Automatic Prosodic Phrase Annotation in a Corpus for Speech Synthesis

Jan Romportl

Department of Cybernetics, Faculty of Applied Sciences
University of West Bohemia, Pilsen, Czech Republic
rompi@kky.zcu.cz

Abstract

In order to improve speech naturalness of a unit selection TTS system it is necessary to annotate prosodic phrase boundaries in the whole source corpus, which is extremely difficult to achieve manually. It is thus usefull to employ a machine classifier. This paper discusses suitable feature selection for such classification of a Czech TTS corpus, presents results of experiments with linear and quadratic classifiers and artificial neural networks, and compares them with human annotators.

Index Terms: speech synthesis, prosody, prosodic phrase, classification, neural network, unit selection, corpus

1. Introduction

Unit selection as a paradigm for classical concatenative speech synthesis has introduced a specific shift in methodology of text-to-speech (TTS) system design and development: many tasks in speech synthesis call for solutions based on methods rather from the area of automatic speech recognition (ASR) than from the TTS domain as it has usually been perceived. The reason for this lies in the fact that the unit selection approach relies more on fine segmental and suprasegmental description of huge speech segment databases than on techniques for signal modification of concatenated segments.

One of the important features of such a description of a speech segment database for unit selection TTS is undoubtly proper designation of *prosodic phrases* in the source corpus. A concatenation algorithm must select units with compatible suprasegmental parameterization to ensure natural prosody without disturbing phenomena, and as we have discussed elsewhere [1], position of a speech segment within its prosodic phrase is an important part of this parameterization.

Should the target cost function be enhanced by the feature of prosodic phrase position, it is necessary that the whole speech corpus be annotated with prosodic phrase boundaries. However, it is usually infeasible to perform this task manually due to vast amounts of data (often several thousands of recorded sentences in a corpus for a single voice), and therefore an automatic approach must be utilized. The one based on artificial neural networks (ANN) is presented further in this paper, on the example of a Czech male voice in the corpus of 10,000 declarative sentences for the Czech TTS system ARTIC (the corpus comprises approx. 12,000 sentences out of which 10,000 are declarative and the rest is with other modality [2]).

2. Prosodic phrase annotation

2.1. Reference data

The concept of *prosodic phrase*, as understood here, basically corresponds to a traditional phonetical view, that is such a phonetic unit which constitutes perception of the *rhythmical* qualities in language on a level higher than the lexical. A prosodic phrase is mainly delimited by acoustical features of its boundaries and it can also contain an "intonation peak". However, as Palková discusses [3], there is no empirical evidence supporting any stronger assumption about the intonation peak presence/absence or their number in a Czech sentence.

This, together with significantly less dynamical intonation of Czech in comparison with English, can lead to difficulties with objective phrase boundary designation even for human listeners. Human annotators are usually very inconsistent in judging what is and what is not a prosodic phrase boundary, and this can be overcome by utilization of a machine tagger. However, there is still a major problem: how can we obtain consistent reference data for machine learning when three different human annotators produce three different prosodic phrase annotations of a single utterance?

We have solved this issue by acquiring 100 parallel annotations of 250 sentences (randomly selected from our TTS corpus) and then using a maximum likelihood approach to estimate the objective prosodic phrase annotation – details can be found in [1]. We have worked with naive listeners (as Buhmann et al. concludes [4], naive listeners are reliable enough in similar tasks such as this one; moreover, we have achieved very similar inter-annotator agreement for Czech to what Mo et al. reports for English [5]) and in conformity with Wightman [6] we wanted them to designate places where they perceptually sensed phrase boundaries, not where they observed specific intonation events in F_0 contour in terms of the ToBI or Tilt phonologies.

This set of annotated 250 sentences is used as the reference data for machine learning and classification described further in this paper. Unlike e.g. [7] or [8], reporting on a similar task of automatic prosodic phrase detection (both in Boston University Radio News Corpus), we do not detect ToBI-based boundary tones, but strictly perceptually based events (though produced by a maximum likelihood model of an "objective listener/annotator"), whatever their acoustical and textual correlates may be.

