Accent Label Prediction by Time Delay Neural Networks
Using Gating Clusters

Achim F. Müller\(^{(1)}\) and Rüdiger Hoffmann\(^{(2)}\)

\(^{(1)}\) Siemens Corporate Technology, Otto-Hahn-Ring 6, D-81739 Munich, Germany
\(^{(2)}\) Dresden University of Technology, D-01062 Dresden, Germany

achim.mueller@mchp.siemens.de, kom@eakss1.et.tu-dresden.de

Abstract
In this paper a new neural network (NN) architecture for data driven prediction of accent labels—perceptual accents and pitch accents—for speech synthesis is presented. Within the proposed NN architecture, gating clusters are applied in a time delay (TD) framework. Gating clusters enable the dynamic adaptation of a network structure depending on the actual input to the NN. In the proposed TD framework, gating clusters are used to adapt the network structure such that only available input feature vectors from the actual context window are treated. The proposed NN architecture has been successfully applied for accent label prediction within our text-to-speech (TTS) system. Prediction accuracy for our German corpus was determined at 86.1%. On an English corpus the achieved accuracy was 84.5%. This result is superior to results achieved on the same corpus with an approach based on classification and regression tree (CART) techniques [1]. The results were achieved with a simpler feature set than that used in [1].

1. Introduction
In many TTS systems the complex task of prosody generation is split into two parts. In the first part symbolic prosody labels are generated. These labels are used as input to the second part that generates f0-contours. Symbolic prosody labels can be separated into two types of labels: phrase break labels and accent labels.

In this paper the prediction of accent labels will be discussed. Here the term accent label refers to labels on word level that describe either perceptual prosodic accents or pitch accents that can be derived from the f0-contour. This distinction is made, since the two data bases used in this study apply two different labeling schemes (cf. section 4).

For fast and easy adaptation to new languages and/or speakers a data driven approach is favorable. For symbolic accent label prediction prominent data driven approaches are based on classification and regression trees (CARTs) [1][2]. In [3] and [4] feed-forward NNs and tree-based learning methods have been used to predict word and syllable prominence. The mentioned approaches are based on previously fixed tree- or NN-structures. In our approach structures can be adapted depending on the actual input to the NN.

Our system works on sentence level, i.e. our goal is to generate a neutral prosody for each sentence as if the sentence was read isolated from surrounding sentences. Further, we determine context windows which are shifted across each sentence. Hereby problems occur at sentence boundaries as at the beginning of a sentence no left context is available and at the end of a sentence no right context is available. By applying gating clusters within the presented NN architecture, a dynamic adaptation of the network structure to handle the variable context information within the context windows is herein proposed. The architecture is embedded in a TDNN framework.

The paper is organized as follows: In section 2 the input parameters used for prediction are presented. Section 3 explains the NN architecture used within our accent prediction module. In section 4 the two corpora used in this study are presented. In section 5 results for quantitative and qualitative evaluation are presented. Finally, section 6 discusses the presented work.

2. Input parameters
Which features are relevant for symbolic accent label prediction had been investigated in [1], [2] and [5]. Generally the following features are used for accent label prediction both on word level and on syllable level [1][4][6]: positional features that describe at which position the current word or syllable is within an intermediate or intonational phrase (positional features may also describe when the last accent occurred); function/content word features that describe the distinction between function and content words; POS features that describe part-of-speech (POS) information; phrase break features that describe the phrase break information for the current word/syllable and surrounding words/syllables; discourse features that describe the discourse structure [2]).

For the method presented in this paper a simple feature set is used that can be easily transferred for the application within a new language. This is important for the use within our multilingual TTS system. Therefore no features for content/function word distinction and no features that describe discourse information are included in the feature set. Further, no positional features are generated.

Thus, the only features used are based on POS (the POS tags use different tags for punctuation marks, e.g. different tags are used for commas and periods) and phrase break information for the entire sentence. As proposed in [1] and [7], in our system phrase break labels are also predicted for an entire sentence using the method presented in [8] prior to predicting accent labels. Context windows are determined for the two features (cf. section 3).

