CONTEXT-ADAPTIVE PHONE BOUNDARY REFINING FOR A TTS DATABASE

Ki-Seung Lee

Dept. of Electronic Eng. Konkuk University
1 Hwayang-dong, Gwangjin-gu,
Seoul, Korea 143-701
email : kseung@kkucc.konkuk.ac.kr

JeongSu Kim

Human-Computer Interactive Labs.
Samsung Advanced Institute of Technology
Kiheung, Soowon
Kyungki-do, Korea 449-712

ABSTRACT

A method for the automatic segmentation of speech signals is described. The method is dedicated to the construction of a large database for a Text-To-Speech (TTS) synthesis system. The main issue of the work involves the refinement of an initial estimation of phone boundaries which are provided by an alignment, based on a Hidden Markov Model (HMM). Multi-layer perceptron (MLP) was used as a phone boundary detector. To increase the performance of segmentation, a technique which individually trains an MLP according to phonetic transition is proposed. The optimum partitioning of the entire phonetic transition space is constructed from the standpoint of minimizing the overall deviation from hand labelling positions.

With single speaker stimuli, the experimental results showed that more than 95% of all phone boundaries have a boundary deviation from the reference position smaller than 20 ms, and the refinement of the boundaries reduces the root mean square error by about 25%.

I. INTRODUCTION

Since the development of a corpus-based concatenative TTS [1][2], the quality of synthesized speech signals has been improved in naturalness and intelligibility. Most corpus-based TTS systems involve a large database which is built from more than a 1 hour speech corpus [2]. Generating such a huge database is a very time-consuming process in most TTS systems, hundreds of hours are required to label large database concatenative TTS systems by hand [2]. Another drawback of hand labelling is that the results lack consistency because of the subjective decisions involved [3]. Thus, the automatic labelling of a large amount of speech would be highly desirable.

Early studies on automatic labelling were largely based on what had been learned from automatic speech recognition (ASR) [4]. In these studies, an individual phoneme is modelled by HMM, automatic labelling is implemented by the alignment of a phoneme sequence on the given speech signals. The major difference between ASR and automatic phone labelling is that the phoneme sequence is already known in automatic phone labelling, hence the onset or termination time of each phone inventory is more important in phone labelling, whereas the identities of HMM are more important in ASR. The HMM based approach continue to have an important role in automatic labelling task.

In HMM based phoneme segmentation, segment boundaries are determined by maximizing a likelihood function, hence this method is based on a statistical criterion. Assuming that phone boundaries are determined not only by explicit boundaries obtained from HMM, but also implicit boundaries obtained directly from the speech signal, a more accurate estimation of phone boundaries can be achieved by combining implicit segmentation with HMM segmentation. Studies in this area have been reported in [3][5][6][7]. In [3], correlations between neighboring LPCs were used to determine implicit boundaries. A refinement of the segments based on the homogeneity of the speech segments was proposed in [5]. Multi-Layer-Perceptron (MLP) was also applied to achieve an improvement in the accuracy of the segmentation [6][7].

In this work, several specialized MLPs were used to refine different types of phonetic transitions. In [6], the use of specialized MLPs failed to yield remarkable improvements over the single MLP case. In a similar study, however, the performance of automatic segmentation was improved by phone-specific MLPs [7]. One of the reasons for the inconsistent performance of multiple MLPs lies in the partitioning of the entire phone-transition space and allocating MLPs to each partition. In [6], four different MLPs were used, each of which specialized in one of the four possible combinations of voiced and unvoiced phonemes in a transition. Since such a pre-determined partitioning does not consistently guarantee optimal partitioning in the sense of minimizing the overall deviation from reference phone boundaries, the performance may be inconsistent, even when more than one MLP were used.

To cope with this problem, a joint partitioning and training algorithm is proposed in this work, which is based on a data-driven approach. A similar approach was investigated in [8] in which multiple MLP-based predictors were designed to implement low bit rate speech coders. The optimized set of MLPs were constructed by minimizing the overall distortion between manually labelled positions and estimated ones. Moreover, the set of phonetic transitions for each MLP is automatically determined.

To evaluate the performance of the proposed phone boundary refinement algorithm, we compared the results before/after applying MLP-based postprocessing.

This paper is organized as follows. Section II provides an overview of the proposed refinement algorithm. The method used for partitioning and training multiple MLPs is introduced in Section III. Performance evaluations and concluding remarks are presented in Sections IV and V.

