
dziad**kom**, dziad**kami**, dziad**kach**

As we can see this word contains root *dziad-*, and can have following suffixes:

**-ek, -ka, -kowi, -kiem, -ku, -ków, -kom, -kami, -kach.**

Such a set of suffixes can be treated as an inflection pattern. An another Polish word - *szybko* (adverb *quick* in English) - can be written in the following forms:

szyb**ko**, szyb**ciej**, naj**szybciej**

The forms of this word have root: *-szyb-*; suffixes: *-ko, -ciej*; and prefix *naj-*. An inflection pattern valid for this word can be written as the following set of supplements:

**-ko, -ciej, naj-ciej**

Each supplement in the inflection pattern is described by the grammatical information. One of these supplements is marked as a base form. So, the full information about inflection pattern for the Polish word *szybko* can be written in the following form:

**-ko** [baseform + adverb], **-ciej** [adverb + comparative], **naj-ciej** [adverb + superlative]

All possible inflection patterns are stored in the inflection patterns base.

A simple way to make morphological analysis is to find a valid inflection pattern for the analysed form. An assignment of a given word's form to a valid supplement stored in the correct inflection pattern allows to describe word's form by the grammatical information, and allows to create any word's form (specially the base form). Creation of any possible form can be useful in a text generation task [7].

The above method seems to be quite simple to apply, but unfortunately, it hides many difficulties. One of the problems is a creation of the inflection patterns. There are inflection patterns developed for many languages by the linguists, but these patterns are too general and cannot be used in the computational linguistics in many cases [6]. Another problem is the required performance of an analyser. A search for the matching pattern could be very time-consuming especially when a number of patterns is high (i.e. greater than 1000). Nevertheless, the main problem for the analyser is to remove ambiguity. It turns out that supplement analysis can assign a single word's form to many inflection patterns. For example the supplement *\*a* can be assigned to most of the inflection patterns in the Polish language (wildcard *\** means any sequence of characters here). So, an analyser needs to assign the word's form to a valid inflection pattern (in some situations an assignment to many patterns is also correct).

The first problem - creation of the inflection patterns - is solved using a full dictionary of a given language. Such dictionary contains all possible forms of the frequently used words. It is possible to determine root and supplements of a given word having all its forms. Obtained supplements, sorted in the order proper for the part-of-speech, create an inflection pattern. Words inflected in the same way create a common inflection pattern. A set of inflection patterns

consist of all different patterns. The grammatical information for all patterns can be easily added by a linguist.

A total number of inflection patterns depends on the language and a quality of the dictionary used as a training set. A set of inflection patterns can contain hundreds or even thousands of elements. The morphology analysis could be very time-consuming if it is applied to a huge collection of documents. Hence, it is important to optimize searching in the inflection patterns' set. This can be done by storing inflection patterns as a decision tree.

Successive characters of the supplements (starting from the supplement's end) are stored as the nodes of the tree. The nodes attached to the tree's root are related to the last characters in the supplements. The nodes from the next levels are related to the succeeding characters. Wildcart character (\*) is also stored as a node. The leaves of the tree relate to the inflection patterns.

