Speech Segmentation using Regression Fusion of Boundary Predictions

Iosif Mporas[1], Todor Ganchev, Nikos Fakotakis
Artificial Intelligence Group, Wire Communications Laboratory,
Department of Electrical and Computer Engineering, University of Patras, Greece
Tel. +30 2610 996496, Fax. +30 2610 997336, GR-26500
{imporas, tganchev, fakotaki}@upatras.gr

Abstract

In the present work we study the appropriateness of a number of linear and non-linear regression methods, employed on the task of speech segmentation, for combining multiple phonetic boundary predictions which are obtained through various segmentation engines. The proposed fusion schemes are independent of the implementation of the individual segmentation engines as well as from their number. In order to illustrate the practical significance of the proposed approach, we employ 112 speech segmentation engines based on hidden Markov models (HMMs) which differ in the setup of the HMMs and in the speech parameterization techniques they employ. Specifically we relied on sixteen different HMMs setups and on seven speech parameterization techniques, four of which are recent and their performance on the speech segmentation task have not been evaluated yet. In the evaluation experiments we contrast the performance of the proposed fusion schemes for phonetic boundary predictions against some recently reported methods. Throughout this comparison, on the established for the phonetic segmentation task TIMIT database, we demonstrate that the support vector regression scheme is capable of achieving more accurate predictions, when compared to other fusion schemes reported so far.

Keywords: speech segmentation, regression fusion, hidden Markov models

1. Introduction

The contemporary speech technology heavily depends on large speech corpora, whose annotation is a tedious task and is usually performed manually or semi-automatically. In general, speech databases consist of recordings and some sort of indexing, which can include word transcription, phonetic transcription, phone level time-alignment and prosodic annotation (Sagisaka et al., 1997; Campbell and Black, 1997; Iwano et al., 2004). While in automatic speech recognition (ASR) word and phonetic transcriptions are sufficient for the training of acoustic models, in text to speech (TTS) synthesis phone-level time-alignment is also needed (Dutoit, 1997). Furthermore, when bootstrap data with time-alignment are available the HMM parameters are better initialized and fine-tuned (Malfrere et al., 2003). In general, word transcriptions are extracted easily from the speech waveform by utilizing automatic transcribers and manual corrections over the automatically extracted word sequence. Similarly easy, phonetic transcription is usually extracted from the word level annotation using grapheme to phoneme converters. In contrast to the above indexes, the extraction of phonetic time-alignment is considered as a difficult task.

Presently, the most accurate way to extract the time boundaries of the phones of a speech waveform is manually. However, manual segmentation is a tedious, time-consuming and costly task that can be performed only by expert phoneticians (Acero, 1995). Moreover, the use of human annotators introduces subjectivity in the position of the phone transitions (van Hemert, 1991; Pellom and Hansen, 1998). Due to the difficulties that manual segmentation presents, methods have been developed for the automatic segmentation of speech waveforms to the corresponding phonetic units. Automatic segmentation techniques can roughly be divided into two major categories, implicit and explicit segmentation (van Hemert, 1991). In the explicit case, the segmentation algorithm is linguistically constrained to an a priori known phonetic sequence, while in the implicit case there is no prior knowledge of the corresponding phonetic sequence. Explicit segmentation methods are utilized when indexing database recordings, where the phonetic sequence is usually known.

Various approaches have been proposed for the task of speech segmentation, such as: the detection of variations/similarities in spectral (Svendsen and Soong, 1987; Dalsgaard et al., 1991; van Hemert, 1991; Grayden and Scordilis, 1994; Petek et al., 1996; Aversano et al., 2001) or prosodic (Adami and Hermansky, 2003) parameters of speech, the template matching using dynamic programming and/or

[1] Corresponding author. Tel. +30 2610 996496, Fax. +30 2610 997336, imporas@upatras.gr, I. Mporas

the synthetic speech (Bajwa et al., 1996; Paulo and Oliveira, 2003; Malfrere et al., 2003) and the discriminative learning segmentation (Keshet et al., 2007).

The most frequently used speech segmentation approach is based on HMM phone models (Ljolje and Riley, 1991; Brugnara et al., 1993; Ljolje et al., 1997; Pellom and Hansen, 1998; Mporas et al., 2008). This method became popular as it is less prone to gross errors (Kominek et al., 2003) and because of its well-known structure from the area of speech recognition. In Figure 1, we show the block diagram of the HMM-based segmentation approach for the linguistically constrained case. In this method each speech waveform is initially decomposed to a sequence of feature vectors, using a speech parameterization technique. Afterwards, an HMM phone recognizer is utilized to force-align the feature vector sequence against the corresponding phonetic sequence through the Viterbi algorithm (Viterbi, 1967). The outcome of this process is the time positions of the phonetic transitions.

