Application of Neural Networks for POS Tagging and Intonation Control in Speech Synthesis for Polish

Artur Janicki
Warsaw University of Technology, Institute of Telecommunications,
Division of Teletransmission Systems
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
email:ajanicki@elka.pw.edu.pl

Abstract - **The paper describes use of neural networks in POS (part-of-speech) tagging and intonation control, needed in a speech synthesis system for the Polish language. Feedforward multilayered perceptrons have been proposed for both purposes. Considerations during planning the network architecture, used training data, training process and verification of the results are described.**

I. Introduction and problem description

Control of intonation in TTS (text-to-speech) systems is one of the most difficult tasks in speech synthesis. It has a significant impact on listening comfort.

When generating a synthetic speech, first task is to generate a signal intelligible to the listener. But soon after it is achieved, the next task is to make the speech sound as natural as possible. If this is missing, the speech makes the listener tired and discourages him from using such a TTS system. To achieve a naturally-sounding speech we need a careful control of so called prosodic parameters, i.e. duration, intonation, pausing, rhythm, energy etc., of which intonation is one of the most important.

Controlling the intonation means generating a proper F0 function, i.e. function of changes of fundamental frequency, corresponding to a given sequence of words. The task is not trivial at all [9], because the intonation depends on the meaning of a phrase and also carries the information about emotions of the speaker (anger, surprise, excitation etc.). So there is a need of some mechanism which would retrieve at least some basic information from a text in natural language (in this case: in Polish) and generate upon it a natural-like F0 contour.

It is possible that more than one F0 contour for a given sentence will sound natural to the listener; on the other hand we should strongly avoid the situation when the system uses e.g. an incorrect accent and lets the listener realize, that the TTS system does not really understand what it is saying.

II. Proposed approach

It has been proposed to correlate the F0 contour with sequence of POS (part-of-speech) tags, corresponding to words in a phrase. POS type, i.e. information if a given word is a noun, a verb, an adjective or other, is definitely related to a role which a given word plays in the sentence and is somehow related to its meaning. Similar approaches were successful for other languages [3]. Carrying emotions is beyond scope of this work – neutral emotion will we applied instead.

To use a system for controlling intonation basing on POS-tags sequence, we need to be able to get know the POS type information of given words, i.e. we need a POS tagger, in this case for Polish.

For both purposes, i.e. for POS tagging and for generating F0 contours, neural networks have been proposed, namely multilayered perceptrons – MLP [8]. Neural networks were already used before in POS tagging, e.g. in POS disambiguation for English [10], however the proposed approach of POS recognition does not use a lexicon at all. The details are described in the following chapters.

III. Part of Speech tagging

A. *Proposed neural network architecture*

A neural network to perform POS tagging is expected to have 15 binary outputs, corresponding to 15 POS tags, which it is supposed to recognize (see Table 1). The question of what to take as an input requires more attention. First, it has been decided to ignore the case information. It may come in useful in the future: it can be helpful in disambiguation, detection of proper names etc., but for the time being it has been decided to ignore it, in order not to make the architecture too complex.

The next question was: do we need to analyze a whole word? The more characters we have at the input, the more precise we are, but on the other hand the network looses its ability to generalize its answers. So if a new word comes, e.g. a neologism or an inflected form of a word, the network would likely fail. The smaller number of characters we take, the more we are exposed to ambiguities, but the network becomes more “wise”, because it learns some rules, instead of learning the words by heart. In this case we need also a smaller amount of the training data. To sum up, we need to search for a compromise and most likely cut some of the longer words.

The consequent question was how to cut the words: whether to take some characters from the end or from the beginning or from both sides, and how many characters to consider.

Table 1. Set of POS tags to be recognized by a POS tagger.