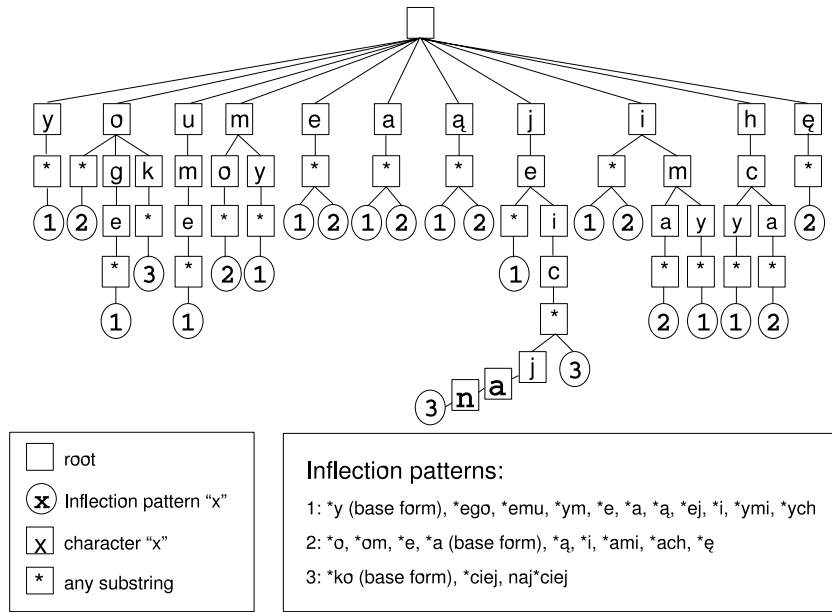


Fig. 1. An example decision tree of the morphological analyser

A decision tree created from the inflection patterns related to the Polish words *arktyczny* (*arctic* in English), *bakteria* (*bacteria* in English) and *szybko* is shown in the figure 1.

Analysis of the word's form is based on traversing a tree starting from its root and reaching its leaves through all possible pathes. On each node, a decision about traversing to the valid sub-nodes is made. Using a terminology from the machine learning domain the edges between nodes represent decisions, and the leaves represent all possible hypotheses. Finding inflection patterns is a simple classification task. An important feature of the "tree solution" is its high speed. Assignment of a single inflection pattern to the analysed form takes time which can be compared to the time used during iteration through the characters from the form's end to the form's begin.

The decision tree allows to find inflection patterns having supplements which match the analysed form. Unfortunately, it is quite often that some patterns towards which the decision tree is pointing should not be assigned to

a particular form. The problem is to choose a valid pattern from all possible candidates returned by the decision tree.

For example, the decision tree shown in the figure 1 assigns inflection patterns 1 and 2 to the forms *arktyczne* and *bakterie*. For the first form only inflection pattern 1 is correct, and for the second form only inflection pattern 2 is correct. So, the supplement *\*e*, which is used in the discussed case, does not identify valid inflection patterns.

Some experiments carried out with the analyser showed that the length of the matched supplement is an essential factor - the probability of making a good choice is greater for the longer supplements. Possible and useful solution is to reduce the number of candidates by extending the decision tree.

The roots of the words inflecting in the same way have many similarities. Quite often last letters are identical in such roots. It is than possible to extend the tree, replacing a leaf by a set of nodes which cover all possible roots connected to this leaf. Each node, added in the place of the leaf, represents a substring which is created from common suffixes of the roots. Such extended tree produces fewer candidates and has a better quality, but still it is difficult to assure high quality of the morphological analysis for the Polish language.

### 3 The Neural Network Approach

As we mentioned in the proceeding paragraph the decision tree can return many candidates causing ambiguity. It is possible to assign probability to each leaf in the decision tree, and then select inflection pattern related to the leaf with the highest probability. Such a solution has major weakness: many inflection patterns can be dominated by the patterns with higher probability. In our view a better solution is to use an artificial neural network as a classifier selecting a valid inflection pattern from all the candidates.

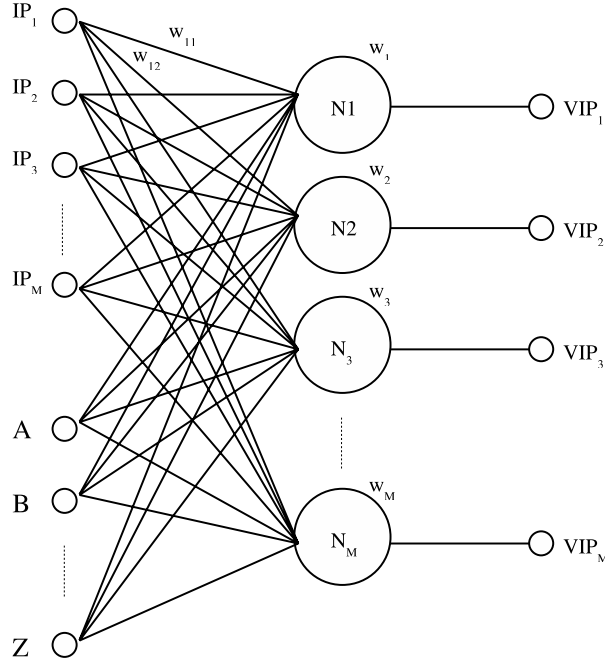
A neural network consists of neurons, which can have many inputs. Each input is multiplied by its corresponding weight and the product of this multiplication is fed into the body of the neuron. The neuron adds up all the products. The weighted sum of the products is usually denoted as net in the neural network literature [9]. Finally, the neuron computes its output as a certain function of net. This function is called the activation or transfer function. The same output value produced by the activation function can be sent out through multiple edges emerging from the neuron. Some neurons are also connected to the input of the neural network. Similarly, the outputs of some neurons represent the output of the network.