2.2. Automatic annotation

The task of automatic prosodic phrase annotation can be reformulated into the task of machine classification whether there is or is not a prosodic phrase boundary between two adjacent prosodic words (further denoted as "left/right context"). Our overall goal is thus to set up a suitable machine classifier using the aforementioned manually annotated 250 sentences and then automatically extend prosodic phrase annotation to the rest of the 10,000-sentence corpus using this classifier.

For the sake of classification performance analysis we have created five different reference sets (further denoted as `Set1`–`Set5`) from the annotated 250 sentences: each set has 50 randomly selected sentences as the testing data while the remaining 200 sentences are used as the training data. This way we can analyse sensitivity of the classifier towards changes in input data.

Although the classification accuracy is often a good measure of classification performance, it is not of the highest importance for us. First of all there are two major types of phrase boundaries: *with* and *without a pause*. More than 99 % of intrasentential pauses are perceived as phrase boundaries [9], thus we are not much interested in such cases. Far more important for us are the cases without any pause indication (non-trivial cases). Moreover, from the overall goal of our work (i.e. such TTS corpus annotation which would eventually lead to elimination of unwanted disturbing prosodic phenomena in concatenated speech) we can infer that we are primarily interested in the *false negative rate* (FNR) of the classified *non-pause* cases – we want as few false rejections as possible, while false alarms are not that crucial. This results from the fact that if a speech unit (e.g. a diphone string) from a prosodic word realizing prosodic phrase boundary in the corpus is not labelled as being within such an intonationally functioning segment of speech, it can be then erroneously used by the concatenation algorithm in such a place where a phrase boundary is unwanted, therefore causing disturbing prosodical phenomena. The opposite case is far less problematic: it would result in surface non-realization of textually suggested phrase boundary, which would in most cases remain unnoticed by a listener. And since the average number of non-pause positives in our data is significantly smaller than non-pause negatives (approx. 17 % of the non-pause cases are phrase boundaries), FNR in non-pause cases is also the hardest criterion.

3. Features for classification

3.1. Types of features

Each word in the TTS corpus basically offers two domains of features to be parameterized by: acoustical and textual. More specifically, we can think of the following types of features:

- F_0 contour
- speech signal energy
- phone lengths
- local variability of phone lengths
- syntactical-analytical functions of two adjacent words
- parts-of-speech of two adjacent words

Relevancy of each type is often at least partially language-dependent. We have, therefore, excluded energy (it is too dependent on phone types, whereas we did not find any consistent relation with phrase boundaries, except for pre-pause cases) and parts-of-speech (phrasing, at least in Czech and other similar highly inflectional languages, is in relation with syntax, not with morphology). This general feature type selection is motivated by phonetical research, e.g. [3].

The following list shows the concrete features which were taken into account:

- **Absolute phone length** (`La`). Feature vector comprises the absolute lengths (in milliseconds) of the last three phone tokens in the left context and the first three phone tokens in the right context.
- **Relative phone length** (`Lr`). Similar to `La` but the relative length of a phone token is the ratio of its absolute length to the average length of the corresponding phone type (phone identity). If we write `LrN` where `N` is a number, it means that the last/first `N` phone tokens are taken instead of three.
- **Average length of phones in prosodic words** (`Lavg`). The feature vector comprises a value given as the sum of phone token lengths of the left context divided by the number of phone tokens in the left context. The same is calculated for the right context. If we use `La`, then `Lavg` is in absolute values, if `Lr`, then `Lavg` is in relative values.
- **Standard deviation of lengths of phones in prosodic words** (`Lstd`). Calculated analogically to `Lavg` but `Lstd` expresses overall variability of phone lengths in the left/right context.
- F_0 **contour** (`Fx`). The F_0 contour of each prosodic word in the corpus is normalized and represented by 10 equidistant values as described in [10]. The feature vector comprises `x` last values of such a representation of F_0 of the left context and `x` first values of the right context.
- **Cadence ID** (`Fcad`). Following [10], the F_0 contour of each prosodic word in the corpus is approximated by one of ten characteristic contours, so called cadences. The feature vector comprises ID of a cadence in the left/right context. `Fcad` is a categorical feature, and therefore it is coded as a vector of 0's with a 1 in the dimension corresponding to the cadence ID.
- **Analytical functors** (`AFUN`). Analytical functors represent *syntactical functions* of lexical words. The inventory of functors we have used originates from Prague Dependency Treebank 2.0. It has been slightly modified and it is listed in Table 1. Our whole corpus has been syntactically parsed using the `TectoMT` application [11] with McDonald's dependency parser yielding accuracy 85 % for Czech text. The parser assigns each lexical word an analytical functor. The feature vector for prosodic phrase boundary classification then comprises an analytical functor of the last lexical word of the left context (the context is a prosodic word which can consist of more lexical words) and the first lexical word of the right context. `AFUN` is also a categorical feature, coded analogically to `Fcad`.
- **Apriori estimation of analytical functors** (`AFUNap`). Each lexical word form can be parameterized by a vector of apriori probabilities of analytical functions this word form can appear in (e.g. $p(w = Obj) = 0.5$, $p(w = Subj) = 0.2$, etc.). Advantage of such a parameterization is that no syntactical parsing is needed – only a lexicon with word forms and probabilities, which – in our case – has been derived from data in Prague Dependency Treebank 2.0.

3.2. Feature selection

The key factor for successful machine classification is the discriminative ability of selected features which should divide different classes in a feature space as much as possible. Dichotomy classifier splits the feature space (or the space of classified vectors respectively) into two sets by a hypersurface given by a function f and parameters $\mathbf{\Theta}$. If $\mathbf{x} \in \mathrm{R}^n$ is n-dimensional classified vector, then the dichotomy classifier can be written as

$$c(\mathbf{x}, \mathbf{\Theta}) = \begin{cases} 0 \Leftrightarrow (f(\mathbf{x}, \mathbf{\Theta}) \geq 0) \\ 1 \Leftrightarrow (f(\mathbf{x}, \mathbf{\Theta}) < 0) \end{cases} . \quad (1)$$

Table 1: *List of analytical functors.*

abbrev.	description
Pred	Predicate
Sb	Subject
Obj	Object
Adv	Adverbial
Atv	Complement
Atr	Attribute
Pnom	Nominal predicate
AuxV	Auxiliary verb “be”
Coord	Coordination
Apos	Apposition
AuxTR	Reflexive tantum
AuxP	Preposition
AuxC	Conjunction
AuxOZ	Redundant or emotional item
AuxY	Adverbs and particles

Table 2: *Classification performance with various feature vectors. A – accuracy on all classified vectors, FPR.– false positive rate in non-pause cases, FNR – false negative rate in non-pause cases (the most important criterion).*

feature vector	A	FPR	FNR
AFUN	0.7214	0.0000	1.0000
AFUNap	0.9046	0.0560	0.7313
La,F10	0.8820	0.0091	0.9730
La,Fcad	0.8673	0.0182	0.9750
La,Lavg,Lstd,F3	0.8850	0.0045	0.9730
Lr,Lavg,Lstd,F3,AFUN	*0.9147*	*0.0317*	**0.5142**
Lr,Lavg,Lstd,F3,AFUNap	0.8996	0.0371	0.6041
Lr5,Lavg,Lstd,F10,AFUN	0.8968	0.0182	0.7838
Lr5,Lavg,Lstd,F5,AFUN	0.9027	0.0136	0.8108
La5,Lavg,Lstd,F5,AFUN	0.8850	0.0000	1.0000
La,F3,AFUN	0.8850	0.0045	0.9730
Lavg,Lstd,F3,AFUN	0.8820	0.0091	0.9730
Lr,Lavg,Lstd,F3	0.8968	0.0318	0.7027
F3,AFUN	0.8820	0.0091	0.9730