Tag	Name	E.g. (Polish)	(English)
A	preposition	*o, nad*	*about, over*
C	conjunction	*i, lecz*	*and, but*
D	adverb	*mocno*	*firmly*
E	interjection	*ejże, no*	*eh, hey*
I	verb – infinitive	*iść*	*to go*
J	adjective	*mały, stare*	*small, old*
M	numeral	*cztery*	*four*
N	noun	*domek*	*house*
P	possessive pronoun	*mój, ich*	*my, their*
R	particle	*się, by, nie*	*oneself, not*
T	indicative pronoun	*ten, tamten*	*this, that*
U	adverbial participle	*jadąc*	*going*
W	WH-quest pronouns	*kto, gdzie*	*who, where*
V	verb (other forms)	*idzie*	*goes*
Z	personal pronoun	*ja, oni*	*I, they*

The Polish language is morphologically rich, like other Slavic languages, unlike e.g. English. According to Polish morphology rules the suffix plays significant role as for the word's POS type, whilst the prefix usually modifies the meaning, not changing the POS type. See example below, showing different words derived from the nucleus "prac-" ("work-"):

- prac**a**, prac**ownia,** prac**y**, prac**ę**, prac**ą**, prac**e**, prac, prac**om**, prac**ami** – are all nouns (*a work, a studio* etc.),
- prac**ować**, **prze**prac**ować**, **do**prac**ować**, **na**prac**ować**, **o**prac**ować**, **od**prac**ować** – are all verbs in infinitive form (*to work, to overstrain, to polish up, to toil* etc.),
- prac**owy**, prac**owego**, prac**owemu**, prac**owym**, prac**owych**, prac**owymi** – are all adjectives (*work-related* etc.), etc.

The problem is that the same endings can signalize a different POS for different words. So we have to let the network analyze longer part of a word, so that the network has a chance to realize which word it is dealing with.

Experiments were conducted to check how many letters from end of the word have to be analyzed for the best performance of a POS tagger. In the annotated text containing more than 40,000 forms of Polish words, analyzed were last 2, 3 and more characters. Then a search was performed to check if all words having this ending belong to one POS category, or if this ending is ambiguous. For example words:

- **ziel**ony ("*green"*, J - adjective, masc. sing.)
- **ż**ony ("*wife*", N -noun, genit. sing. or nomin. pl.)

belong to different POS categories. The network will be able to find that difference if it analyzes 4 or more last characters, while using only 3 characters or less would result in an ambiguity. It turned out that although 4 characters would be sufficient in this example, for the whole text corpora it would give still almost 5% of ambiguities (see Figure 1). Adding next characters improves the situation significantly, e.g. for 8 characters we achieve the level of number of ambiguities below 100 (what gives 0.2% of total corpus size). Analyzing entire words would result in 74 ambiguities, which are impossible to distinguish by simple analysis of characters forming those words. Disambiguating those words would require incorporating additional techniques (analyzing context etc.), what is beyond the scope of this work.

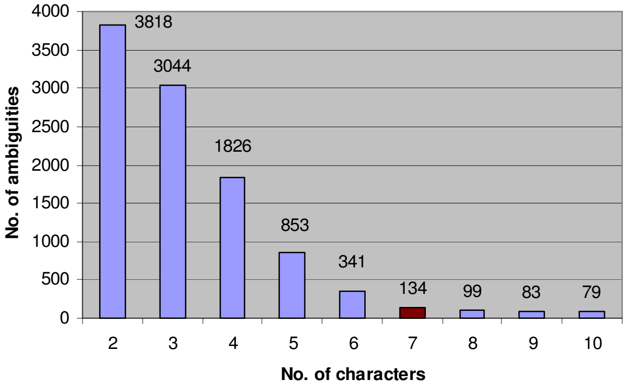

Figure 1. Number of POS ambiguities versus the number of analyzed characters from the end of a word.

Finally it has been decided to analyze last 7 characters of the word, what gives a satisfactory level of ambiguity (ca. 0.3%) and possibly will enable the network to work correctly for words outside of the training set, using its ability of generalizing.

As for the ambiguous forms, the one which seemed more frequent has been left in the training set. The list of ambiguous words has been created, to be used in possible future works.

At each input we can expect one of 35 characters of the Polish alphabet or a space, if the word is shorter than 7 letters. That gives total number of 252 binary inputs to the network.

It has been decided to use simple feedforward 2-layers neural network. The output layer will consist of 15 neurons, later it will be decided what number of neurons in the input layer is sufficient to solve this problem. Sigmoidal function will be used as the activation function.

Figure 2 shows the proposed architecture of the neural network for the POS tagger. The answer of the net is set to be pointed by the output with the highest value, if it is above 0.5. If none of them is above 0.5, "don't know" answer is given.