In most cases layered architecture of the neural network is used. Each layer consist of separate neurons. Each input of the layer is connected to all neurons in the layer. Neurons' outputs stand for layer output. Succeeding layers are connected in the serialized form.

Having set up the architecture, the neural network can be trained. The learning task of the neural network is to adjust the weights so that it can output the target signal for each input signal. The neural network is trained by a learning algorithm, which modifies weights of the network during presentation of example pairs of input and target output signals. Deeper and more advanced analysis of the neural networks can be found, for example, in [9], [12].

In the morphology analysis, neural network selects the valid inflection pattern from all the candidates returned by a decision tree. The selection can

be performed by a neural network constructed from a single layer of neurons. The inputs of this layer points to the inflection patterns, which are stimulated by the decision tree. Each output of the layer points to the target inflection pattern. Decision tree generates a list of candidates and stimulates the neural network and the network produces the output signals. Signals for all inputs pointing to the patterns selected by the decision tree are set to  $1$ . Remaining input signals are set to  $0$ . A neuron with the highest output value designates valid inflection pattern for the analysed form.



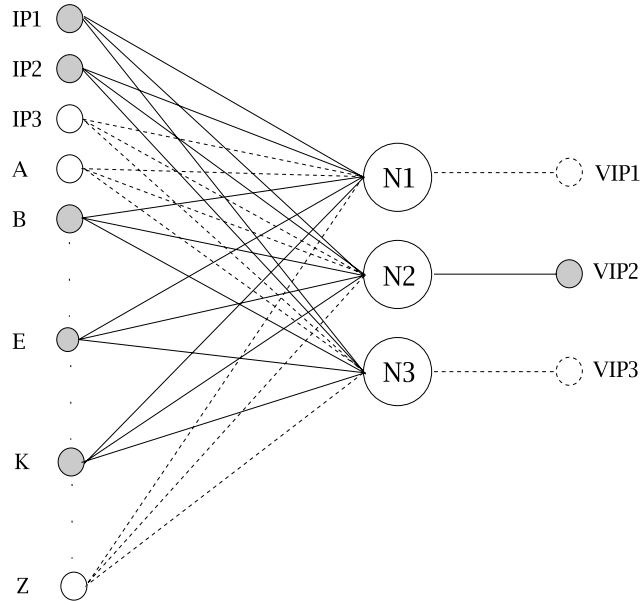
**Fig. 2.** Architecture of the proposed neural network

In some situations, it would be impossible to determine a valid inflection pattern having only list of candidates as an input, particularly if a different choice should be made for the same lists of candidates. The problem is resolved by adding inputs in form of the alphabetic letters. If a letter occurs in the analysed form then corresponding input is activated (set to  $1$ ). This provides enough information to make an appropriate choice.

Architecture of the described neural network is presented in the figure 2. The input signals related to the inflection patterns are denoted as  $IP_1 \dots IP_M$  ( $IP$ – inflection pattern,  $M$ – number of all patterns). Symbols from "A" to "Z" stand for input signals related to the letters which can occur in any word in a given language. Neurons are denoted as  $N_1 \dots N_M$ , and  $w_1 \dots w_M$  indicate weights of neurons. Symbol  $w_{ij}$  indicates weight of the connection between an input  $i$ , and a neuron  $j$ . An output signal related to the inflection pattern  $VIP_x$  (Valued Inflection Pattern number  $x$ ) is calculated using a following activation function:

$$VIP_x = w_x \cdot \left( \left( \sum_{i=1}^M IP_i \cdot w_{ix} \right) + \left( \sum_{l="A"}^{"Z"} l \cdot w_{lx} \right) \right) \quad (1)$$

A neural network training was based on applying a back-propagation algorithm [12] to a training set. In the discussed case the training set includes all succeeding words' forms from the full dictionary.