II. MLP-BASED BOUNDARY REFINING SYSTEM

The overall block diagram of the MLP-based phone boundary refining system is shown in Fig. 1. HMM-based explicit segmenta-

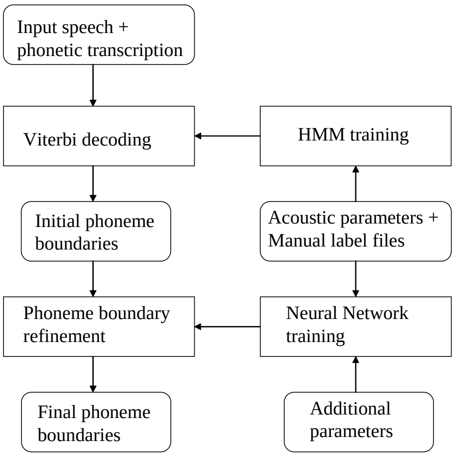

Fig. 1. Block diagram of an MLP-based phone boundary refining system

tion is initially performed with a given phonetic transcription and speech signal. This procedure is implemented by Viterbi decoding. The results from HMM segmentation are used as the initial phone boundaries for the subsequent refinement procedure. Final phone boundaries are obtained by postprocessing involving MLP.

II.A. HMM segmentation overview

The phoneme set of the underlying TTS includes 49 monophones, 23 diphones, silence and a short pause. In training HMM parameters, a typical left-to-right model is used. Feature parameters consist of 13 MFCCs(Mel Frequence Cepstrum Coefficients), 13 delta MFCCs and 13 delta-delta MFCCs. MFCCs are computed every 10 ms and, hence, the time resolution of the HMM segmentation is 0.01 s. We used HTK to build all phonemes' HMMs and to perform phoneme alignment. Several experiments were performed to determine the appropriate number of states and gaussian components which provide the best results. Based on experiments, the best results were achieved in the case of 5 states and 3 gaussian components for all phonemes.

II.B. MLP-based boundary refining

MLP was applied to refine the initial phone boundaries. The time alignment of the input features and target outputs are shown in Fig. 2. The input features of MLP include the following acoustic feature variables : 1) 4 consecutive MFCCs 2) 2 short-time ZCR(Zero Crossing Rate), 3) 1 SFTR (Spectral Feature Transition Rate)[9], 4) SKLD (Symmetrical Kullback-Leibler Distance)[10]. Hence, the total number of input nodes is 56. SFTR and SKLD are computed from the two consecutive frames. As shown in Fig. 2, the target output of MLP is set to 1.0 when a phone boundary exists between the left two consecutive frame and the right two consecutive frames. Otherwise, the target output of the MLP was set to 0. Note that if an adjacent frame is a boundary frame, the target output is 0.5. This allows MLP to have a slowly varying transition effect in the neighboring boundary frames [7].

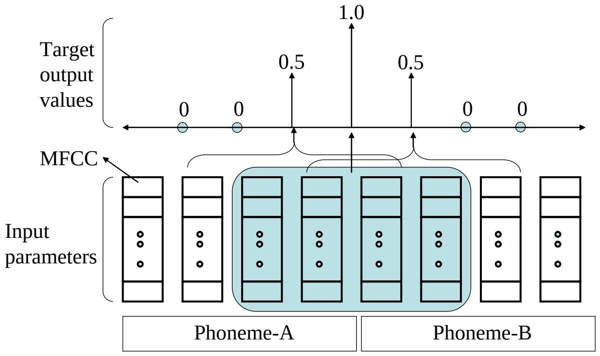

Fig. 2. Relationship between MLP input and target output

Each MLP contains 1 hidden layer having 15 nodes. We performed several experiments to investigate the relationship between the number of hidden layers and the accuracy of the refined phone boundaries. No clear relationship between them was found. We therefore concluded that one hidden layer is sufficient for our application. A standard error propagation algorithm [11] is used to train the MLPs. To decrease errors at the phone boundaries, the error for the top (output) node can be adaptively emphasized according to its error pattern. For example, if the target output is 1.0 and the actual MLP output is less than 0.5, the error for the top node is multiplied by 2. Applying adaptive weight to the output node also alleviates the training corpus bias problem which is caused by the fact that the frequency of the target output 1.0 is much smaller than that of 0.0.