B. *Network training*

To train a neural network for a POS tagger an existing corpus called WKSF has been used. WKSF stands for *Wzbogacony Korpus Słownika Frekwencyjnego Współczesnej Polszczyzny* (Enriched Corpus for Frequentative Dictionary of Contemporary Polish) and has been prepared in 2001 by Janusz S. Bień and Marcin Woliński from University of Warsaw [2]. The corpus contained 500 000 words taken from different newspapers, popular scientific texts, prose and drama texts, coming from years 1963-67. This corpus was meant to serve as a subject for researches on concordances and word frequency. It is well annotated with rich morphological information, so this enabled the author to use it as a corpus to develop and test a POS tagger.

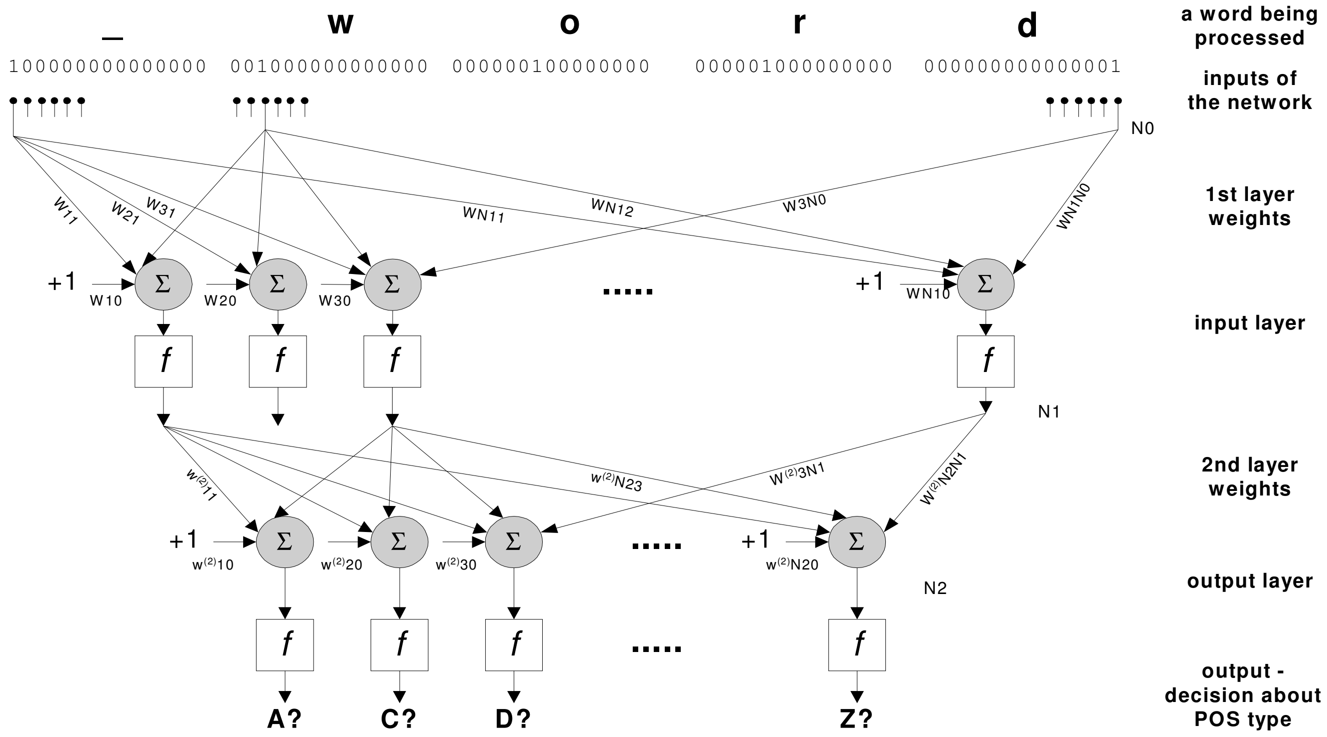

Figure 2. Schematic picture of the neural network for POS tagging (here for 5 characters as the input).

The words together with related POS information were extracted from the WKSF corpus; tags used to annotate morphological forms in the corpora were mapped to those needed for the POS tagger. Repeated forms were omitted.