3. Network architecture
Figure 1 shows the block diagram of a causal and retro-causal TDNN [9]. As can be seen the system is built up by connecting subnetworks $NN_i$, $i \in \{1, \ldots, r\}, l < 0, r > 0$. Each subnetwork is associated with one time step. The connection of these subnetworks is realized by shared weight matrices $A$ and $A'$. 

Figure 1: Block diagram of a causal and retro-causal TDNN.
(shared weights means to use the same set of weights in each
connector denoted by the same upper case bold face letter). A
propagates the causal information flow (past to future) and A'
propagates the retro-causal information flow (future to past).

Within the architecture of figure 1 shared weight matrices allow a symmetric handling of past and future information. In
our application past refers to left context and future refers to
right context.

As can be seen in figure 1, each subnetwork1 \( \mathbf{NN}_i \) has a vector \( x_t \) as an input and a vector \( y_t \) as output: \( x_t \in \mathbb{R}^n, y_t \in \mathbb{R}^m \). Since the input and output vectors, respectively, are of the same dimension for all subnetworks \( \mathbf{NN}_i \). The input
to the entire system is given by a sequence of input vectors \( x_{t_0}, \ldots, x_{t_{i-1}}, x_{t_i}, \ldots, x_{t_r} \) and the output is given by a sequence of output vectors \( y_{t_0}, \ldots, y_{t_{i-1}}, y_{t_i}, \ldots, y_{t_r} \).

In our application the input vectors \( x_{t_0}, \ldots, x_{t_{i-1}}, \ldots, x_{t_r} \) of figure 1 represent a sequence of feature vectors. Feature vec-
tors are calculated on word level, i.e. each time step \( t \) in figure 1 is associated with one word. A feature vector contains the information mentioned in section 1, i.e. POS information and phrase break information for the word at position \( t \). Output vectors, representing the target during training, are also determined on word level. An output vector contains the coded (binary) accent label for each word. By applying a sequence of output vectors as target during training (cf. figure 1), the NN parameters can be adapted according to the context of surrounding accent labels.

As mentioned in section 1 our system works on sentence level, i.e. no sentence interdependencies are modeled. There-
fore, a problem occurs at sentence boundaries: At the begin-
ing (end) of a sentence no left (right) context is available, i.e. no de-
defined input can be determined for the first (last) \( I \) (r) feature vectors \( x_{t_0}, \ldots, x_{t_{i-1}} \) and the first (last) \( I \) (r) output vectors \( y_{t_0}, \ldots, y_{t_i} \) of the input/output sequence. The broader the left and right context, i.e. the greater \( I \) and \( r \), the more feature vectors \( x_{t_j} \) and output vectors \( y_{t_j} \) are not defined.

If the used network structure is static, i.e. the structure cannot be adapted dynamically at runtime, a well defined input sig-
nal must be determined for each input neuron. In figure 1 this
means an input vector \( x_{t_i} \) must be determined for each sub-
network \( \mathbf{NN}_i \).

A quick solution to the problem might seem to be the fol-
lowing: If \( x_{t_j} \) and \( y_{t_j} \) are not defined, \( x_{t_j} \) and \( y_{t_j} \) will be set to the zero vector \( \mathbf{0} \) (all elements equal to zero), i.e. \( x_{t_j} = \mathbf{0} \) and \( y_{t_j} = \mathbf{0} \). This solution, however, does not prevent weight

Figure 2: Subnetwork \( \mathbf{NN}_i \), using a gating cluster.

adaptation during training. This is because \( x_{t_i} \) is not the only input to \( \mathbf{NN}_i \). Other inputs to \( \mathbf{NN}_i \) result from signals propagated via the shared weight matrices \( A \) and \( A' \). These signals produce an output signal for \( \mathbf{NN}_i \), which in turn produces an error signal during training. The produced error signal has as consequence weight adaptation. This cannot be prevented by setting \( y_{t_i} = \mathbf{0} \), because the output vector \( y_{t_i}^{\text{init}} \) is in general different from zero.