The value $c(\mathbf{x}, \mathbf{\Theta})$ thus assigns the vector $\mathtt{x}$ a numeric ID of a class according to its position against the hypersurface. If we want to see how well the selected features discriminate the classes, we can propose a simple suitable class of hypersurfaces:

$$f_k(\mathbf{x}, \mathbf{\Theta}) = a_0 + \sum_{i=1}^{k} \sum_{j=1}^{n} a_{i,j} \cdot x_j^i. \tag{2}$$

For $k = 1$ we have a linear classifier, for $k = 2$ quadratic and for $k = 3$ cubic. Their geometric interpretation is intuitive and they are more prone to overtraining than for example neural networks or CARTs – their performance is thus a suitable measure of how well the selected features discriminate the classes.

We have performed series of experiments with different combinations of features parameterising the left and right contexts in classification of presence/absence of prosodic phrase boundaries in Set1–Set5. The goal of each experiment was to train both linear and quadratic classifiers on training data of each Set to achieve classification performance on respective testing data as high as possible. Parameters $\mathbf{\Theta}$ of both classifiers were trained (optimized) in our system Modular using a simple genetic algorithm. In each experiment only the classifier performing better was taken into account. The feature vector which leads to the best average classification performance on testing data from Set1–Set5 (measured primarily as FNR in non-pause cases) will be used in classification experiments with ANN.

Table 2 shows the results of the classification experiments with various feature vectors. We can claim that the best results can be achieved (using the given data) with the 50-dimensional feature vector given as Lr,Lavg,Lstd,F3,AFUN.

The values in the table are averaged over Set1–Set5. The average number of classified vectors in testing data of each Set is 319.8, the average number of pauses is 69.6 and positive non-pause cases 43 (i.e. phrase boundaries not followed by a pause). Intra-sentential pauses are treated as separate prosodic words and cases with pause as the right context are also classified but no special feature indicating pause is used.

We can also see an interesting fact from Table 2: should we consider only textual features, AFUNap quite unexpectedly outperforms AFUN, but if we consider both textual and acoustical features, AFUN helps more than AFUNap. We can therefore say that without any acoustical cues it is better to know only what analytical function a word could be in rather than what it actually is.

4. ANN classification

Since the feature vector given as Lr,Lavg,Lstd,F3,AFUN has proven best, we used it further in experiments aimed at improving classification performance, primarily decreasing FNR in non-pause cases.

We have decided to use a simple fully connected feed-forward artificial neural network (ANN) with 50 units in the input layer (this number equals to the dimension of the selected feature space), one unit in the output layer (by its value in the range from 0 to 1 indicating the class), sigmoidal activation function given as

$$s(x) = \frac{1}{1 + e^{-\lambda x}}, \tag{3}$$

and a common backpropagation learning algorithm. After series of experiments with the number of hidden layers and hidden units (their numerical results are not important here) it has shown that the network with one hidden layer comprising 100 units can learn the training data very well and adding more units does not improve its performance.

As the number of training epochs reaches a specific value, ANN becomes overtrained and the classification performance on the testing data starts to degrade. It is thus vital to stop the training process just before reaching this number. However, the actual number strongly depends on the initial conditions of ANN as well as on changes in training and testing data – in other words, it has turned out that the ANN classifier is very sensitive towards changes in input data. The optimal number of training epochs and the optimal ANN initialization are thus the classifier parameters to be experimentally estimated.

The classification experiments with ANN were performed on Set1–Set5 with four different random ANN weight initializations (Init1–Init4). In each experiment ANN was trained on the training data of a particular Set and after a given number of training epochs its performance was evaluated on the testing data of the Set.