The disadvantage of the WKSF corpus was that it was missing many useful words, which came into being after year 1967. So it needed adding several updates, to reflect changes in the language. Many words have been added, mainly technical and computer-related, but also many words from colloquial language. The amended words were manually annotated with POS tags.

The network was initialized with random weights. Supervised learning was used, i.e. output of the network was compared with the desired value. MBP (Momentum Back Propagation) technique was used to train the network, i.e. weights were being adjusted having computed the maximum error gradient, but with a certain momentum, to avoid being trapped in local minima.

Learning coefficients were initially set to 0.05, but later increased for wrong answers. Momentum was set to 0.3. Additional learning sessions were run using words which the network had problems with. The situation was quite parallel to the one from real life – when a student of a foreign language has problems with some words, he is exposed to those words by a teacher over and over, until he memorizes them, whilst he repeats other, already learned words, less intensively. Every 20-40,000 iterations the network was examined to check, which words it already recognizes correctly and which it still fails to recognize.

The learning process was started with just 10 neurons in the input layer; then higher numbers were used, to check if the network converges to any minimum and what level of correctness corresponds to that minimum.

One can see in Figure 3 that obviously 10 neurons are not sufficient and the net behaves instable. However when increasing the number, we do achieve better stability, but the overall result is still not satisfactory: after 5 million of iterations it reaches ca. 95% and stabilizes. Noticeable is that having increased the number of neurons 4 times, from 50 to 200, we get only a slight improve in correctness: from 94% to 95%.

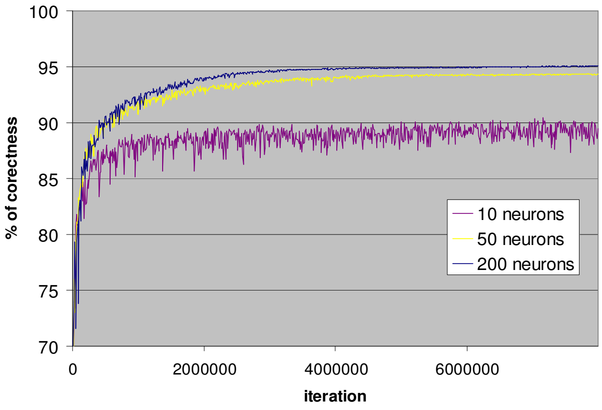

Figure 3. Percentage of correctness of POS recognition against the iteration number of the tagger neural network learning process.

The relatively poor performance of such network was probably caused by shorter words. Shorter words introduce “spaces” at the beginning, so they practically switch the initial inputs off, setting them to constant values. What's more, the mechanism of recognizing shorter words is often different from the longer ones – shorter words more often are irregular, so the network is rather learning them by heart than trying to "understand" the rules.

To avoid the impact of short words it has been decided to split the analyzed words into 2 groups: "long" words, having 6 characters or more, and the remaining, "short" words.

The experiments with adjusting the number of neurons in the input layer were re-run for those two groups and turned to perform significantly better. Starting from 10 neurons the number was increased. When the network already caught some dominant rules for POS recognition and the number of misrecognized words decreased significantly to single %, the training was being repeated over and over for the mistaken words, with increased learning coefficient for wrong answers and decreased for correct ones (down to 0.001). Attention was paid, so that the learning coefficient for wrong answers was not too high either, because this could have disturbed convergence of the learning algorithm.

Table 2. Summary of parameters and results for 2 separate networks - for "short" and "long" words.

	short words	**long words**
criterion	≤ 5 characters	> 5 characters
population of the training set	7079	49046
final # of neurons	140	170
final correctness for the training sequence	99.97%	100%

Finally the number of neurons reached 140 and 170 for short and long words respectively (see Table 2) and gave a perfect result of almost 100% of correctness for the training set, what obviously was accepted as a satisfactory result.

C. Verification of results

The performance of the POS tagger has been verified by testing using a relatively large text corpus outside of the training set, containing over 80000 words (250 of A4 text pages). The corpus contained fragments of works of Polish writers and poets: prose by Stanisław Lem and Marek Hłasko, several poems of Wisława Szymborska and Agnieszka Osiecka, who are/were using fairly conversational vocabulary in their works, also articles from the newspaper "Rzeczpospolita". In addition, stories by Stanisław Lem are usually rich in neologisms, what makes it even more difficult to deal with.