This problem can be overcome by dynamic adaptation of the
network structure depending on the sequence length of de-
defined input vectors \( x_{t_j} \), i.e. by shortening the sequence of \( \mathbf{NN}_i \) when necessary. This can be achieved by choosing subnetworks \( \mathbf{NN}_i \) using gating clusters as displayed in figure 2. A gating cluster multiplies each element of the two incoming signal vec-
tors with each other, whereby the two signals must be of the
same dimension. The dynamic adaptation is achieved as ex-
plained in the following:

For a defined \( x_{t_j} \), we choose the input vector \( f(x_{t_j}) = x_{p,t_j} \) (cf. figure 2) as follows:

\[
f(x_{t_j}) = x_{p,t_j} = [1, \ldots, 1]^T .
\] (1)

As a consequence \( y_{t_j} = x_{t_j} - x_{t_j} \). This means an output of
\( \mathbf{NN}_i \) is generated and during training the error signal is back-
propagated to \( \mathbf{NN}_i \). This in turn leads to weight adaptation within \( \mathbf{NN}_i \) and the rest of the network (as can be seen, the
error signal is propagated to the rest of the network via \( C \rightarrow A \) and \( C' \rightarrow A' \), respectively). The rest of the network means all
\( \mathbf{NN}_j \) with \( j \neq i \).

For an undefined \( x_{t_j} \) we let

\[
f(x_{t_j}) = x_{p,t_j} = \mathbf{0} \quad \text{and} \quad x_{t_j} = \mathbf{0} \] (2)

Thus, there are no signals from \( \mathbf{NN}_i \) to the rest of the network,
i.e. the gating cluster prevents the back-propagation of an error
signal. Therefore, weight adaptation according to an invalid error signal is prevented. Further, by choosing \( x_{t_j} = \mathbf{0} \) no signals are propagated to the rest of the network via \( B \rightarrow s_i \rightarrow A \) and \( B' \rightarrow s_i' \rightarrow A' \), respectively. Under these conditions the network behaves, as if the respective subnetwork \( \mathbf{NN}_i \) did not exist. This is used to shorten the sequence of subnetworks \( \mathbf{NN}_i \) in figure 1 if the given context is shorter than the number of subnetworks \( \mathbf{NN}_i \).
4. Corpora and labeling scheme

For the evaluation of the proposed method, two different corpora with different labeling schemes were used.

The first corpus contains 1000 sentences (21549 words) taken from the German newspaper Frankfurter Allgemeine Zeitung. The average sentence length is 21.8 words. The corpus was read aloud in studio environment by two professional male radio news speakers yielding two spoken versions of the corpus (further referenced by data base FAsp1 and data base FAsp2). The corpus is tagged with 35 different part-of-speech tags. FAsp1 was labeled with break labels and accent labels by professional labelers [10]. As break labels three different labels were used, denoting major breaks, minor breaks, and no breaks after each word. The accent labeling scheme used is the same as applied in [11]. Thus the corpus was originally labeled with the following accent labels: PA (perceived primary accent), SA (perceived secondary accent), and EC (perceived emphatic accent). In labeling experiments a low inter-annotator agreement of 70.5% was measured using the above labels. We therefore grouped all accent labels together to one label (PA) and thus only distinguish between words perceived accented and words perceived de-accented. This agrees with experiences made in [1], where it was found that data labeled with two prominence levels was not useful because of the low inter-annotator agreement. With the new labeling scheme an inter-annotator agreement of 86% was achieved. This agreement is still rather low compared to the agreement achieved in other studies (see below). This might be due to the relative monotonous speaking style of the speaker. FAsp2 was labeled with minor and major break labels and PA and SA accent labels. As before, the two accent labels were grouped.

The second corpus used is the portion of the Boston University Radio News corpus read by the female speaker F2B [12](further referenced by data base BUf2b). The corpus is labeled with minor and major phrase breaks. For the studies here, words are assigned accent labels describing whether the respective word bears a pitch accent or not. For this labeling scheme an inter-annotator agreement of 91% was reported in [1].

The inter-annotator agreement can to some extent be seen as an upper bound on performance of prediction algorithms.