Table 3 shows accuracy, FPR and FNR averaged over Set1–Set5 for each initialization Init1–Init4. The evaluation process was performed only in selected training epochs so as to eliminate possible random unstable improvements which are specific only for the given data and would distort the evaluation of the overall ability of ANN to generalise. The table also shows the values of the Matthews Correlation Coefficient

(MC) given as

$$MC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \quad (4)$$

averaged over `Set1`–`Set5` with the given initialization and number of training epochs (TP stands for true positives, FN for false negatives, etc. – in our case all the values are only for the non-pause cases). MC is a scalar measure of "quality" of a classifier and it tries to interpret the whole two-dimensional confusion matrix in a single value. Albeit very important, an effort to decrease FNR can lead to excessive increase of FPR and thus to the classifier performance deterioration. Therefore, our aim is to decrease FNR without decreasing MC significantly.

Table 3: *ANN experiments. A – accuracy on all classified vectors, FPR – false positive rate in non-pause cases, FNR – false negative rate in non-pause cases (the most important criterion), MC – Matthews Correlation Coefficient.*

`Init 1`								
epochs	10	20	50	100	150	200	300	400
A	0.90	0.90	0.90	0.91	0.91	0.91	0.91	0.91
FPR	0.06	0.06	0.07	0.05	0.04	0.03	0.03	0.03
FNR	0.45	0.40	0.39	0.46	0.50	0.51	0.53	0.51
MC	0.54	0.56	0.55	0.55	0.55	0.55	0.53	0.55
`Init 2`								
A	0.91	0.91	0.90	0.91	0.92	0.92	0.92	0.91
FPR	0.05	0.05	0.07	0.05	0.04	0.04	0.03	0.03
FNR	0.45	0.44	0.40	0.43	0.42	0.46	0.48	0.51
MC	0.55	0.55	0.56	0.57	0.59	0.58	0.58	0.55
`Init 3`								
A	0.91	0.91	*0.91*	0.92	0.92	0.92	0.92	0.91
FPR	0.05	0.06	*0.07*	0.04	0.03	0.03	0.03	0.03
FNR	0.40	0.38	**0.36**	0.42	0.43	0.43	0.50	0.52
MC	0.59	0.58	*0.58*	0.60	0.62	0.62	0.57	0.56
`Init 4`								
A	0.91	0.90	0.90	0.91	0.91	0.92	0.91	0.91
FPR	0.06	0.07	0.07	0.04	0.03	0.04	0.03	0.03
FNR	0.41	0.40	0.39	0.43	0.47	0.45	0.52	0.53
MC	0.56	0.55	0.56	0.58	0.58	0.59	0.56	0.55

It is not important what the actual weight values for `Init1`–`Init4` are – the table shows the results for all initializations so as to allow for comparison and illustration of how sensitive ANN is towards initial conditions and that not all of them converge to the best results. We can see that `Init 3` leads to average FNR of 0.36 and average accuracy of 0.91 (with MC only slightly lower than the maximum) after 50 training epochs, hence giving the best performance. This performance estimation is quite robust because it is calculated over `Set1`–`Set5`. The worst FNR with `Init 3` after 50 epochs was 0.42, the best was 0.33.

5. Conclusions

After evaluating the experiments with ANN we can anticipate that if we parameterise each pair of adjacent prosodic words with the feature vector `Lr,Lavg,Lstd,F3,AFUN` and then we perform 50 epochs of ANN training with initialization `Init 3` and the manually annotated set of 250 sentences from our corpus, we will be able to automatically label prosodic phrase boundaries in the remaining 9,750 sentences of the corpus so that approximately 91 % of prosodic word pairs will be correctly classified in terms of a prosodic phrase boundary presence/absence, and 31 % of non-pause boundaries will be missed.