In online processing, the acoustic feature variables used in training procedure are input to a trained MLP and the position of maximum MLP output is taken as the refined phone boundary. Fig. 3 represents the search area for refinement. Limiting the search area shown in Fig. 3 has the advantages of avoiding an exhaustive search and removing suspicious phone boundaries.

III. CONTEXT-DEPENDENT PHONE BOUNDARY REFINING

Assuming that the spectral trajectory of a speech signal at the phone boundary is affected by the underlying context, a more accurate estimation of phone boundaries can be achieved by applying phonetic information to the refining process. This suggests a context-dependent phone boundary refinement method where a specialized MLP is selected according to the phonetic transition. However, two problems should be considered in implementing this method; 1) How to build the optimum set of MLPs 2) How to partition the entire phoneme space. To solve these problems, a joint classification/training algorithm is proposed in this paper. The proposed algorithm is based on a minimum mean square error criterion, hence the resulting set of MLPs and partitioning provides the minimum overall deviation from the reference phone boundaries. The overall procedure for MLP set designing is shown in Fig. 4. Detailed description for each step is as follows :

Step-0. Initialization : Given training set $\{\mathbf{X}(n), y_d(n)\}_{n=1}^{N}$ where $\mathbf{X}(n)$, $y_d(n)$ and N are n-th MLP input feature vectors, target

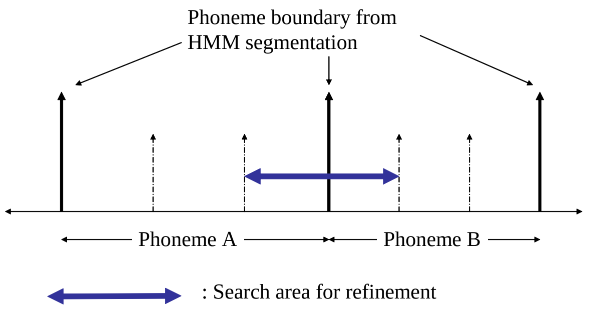

Fig. 3. Refinement area

value, and the total number of training patterns, respectively, an initial MLP set $C_0 = \{\mathbf{\Phi}_k^0\}_{k=1}^K$ is built by an adequate method, where K is the total number of MLPs. Let $\{t_m\}_{m=1}^M$ be the reference phone boundaries, then the phone transition index $P_c(m)$ for the m-th phone boundary is given by one of the possible combinations of the left-right phonemes. Set thresholds ϵ, $D_{-1} = \infty$ and $i = 0$.

Step-1. Classification : For each phoneme combination, find the optimal MLP index having minimum overall distances between the MLP outputs and the target values.

$$c_i(P_j) = \arg\min_k\{\sum_{P_c(m)=j}\sum_{n\in\Delta_m}|y_d(n) - F(\mathbf{\Phi}_k^i, \mathbf{X}(n))|^2\} \tag{1}$$

where $c_i(P_j)$ is the optimal MLP index for the j-th phoneme combination at the i-th iteration and $F(\mathbf{\Phi}_k^i, \mathbf{X}(n))$ is the k-th MLP output, when the MLP input is given by $\mathbf{X}(n)$. Δ_m denotes the phone transition interval for the m-th boundary, which is given by

$$\Delta_m = \{t|\frac{t_{m-1}+t_m}{2} \le t \le \frac{t_m+t_{m+1}}{2}\} \tag{2}$$

Step-2. Partioning : The training set $\{\mathbf{X}(n), y_d(n)\}_{n=1}^N$ is partitioned into $A^i = \{s_k^i; k = 1, ..., K\}$ according to the optimal index $c_i(P_j)$ from *step-1*. Let s_k^i be the k-th cell of partition A^i, then

$$s_k^i = \{\{\mathbf{X}(n), y_d(n)\}|n \in \Delta_m,\ where\ P_c(m) \in W_k^i\} \tag{3}$$

where

$$W_k^i = \{P_j|c_i(P_j) = k\} \tag{4}$$

Note that all $\{\mathbf{X}(n), y_d(n)\}$ in s_k^i have the optimum MLP $\mathbf{\Phi}_k^i$.