5. Experiments and results

5.1. Quantitative evaluation

For training and testing both corpora have been separated into three subsets which contain approximately the following percentage of data: training set: 70%, validation set: 10% and test set (independent testing data): 20%. All results reported are determined on the test set. The validation set is used to avoid over-fitting to the training data. For the presented NN architecture the context of three feature vectors to the left and four feature vectors to the right has been used which in figure 1 corresponds to setting $l = -3$ and $r = 4$.

In our approach the issue of accent placement within the word, e.g. early accent and double accent, is not modeled, since in our German corpus such phenomena occur only for 6% of the words. For the English corpus such phenomena might need special attention [1][7]. However, in this work the focus lies on accent label prediction on word level, which has shown to be a robust basis for prosody generation in our system [9] and for prosody generation across languages in the system used in [4] (even though in [4] prominence is predicted on a scale from 0 to 9, the synthesizer used only distinguishes between accented words and de-accented words). Thus, for the experiments and results discussed below the prediction has been done on word level.

Table 1 shows the results for accent label prediction of twelve independent experiments. For the experiments in the first and second column the break labels as they are originally labeled in the three data bases (FAsp1, FAsp2, and BUf2b) were used as input information. For the experiments in the third and fourth column predicted break labels were used. The break labels were predicted using the method presented in [8]. For the respective experiments in column three and four the NN architecture presented in this paper has been trained on the data with the predicted break labels. We feel this is important since thus the accent prediction module can adapt to possible reoccurring errors from the phrase break prediction module. As in [4], the separation of training and testing data was the same for break and accent prediction, such that the testing data represents independent data for both, phrase break label prediction and accent label prediction.

“Min=Maj” in column one and three means that minor and major phrase break labels were used as an input and “Min=Maj” in column two and four means that minor break labels were grouped with major breaks. This grouping was done prior to the training of the phrase break prediction module. Thus either minor and major break labels have been predicted (Min=Maj) or only one break label (Min=Maj).

<table>
<thead>
<tr>
<th>orig. labeled breaks</th>
<th>predicted breaks</th>
</tr>
</thead>
<tbody>
<tr>
<td>orig. labeled breaks</td>
<td>predicted breaks</td>
</tr>
<tr>
<td>FAsp1 Min=Maj</td>
<td>83.1%</td>
</tr>
<tr>
<td>FAsp2 Min=Maj</td>
<td>50.9%</td>
</tr>
<tr>
<td>BUf2b Min=Maj</td>
<td>83.8%</td>
</tr>
<tr>
<td>FAsp1 Min=Maj</td>
<td>82.2%</td>
</tr>
<tr>
<td>FAsp2 Min=Maj</td>
<td>85.8%</td>
</tr>
<tr>
<td>BUf2b Min=Maj</td>
<td>84.5%</td>
</tr>
<tr>
<td>FAsp1 Min=Maj</td>
<td>82.6%</td>
</tr>
<tr>
<td>FAsp2 Min=Maj</td>
<td>86.1%</td>
</tr>
<tr>
<td>BUf2b Min=Maj</td>
<td>84.0%</td>
</tr>
<tr>
<td>FAsp1 Min=Maj</td>
<td>82.9%</td>
</tr>
</tbody>
</table>

Table 1: Results for accent label prediction with different input information for the three data bases.

As can be seen from table 1, prediction accuracy was in some cases lower, for the cases Min=Maj. The reason for this could be the higher inconsistency in labeling minor and major breaks and the higher error rate in prediction, respectively. This indicates that in our case a distinction between minor and major breaks is not necessary for accent label prediction. Results were between 1.5% and 2% lower, if no gating clusters were used.

A comparison between the results achieved when using the original break labels as an input and the results achieved when using the predicted break labels as an input shows a slight decrease in performance for the latter condition. Prediction accuracy for phrase breaks was around 90% for the case Min=Maj and 91% for the case Min=Maj (see [8] for details). The results in table 1 can be compared to results obtained when all content words are labeled accented. For the German and English corpora this leads to a prediction accuracy of 75.6% and 74.6%, respectively. Other quantitative measures are the insertion rate and the deletion rate. These are given as pairs (insertion rate/deletion rate) in the following for the case Min=Maj of predicted breaks (cf. column four in table 1). FAsp1: 10.1/7.3; FAsp2: 8.2/5.8; BUf2b: 10.6/6.5. An important question is whether the insertions and/or deletions occur at positions within a sentence where they disarrange the prosodic structure of the sentence. To answer this question a new qualitative evaluation is proposed in section 5.2.