The corpus was cleaned from any non-words, including punctuation marks. The corpus contained mainly nouns (44.3% of word forms), followed by verbs (23.2%) and adjectives (21.9%). The remaining POS types all together took only slightly over 10% of different word forms.

Such a corpus was exposed to the newly constructed neural POS tagger. The results of the recognition were carefully analyzed. First it was distinguished, which words occurred already in the training set, and which were new ones to the tagger. The words repeated from the training set were of course almost 100% correct, however the words outside of the training corpus were quite numerous as for the forms – they formed almost 29% of the word forms, although not that numerous as for the occurrences (slightly over 10%) – see the summary in Table 3 for details.

Table 3. Summary of the POS tagger testing.

	words in total	**word forms**
# of words from training set	72814	11890
# of words outside of training set	8908	4826
% of words outside of training set	10.90%	28.87%
total # of words	**81722**	**16716**
# of wrong decisions	809	329
errors against # of words out of training set	9.08%	6.82%
correctness within words out of training set	90.92%	93.18%
errors against total # (%):	0.99%	1.97%
overall correctness (%)	**99.01%**	**98.03%**

The words outside of the training corpus were POS-tagged manually and the result was compared with the output from the neural network. The words outside of the training corpus consisted of more rarely used words, proper names, neologisms and commonly used words, but which occurred in a form different than in the training set, due to flexion: mainly declension or conjugation.

The POS tagger proved to perform quite well. It misrecognized or did not recognize at all 329 word forms, what makes almost 7% of the set outside of the training sequence. If counting all word forms, the error is less than 2%. When taking into account the number of occurrences of words, the result is even better, due to the fact that the words not present in the training set were usually rare. In this case the error is slightly less than 1%, what gives a very good total correctness of 99.01%.

Analyzed were also types of confusion during the POS recognition. Table 4 shows the confusion matrix. The conclusions from it are as follows:

- the most common confusion was recognizing a noun instead of a verb, it was happening vice versa also, but far less frequently,
- quite common was also confusion noun-adjective, similar in both directions, also (less numerous) confusion: noun-adverb, verb-adjective,
- when the tagger failed to recognized a POS, usually the most common wrong answer of the tagger was: "noun",
- the network gives the answer "don't know" most often for nouns and verbs.

Table 4. Confusion matrix for the POS recognition, for the words outside of the training set.

		actual result																sum
		A	C	D	E	I	J	M	N	P	R	T	U	V	W	Z	?	
requested result	A	2							2							1	1	6
	C		7						1								2	10
	D		1	68			2		8								6	85
	E				0				4								1	5
	I					254			4					2			3	263
	J		1	1			993		**30**					5			10	1040
	M						1	4	3					1				9
	N		1	10			**26**		1952					**19**		1	**16**	2025
	P									0							3	3
	R						2		2		1			3				8
	T						3					1						4
	U					1			3				66				2	72
	V		1	4			12		**109**					1144			14	1284
	W						2		1					1	0		1	5
	Z								2							5		7
sum:		2	11	83	0	255	1041	4	2121	0	1	1	66	1175	0	7	59	**4826**

IV. An approach to intonation control using neural networks

Basing on the sequence of POS-tagged words, the block of intonation control is supposed to decide about position of intonational events. Another neural network was proposed for this purpose, trained on another set of training data.

A. *Network description and training*

It has been proposed to observe the current word and its neighborhood of 2 words to the right and to the left and any punctuation marks within that distance. Only statements were taken into consideration and punctuation marks were narrowed down to full-stops, commas, semicolons and dashes.

During previous researches of the author it has been tested, that we can generate a fairly naturally sounding intonation of a Polish statement, by forming a F0 (fundamental frequency) function using 5 different intonational events: 3 boundary tones and 2 accents. This approach is similar to a simple ToBI model, of which more complex examples are given in literature ([1],[4]).

Therefore the neural network will need to have 5 x 19 binary inputs, corresponding to 5 words and both POS tags and punctuation marks, and 5 outputs, corresponding to 5 intonational events.