Although accuracy 91 % can be considered as plausible, 31 % of missed non-pause boundaries might seem as rather disappointing. However, firstly we must state that 69 % hit rate is significant improvement against chance level, and secondly we must point out that the performance of the classifier is as good as the best human annotators. The inter-annotator agreement on the phrase boundary placement in our reference data measured as the Fleiss' kappa among 100 annotators [1] yields $\kappa_F = 0.6636$, which means substantial agreement but still with considerable differences in phrase boundary perception. Another measure of the agreement can be the average Cohen's kappa over all pairs of annotators, for our reference data yielding $\kappa_C^{avg} = 0.6710$. If we think of the reference annotation (created by the maximum likelihood model) as being produced by a virtual "objective annotator", then we can measure the agreement of the human annotators with the reference annotation by the Cohen's kappa too – we get the average value $\kappa_{C1}^{avg} = 0.7578$ including the pause cases and $\kappa_{C2}^{avg} = 0.5488$ disregarding the pause cases. Finally, the corresponding agreement between the reference annotation and the annotation generated by the described ANN classifier yields $\kappa_{C1}^{ANN} = 0.8793$ and $\kappa_{C2}^{ANN} = 0.6635$, which means that the annotation quality of ANN is significantly above the average of the human annotators. We can just note that in terms of the Cohen's kappa only five out of 100 human annotators had higher agreement with the reference annotation than the classifier presented and tested in this paper.

6. Acknowledgements

Support for this work was provided by the Ministry of Education of the Czech Republic, project LC536.

7. References

[1] Romportl, J., "Prosodic phrases and semantic accents in speech corpus for Czech TTS synthesis", in Proc. TSD, Lecture Notes in Artificial Intelligence, 5246:493–500, 2008.

[2] Matoušek, J. and Romportl, J.,"Recording and annotation of speech corpus for Czech unit selection speech synthesis", in Proc. TSD, Lecture Notes in Artificial Intelligence, 4629:326–333, 2007.

[3] Palková, Z., "Rytmická výstavba prozaického textu (with English resume: The rhythmical potential of prose)", Academia, Prague, 1974.

[4] Buhmann, J., Caspers, J., van Heuven, V. J., Hoekstra, H., Martens, J-P. and Swerts, M., "Annotation of prominent words, prosodic boundaries and segmental lengthening by non-expert transcribers in the Spoken Dutch Corpus", in Proc. LREC, 779–785, Canary Islands, 2002.

[5] Mo, Y., Cole, J. and Lee, E-K., "Naïve listeners' prominence and boundary perception", in Proc. Speech Prosody, 735–738, Campinas, 2008.

[6] Wightman, C. W., "ToBI or not ToBI", in Proc. Speech Prosody, 25–29, Aix-en-Provence, 2002.

[7] Chen, K., Hasegawa-Johnson, M. and Cohen, A., "An automatic prosody labeling system using ANN-based syntactic-prosodic model and GMM-based acoustic-prosodic model", in Proc. ICASSP, vol. 1, 509–512, 2004.

[8] Ananthakrishnan, S. and Narayanan, S. S., "Automatic prosodic event detection using acoustic, lexical, and syntactic evidence", IEEE Trans. Audio, Speech and Language Proc., 16(1):216–228, 2008.

[9] Romportl, J., "Zvyšování přirozenosti strojově vytvářené řeči v oblasti suprasegmentálních zvukových jevů (Improving Naturalness of Machine-Generated Speech on the Suprasegmental Level)". University of West Bohemia dissertation, Pilsen, 2008.

[10] Romportl, J., "Structural data-driven prosody model for TTS synthesis", in Proc. Speech Prosody, 549–552, Dresden, 2006.

[11] Žabokrtský, Z., Ptáček, J., Pajas, P., "TectoMT: Highly modular MT system with tectogrammatics used as transfer layer", in Proc. Third Workshop on Statistical Machine Translation, 167–170, Columbus, OH, USA, 2008.