Step-3. Convergence test : Given A^i and $C_i = \{\mathbf{\Phi}_k^i\}_{k=1}^K$, compute the overall distortion at the i-th iteration

$$D_i = \sum_{k=1}^K \sum_{P_j\in s_k^i} \min_k\{\sum_{P_c(m)=j}\sum_{n\in\Delta_m}|y_d(n) - F(\mathbf{\Phi}_k^i, \mathbf{X}(n))|^2\} \tag{5}$$

If $(D_{i-1} - D_i)/D_i \le \epsilon$, stop with A^i, W_k^i and $C_i = \{\mathbf{\Phi}_k^i\}_{k=1}^K$ describing final partitioning and MLP set. Otherwise continue.

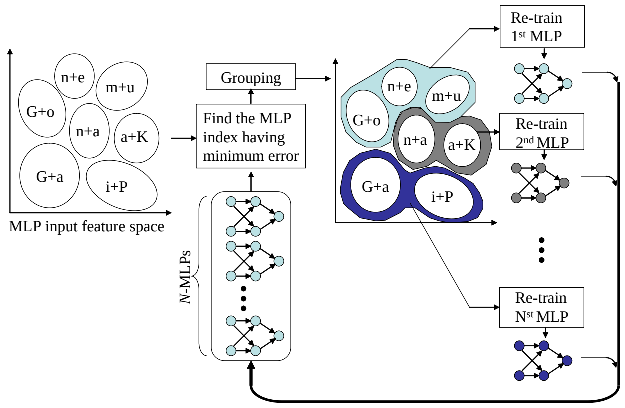

Fig. 4. MLP set designing procedure

Step-4. Re-training MLPs : Updating equation for k-th MLP parameters is given by

$$\Delta\mathbf{\Phi}_k^i = \eta \sum_{\{\mathbf{X}(n),y_d(n)\}\in s_k^i} \nabla_{\mathbf{\Phi}_k^i} \frac{1}{2}|y_d(n) - F(\mathbf{\Phi}_k, \mathbf{X}(n))|^2 \tag{6}$$

Note that the k-th MLP is trained with training patterns having an MLP index k. After re-training all MLPs, update a MLP set C_i with newly trained MLPs. Replace i by $i + 1$ and go to *Step-1*.

The algorithm iteratively finds the optimum partitioning with a given set of MLPs and updates each MLP to have the minimum mean square error within each cluster.

In online processing, the phone transition of the underlying frame is first taken, an appropriate MLP for the current phone combination is then selected using A^i, W_k^i from equation (4).

IV. EXPERIMENTAL RESULTS

The speech corpus used in our experiments consists of 1,000 utterances from a TTS talker. This corresponds to 55250 phone boundaries and 476902 feature vectors. The entire corpus was split into 400 utterances for training and 600 utterances for the test. After training, the test utterances were segmented in four ways: 1) by HMM only, 2) by HMM + the refinement algorithm with a single MLP 3), by HMM + the refinement algorithm with multiple MLPs specialized by the voicing status of phoneme [6], 4) by HMM + the refinement algorithm with multiple MLPs obtained by the joint classification/retraining algorithm. Note that both methods (3), (4) have the same number of MLPs (=4).

Since the goal of this work is to produce phone boundaries which are as close as possible to the manually segmented ones, we evaluated the performance with RMSE(Root Mean Square Error) and MAE(Mean Absolute Error) between phone boundaries obtained by hand and the estimated ones. It was empirically established that a segmentation error of 20 ms is usually acceptable. Thus we also computed the percent of all phone boundaries that have boundary deviations smaller than 20 ms. The results are summarized in Table 1. The results obtained by MLP-based refinement methods are consistently better than the results of the HMM-based

Table 1. Performance of automatic labelling

method	RMSE	MAE	% error
	(msec)	msec	<20msec
HMM olny	13.5	9.3	90.4
HMM + Single MLP	12.2	7.7	91.2
HMM + 4 MLPs (off-line)	10.7	6.8	93.0
HMM + 4 MLPs (joint)	10.1	6.2	95.2