It was decided to use again a feedforward MLP, multilayered perceptron with 2 layers and start with 5 neurons in the input layer, as the data complexness seemed not to be high and in order to preserve the network from overtraining.

As the training data a set of manually annotated over 300 sentences was used. The sentences were first POS-tagged using the neural POS-tagger developed before, then they were manually annotated by an expert, who was marking where to put the closest intonational event. Although the amount of manual work put to annotate the sentences was huge, but still the data sparsity was a severe problem in this case. The theoretical number of all possible combination at the input would be 19^5 = 2.48 million… Not all of them really happen, but still it makes a problem. To cope with this, we focused mainly on middle 3 elements of the input vector, i.e. on the word being examined and 2 neighboring ones. From the edge 2 positions just the information if this is a word or a punctuation mark was taken into account, together with the punctuation mark type. Therefore data sparsity was less significant. Neural network's ability to generalize should do the rest.

During the training, similar parameters were taken as in case of the POS tagger. The number of neurons in the input layer was finally set to 10, because the error level was behaving erratically for lower values and the network could not be able to generalize for higher ones. Learning was stopped after ca. 500 thousands iterations, reaching ca. 96% of correctness for the training set. Unlike as in POS tagger, the training data contained contradict entries; it has been left for the network to decide which answer to choose in such ambiguous cases. This is also why none of the networks would give 100% correctness for such a training data.

B. *Verifying results*

Unlike when testing the POS tagger, in this case it was impossible to judge objectively the correctness of the network. This is because even not placing e.g. the same accent in a place in a prosodic phrase can still result in a good impression of a listener. E.g. both sentences presented in Figure 4 sound naturally, although there is slight difference in accent locations.

This is why listening tests were performed to verify, if the intonation generated using this neural network sounds naturally and is comparable with deterministic, rule-based methods. ElanSpeech's POLVOC™ system [7], was used to implement intonation, generated by the neural network. The tests were performed using Pair Comparison method [6]. 96 students took part in them, they were exposed to 60 pairs of synthesized sentences, with "neural" intonation, deterministic one using Chinks 'n Chunks algorithm [7] and some other.

The detailed results are presented in [5], for purpose of this paper let us note, that 57% of listeners chose "neural" intonation against Chinks 'n Chunks, saying that it sounds more natural. So it is not only comparable with deterministic methods, but it is quite well competing with them.

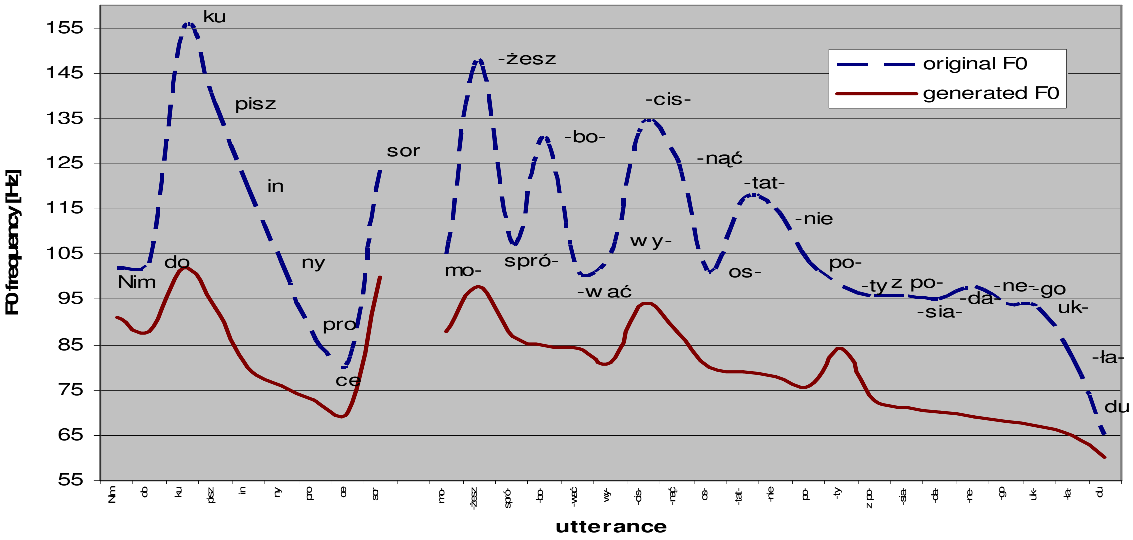

Figure 4. Comparison of original F0 and F0 generated by the neural network.