**Fig. 3.** Neural network analysing the Polish word “bakterie”

The figure 3 shows an example neural network stimulated during an analysis of the Polish word *bakterie*. In this case the decision tree returns 2 candidates: inflection patterns 1 and 2. Values of the related neural network's inputs are set to 1. Also inputs related to the letters which occurs in the word *bakterie* are stimulated. Each of the neurons  $N_1$ ,  $N_2$ ,  $N_3$  produces output signal. Output denoted as  $VIP_2$  has the highest value so the second inflection pattern stands for the output of the analysis.

## 4 Computational experiment results

The quality of the proposed morphological analyser was evaluated through analysis of all word's forms available in the full dictionary of the Polish language. The analysis of any word is correct if this word is converted to the valid base form. This condition is fulfilled if a valid inflection pattern is selected. The quality measure is a ratio between a number of correctly inflected words and a number of all analysed forms. The quality calculated in such way is weighted down by the error rate of the dictionary.

For the Polish language only 58,3% of all available forms (about 2 500 000) are inflected correctly if only simple decision tree is used. Usage of the

extended tree increases the quality measure to 77%. Usage of a dictionary of the base forms (which contains about 100 000 words) allow to inflect with quality near 99.9%. Such result can be achieved only for the common Polish words. The words which base forms aren't presented in the dictionary are inflected with a quality near 93.3%, which is achieved by a neural network based analyser.

Replacing a back-propagation algorithm as a training tool by a genetic algorithm [17] allows to obtain a quality of the analysis reaching 78%. Thus, the back propagation algorithms seems to perform better. Nevertheless it is worthwhile to possibly test more advanced training algorithms with a view to further improving a quality of the morphological analyser.

The described morphological analyser was implemented in a Java language. The tests have been carried on a PC computer with AMD Athlon XP 1900+ CPU, and 512MB RAM on the board. Analyser has been trained and tested using Sun JVM 1.4.2 for Linux. Table 1 shows experiment results.

**Table 1.** Summary of the experiment results

Property	Value
Number of the training words	126263
Number of the all words' forms	<b>2450612</b>
Number of the supplements	13967
Number of the inflection patterns	1589
Number of frequently used inflection patterns	<b>73</b>
Inflection patterns determination time	2h 02m 51s
Extended decision tree construction time (extended tree)	36s
ANN training time (back-propagation, single iteration)	1h 12m 33s
Total construction time	3h 17m
Memory allocated by the inflection patterns	445 kB
Memory allocated by the extended decision tree	1245 kB
Memory allocated by the base forms dictionary	950 kB
Memory allocated by the exceptions dictionary	707 kB
Memory allocated by the ANN	468 kB
Total memory allocated by the morphological analyser	<b>4261 kB</b>
Analyser initialization time	<b>345 ms</b>
Analyser speed (only decision tree)	7391 words/s
Analyser speed (decision tree + ANN)	814 words/s
Analyser speed (decision tree + base forms dictionary + ANN)	<b>4244 words/s</b>
Analyser quality (only decision tree)	77.41 %
Analyser quality (decision tree + ANN)	<b>93.30 %</b>
Analyser quality (decision tree + base forms dictionary + ANN)	<b>99.95 %</b>

## 5 Conclusions

The presented morphological analyser was developed as a part of the ICONS project [13]. Its version is used in a full-text categorisation tool as a stemmer. The analyser is also integrated with the Lucene, an open source full-text search engine, and allows to search words written in the Polish language. The proposed solution provides a very high quality of the inflection analysis for the common words. Using a neural network based analyser with the extended decision tree increases quality of analysis from 58.3% to 93.3% for words not available in the dictionary.