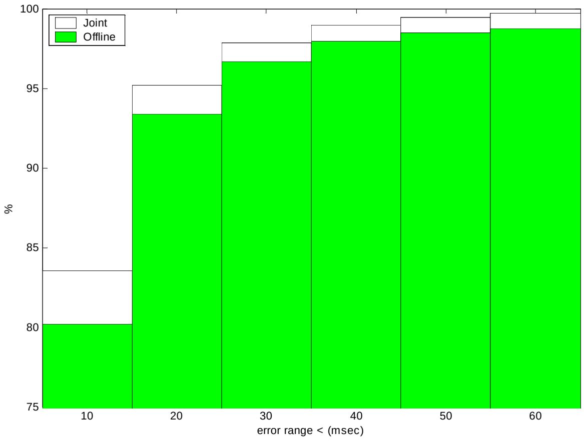

Fig. 5. Cumulative distribution of the difference in location of the phone boundaries obtained by the proposed method and the refinement algorithm with multiple MLPs specialized by the voicing status of the phoneme.

method alone. Fig. 5 shows a comparison between the phone boundaries obtained by the proposed method and the refinement algorithm with multiple MLPs specialized by the voicing status of the phoneme [6]. In the figure, the white bars and dark bars correspond to the proposed method and the method [6], respectively. It appears that the proposed joint classification/retraining method provides superior performance to the pre-determined partitioning method.

An informal listening test was conducted to compare the synthesized speech signals from the two databases; one using the proposed automatic segmentation (HMM+MLP) and the other using manual segmentation. The baseline TTS was a corpus-based waveform concatenating TTS system that employed a PSOLA (Pitch Synchronous Over Lap Addition) technique. 15 listeners participated and were asked to judge which stimulus was preferred over the other. The test data set consisted of 10 pairs of sentences. The result showed that the speech signals synthesized using the database from the proposed method were preferred for 58% of the stimuli. This results indicated that the proposed automatic segmentation can successfully replace manual segmentation.

V. CONCLUSION

A new phone boundary refining algorithm is described. MLP was employed to modify the phone boundary after HMM-based segmentation. The unique issues of this study include the optimal partitioning of phonetic transition and the construction of an optimal set of MLPs from the standpoint of a minimum mean square error criterion.

Both subjective and objective test ensured the superiority of the proposed method. It would be concluded that the proposed automatic segmentation method can yield perceptually satisfactory results and, given the other advantages, may well be preferred over manual segmentation.

VI. ACKNOWLEDGEMENT

This work was supported by Samsung Advanced Institute of Technology (SAIT) University corporation program.

VII. REFERENCES

[1] A. J. Hunt and A. W. Black, “Concatenative speech synthesis using units selected from a large speech database,” *Draft paper.*

[2] M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou and A. Syrdal, “The AT&T Next-Gen TTS system,” in *Proc. Joint Meeting of ASA, EAA, and DAGA*, Berlin, Germany, March 1999.

[3] Jan P. van Hermert, “Automatic segmentation of speech,” *IEEE Trans. Signal Processing*, vol. 39, No. 4, pp. 1008-1012, 1991.

[4] L. R. Rabiner, A. E. Rosenberg, J. G. Wilpon, and T. M. Zampini, “A bootstaping training technique for obtaining demisyllable reference patterns, *J. Acoust. Soc. Amer.*, vol. 71, pp. 1588-1595, 1982.

[5] A. Bonafonte, A. Nogueiras and A. R.-Garrido, ”Explicit segmentation of speech usnig gaussian models,” in *Proc. IEEE Int. Conf. Spoken Language Processing*, pp. 1269-1272, 1996.

[6] D. T. Toledano, ”Neural Network boundary refining for automatic speech segmentation,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 3438-3441, 2000.

[7] E.-Y. Park, S.-H. Kim and J.-H. Chung, ”Automatic speech synthesis unit generation with MLP based postprocessor against auto-segmented phoneme errors,” in *Proc. International Joint Conference on Neural Networks*, pp. 2985-2990, 1999.

[8] L. Wu, M. Niranjan, and F. Fallside, “Nonlinear Predictive Vector Quantization with Recurrent Neural Nets,” *Proc. IEEE-SP Workshop on Neural Networks for signal Processing*, pp. 372-381; Baltimore, MA, 1993.

[9] A. C. R. Nandasena and M. Akagi, ”Spectral stability based event localizing temporal decomposition,” *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 3438-3441, 1998.

[10] E. Klabbers and R. Veldhuis, “Reducing audible spectral discontinuities,” in *IEEE Trans. Speech and Audio Singal Processing*, vol. 9, No. 1, pp. 39-51, 2001.

[11] Richard P. Lippmann, ”An introduction to computing with neural nets,” *IEEE ASSP magazine*, pp. 4-22, April, 1987.