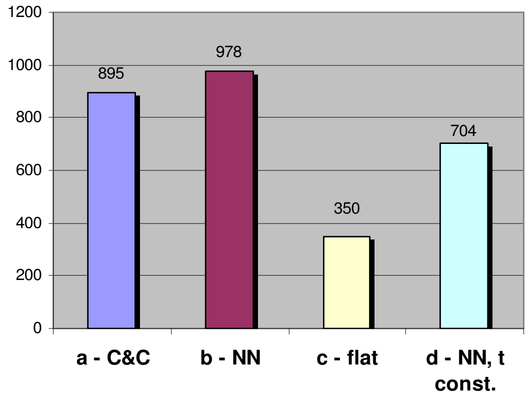

Figure 5. Comparison of scores for different variants of synthetic speech. a) Chinks 'n Chunks, b) using this neural network, c) no accents, d) neural networks, but limited control of timings [5].

V. Conclusions

To sum up, neural networks proved to be very useful also in Natural Language Processing (NLP) applications, namely in POS tagging and intonation modeling for the Polish speech synthesis system. POS tagger built on a feedforward MLP (actually on 2 separate MLP's, for short and long words), turned out to be very successful. The tests using texts outside of the training set were POS-tagged by this network with the error lower than 1%, what is considered as a good result.

Also intonation generated by another network, described above in this paper, performed quite well. In this case the network was deciding about location of intonational events. Formal listening tests run on 96 listeners showed that naturalness of such speech is not only comparable with deterministic methods, but it tends to prevail. It is planned to run tests on longer fragments of speech, where it is quite likely that the difference will be more distinct.

Both networks can be easily re-trained for other languages, on condition that the required corpora are at disposal. The number of inputs of the POS tagger may need to be adjusted, depending on the alphabet size for a given language.

As for future works on the Polish language, it is planned that the POS tagger will be further developed to recognize more detailed POS types or even grammatical forms. Also the "intonational" neural network in the future can use wider inventory of intonational events, to model the F0 contour more precisely.

References

[1] S.Bauman, M.Grice, R.Benzmüller. GToBI – A phonological system for the transcription of German intonation, *Prosody 2000*, Kraków, 2000.

[2] J.Bień, M.Woliński. Numeryczne kody gramatyczne we wzbogaconym korpusie Słownika frekwencyjnego polszczyzny współczesnej (in Polish), CD published by Warsaw University, 2001.

[3] J.Buhmann, H.Vereecken, J.Fackrell, J.P.Martens,B.VanCoile. Data driven intonation modelling of 6 languages, ISCLP 2000, Beijing, 2000. http://chardonnay.elis.rug.ac.be/papers/2000_0002.pdf

[4] S.Godjevac. Serbo-Croatian ToBI (SC ToBI), University of California, San Diego, 2001, http://ling.ohio-state.edu/people/Alumni/godjevac

[5] A.Janicki, S.Kula. Badanie wpływu modelowania intonacji na jakość mowy syntetyzowanej z tekstu (in Polish), *Krajowe Sympozjum Telekomunikacji*, Bydgoszcz 2004.

[6] V.Kraft, T.Portele. Quality Evaluation of Five German Speech Synthesis Systems, *Acta Acustica 3*, pp. 351-365, 1995.

[7] S.Kula, P.Dymarski, A.Janicki, C.Jobin, P.Boula de Mareüil. Prosody control in diphone-based speech synthesis system for Polish, *Prosody 2000*, Kraków, 2000.

[8] A.Lapedes, R.Farber. How Neural Networks Work. *Neural Information Processing Systems*. American Institute of Physics, pp. 442-456, 1988.

[9] J.P.H.van Santen. Prosodic Modeling in Text-to-Speech Synthesis, *Proc. Eurospeech '97*, pp. KN19—28, 1997. http://citeseer.ist.psu.edu/vansanten97prosodic.html.

[10] H.Schmid. Part-of-speech tagging with neural networks. *International Conference on Computational Linguistics*, (Coling'94) pp. 172-176, Kyoto, Japan, 1994.