The results achieved for the data base BUf2b are superior to the results reported for the latter data base in [1] where pre-
dition accuracy was 82.5%. For the studies in [1] the originally labeled break labels were used as an input and not predicted labels. Thus the results in column one and two of table 1 must be compared to the results of [1]. As can be seen, for the case \( \text{Min} = \text{Maj} \) the results achieved with the proposed method are better by 2.0%.

5.2. Qualitative evaluation

For quality analysis, a prosodically labeled version of the sentences in the test set of the German corpus was predicted, whereby predicted breaks have been used as an input (\( \text{Min} = \text{Maj} \) in table 1) and the NN architecture has been trained on FA sp1. Thus, the resulting testing part of the corpus (177 sentences) includes predicted phrase break labels and predicted accent labels. This testing part was given to two subjects who were asked to rate the quality of the prosodic labeling for each sentence. Both subjects had labeled speech corpora prosodically based on perception before which makes them sensitive and critic to possible errors. The subjects rated whole sentences as either good, acceptable or bad.

Table 2 shows the confusion matrix for the ratings of the two subjects (subj. 1 and subj. 2) for the sentences in the test set. As can be seen the main disagreement in rating was between the rating good/acceptable (i.e. one subject rated good and one acceptable) while almost no good/bad cases occurred. Further, it can be observed that one of the subjects was more critical. If the good/acceptable ratings are grouped, an agreement of 91% for the two subjects is met. This indicates that the proposed subjective classification scheme is a valid indication for the quality of the results.

<table>
<thead>
<tr>
<th></th>
<th>subj. 2</th>
<th>subj. 1</th>
<th>( \sum_{\text{subj.2}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>good</td>
<td>29</td>
<td>41</td>
<td>70 (40%)</td>
</tr>
<tr>
<td>accept.</td>
<td>15</td>
<td>56</td>
<td>83 (47%)</td>
</tr>
<tr>
<td>bad</td>
<td>2</td>
<td>2</td>
<td>24 (13%)</td>
</tr>
</tbody>
</table>

\( \sum_{\text{subj.1}} \) = 46 (26%) 99 (56%) 32 (18%)

Table 2: Confusion matrix for sentence-ratings of two subjects.

6. Discussion

In this paper a new approach to symbolic accent labeling—perceptual accents and pitch accents—was introduced. The approach is based on an NN architecture applying gating clusters in a time delay framework. With gating clusters the number of time steps can be adapted dynamically which enables a consistent modeling at sentence boundaries where no left or right context is available. The approach is fully integrated in our multi-lingual TTS system.

Experiments have been made with a German and an English radio news corpus. Prediction accuracy for the German and the English corpus was 85.9% and 84.5%, respectively, with hand labeled phrase break labels used as an input. The results for the English corpus are superior to results reported in [1] achieved with a CART based approach for the latter corpus. Further, experiments were made with predicted phrase break labels used as an input. For the German corpus the best result for this condition was 86.1%. It is surprising that results for the German corpus are in some cases better if predicted break labels are used as an input instead of the hand labeled break labels. The reason for this could be, that the consistency for the predicted breaks is higher and that the method presented here for accent label prediction can adapt to typical errors in phrase break prediction.

We also introduced a new qualitative evaluation method. We found, that in most cases the insertion and deletion errors from our phrase break prediction module [8] and from our method for accent label prediction presented in this paper do not lead to unacceptable prosodic parses of sentences. This makes the method highly suitable for practical application.

As input features, no positional features and no content/function word features were used. The good results achieved without the explicit use of such information might indicate that the used model can extract this information and represent it internally. The fact that no content/function word distinction must be made explicitly is a clear advantage for the application within our multi-lingual TTS system, since the classification into content and function words is often arguable and may not be available for some languages without profound linguistic studies.

Acknowledgments: We would like to thank Anton Batliner for his support during the labeling of the German corpus.

7. References