The proposed morphological analyser is fully functional but it seems still possible to increase quality of the analysis. One of the possible approaches is using a better training dictionary containing less errors and more training examples. Some errors can be removed by the inflection patterns analysis. For example, many of the inflection patterns related to less than 10 words were created because of the errors in the dictionary. It is possible to attain a good inflection analysis for the words related with such inflection patterns, by improving a dictionary and reducing a number of inflection patterns.

Another possibility for improving the analyser is upgrading inflection patterns. It is possible to create a diminutive or augmentative forms adding special affixes to the Polish nouns. The number of well inflected forms can be doubled by adding to the inflection patterns the supplements adequate to these forms. Also supplements for the derivatives (i.e. noun created from the verb) can be added.

The morphological analyser should work better for the frequently used words in the real applications. A good approach is to include a distribution of the words' occurrence in the training process. This could increase the quality of the analysis of the real documents.

Often a sentence written in the Polish language contains specific characters (i.e. ą ł ó ź ć ń). Sometimes in place of these characters corresponding Latin letters are used (i.e ą -> a, ł -> l, ó -> o). The inflection patterns can be extended or modified to work with the documents written in such way. Similarly, some spelling errors can be supported.

Back-propagation algorithm trains neural network with good results. Nevertheless, other learning algorithms or network's architectures can be tried to reach higher quality.

## References

1. Allen J. (1995) Natural Language Understanding. - Redwood City. CA: Benjamin Cummings
2. Dubisz S. ed. (2003) Uniwersalny słownik języka polskiego. - Wydawnictwo Naukowe PWN
3. Frakes W.B., Baeza-Yates, R. (1992) Stemming Algorithms. Information Retrieval - Data Structures and Algorithms. 131-160 . Prentice Hall, London
4. Hajic J. (2000) Morphological Tagging: Data vs. Dictionaries. - In Proceedings of ANLP-NAACL Conference, 94-101. Seattle, Washington, USA
5. Hull D. (1996) Stemming algorithms: A case study for detailed evaluation. - Journal of the American Society for Information Science, 47(1), 70-84.
6. Jagodziński G. (2004) Gramatyka Języka Polskiego. - <http://grzegorz.w.interia.pl/gram/isopl/gram1.html>
7. Jurafsky D., Martin J.H. (2000) Speech and Language Processing: An Introduction to Natural Language Processing. - Computational Linguistics and Speech Recognition. Prentice Hall
8. Karttunen L., Beesley K.R. (2001) A Short History of Two-Level Morphology. - Presented at the ESSLLI-2001 Special Event titled "Twenty Years of Finite-State Morphology." Helsinki, Finland
9. Korbicz J., Obuchowicz A., Uciński D. (1994) Sztuczne Sieci Neuronowe. Podstawy i Zastosowania. - Akademicka Oficyna Wydawnicza PLJ
10. Porter M. (1980) An algorithm, for suffix stripping. - Program.
11. Prywata M., Gackiewicz P., Macewicz W. (2004) Polish dictionary for `aspell`, `ispell`, `myspell`. - <http://www.kurnik.pl/slownik> <http://ispell-pl.sourceforge.net/>

12. Rutkowska D., Piliński M., Rutkowski L. (1999) Sieci neuronowe, algorytmy genetyczne i systemy rozmyte. - Wydawnictwo Naukowe PWN.
13. Staniszek E., Nowicki B. (2004) ICONS based Knowledge Management in the Process of Structural Funds Projects Preparation. - Accepted for the e-Challenges, e-2004 Conference. Vienna, Austria
14. Szafran K. (1993) Automatyczna analiza fleksyjna tekstu polskiego. - Rozprawa doktorska, Wydział Polonistyki UW. Warszawa
15. Viks (1994) A morphological analyzer for the Estonian language: the possibilities and impossibilities of automatic analysis. - Automatic Morphology of Estonian 1: 7-28
16. Weiss D. (2002) Polski lematyzator. - <http://www.cs.put.poznan.pl/dweiss/>
17. Xin Yao (1999) Evolving Artificial Neural Networks. - Proceedings of the IEEE, vol 87, No 9