Figure 1

There are two main training strategies of the HMM phone models, depending on the availability of manually segmented speech data (bootstrap data). When bootstrap data are available, isolated-unit training is performed, where the speech frames that correspond to each phone are separately used to initialize and refine, through Viterbi algorithm the HMM parameters of the corresponding phone model. When bootstrap data are not available, embedded training is performed, where the HMM parameters of all models are computed simultaneously utilizing all the speech frames of the training data. In embedded training the models are initialized by setting global values to the HMM parameters of all phone models (flat initialization) and refined by Baum-Welch (Baum et al., 1970) algorithm. Phone models can be trained on other speech corpora and further be used with/without adaptation on the target data.

HMM-based segmentation has successfully been combined with post-processing techniques to refine the predicted phone boundaries (Sethy and Narayanan, 2002; Kim and Conkie, 2002; Toledano et al., 2003; Matousek et al. 2003; Wang et al., 2004; Adell et al., 2005; Lee, 2006; Lin and Jang, 2007; Lo and Wang, 2007). Furthermore, methods for fusion of the segmentation outputs from different approaches and/or systems have been proposed. In (Jarifi et al., 2008) it has been shown that linear combination of the predictions of global and local approaches for automatic segmentation improves the segmentation accuracy. In (Park and Kim, 2006; Park and Kim, 2007) the overall segmentation accuracy is improved using a linear combination of the predictions of several independent HMM-based segmentation methods and a gradient projection method for the computation of the weights. (Kominek and Black, 2004) showed that big segmentation mistakes have a greater impact on the perceived quality of an utterance than several smaller ones, and therefore averaging among a number of estimates for each boundary is a simple and effective way to avoid gross inaccuracies.

Up to the authors' best knowledge, all previous studies on fusion of a number of segmentation engines (Kominek and Black, 2004; Park and Kim, 2006; Park and Kim, 2007; Jarifi et al., 2008) can be generalized to some form of linear combination of the boundary positions that were predicted from several independent segmentation engines. Moreover, these studies considered segmentation of speech waveforms only for the case of single-speaker recordings.

Here, we propose the use of regression analysis for the fusion of the predictions of independent segmentation engines. Specifically, we evaluate both linear and non-linear regression algorithms that have been successfully used on different numerical prediction tasks, such as forecasting and phone duration prediction.

In contrast to the previous studies on fusion of segmentation engines, in the present work we consider the general case of speaker-independent phonetic segmentation and thus perform validation experiments on the well-known TIMIT multi-speaker database (Garofolo, 1988), which has been established for the validation of phonetic segmentation approaches (Ljolje and Riley, 1991; Brugnara et al., 1993; Grayden and Scordilis, 1994; Wightman and Talkin, 1997; Pellom and Hansen, 1998; Aversano et al., 2001; Keshet et al., 2007; Lo and Wang, 2007; Mporas et al. 2008). In order to increase the variability among the segmentation engines' predictions we utilized seven different speech parameterization techniques that have successfully been used on the speech recognition task. It should be noted that five out of the seven speech parameterizations considered here have not been studied on the speech segmentation task before, and as reported in Section 4 some of them offer an advantageous performance when compared to the widely-used Mel frequency cepstral coefficients (MFCCs).

The proposed fusion scheme is independent from the implementation of the individual segmentation engines as well as from their number. We assume that the output of any given regression algorithm, i.e. the predicted phonetic boundary positions, will be more precise than (or at least as good as) the ones predicted by each of the individual segmentation engines. This is because the regression algorithms are capable of capturing and modelling the systematic errors of each segmentation engine, as well as the systematic boundary shifts among the segmentation engines across each boundary type. By the term *boundary type* we refer to the transition between the left context phonetic class of a boundary to the right context class, e.g. vowels, affricates, fricatives, nasals, glides, stops and silence. In the experimental comparison presented in Section 4, we demonstrate that the support vector regression scheme is capable of achieving more accurate predictions, when compared to various implementations of linear fusion schemes reported in the literature.

Since in the present work we do not examine the recognition of the phonetic sequence but the accurate detection of the phonetic transition positions in what follows explicit segmentation is assumed.

The remaining of this article is organized as follows: In Section 2 we describe the general regression fusion structure for combining multiple phonetic boundary predictions, as well as the regression algorithms evaluated here. In Section 3 we explain the experimental setup and outline the baseline segmentation engines utilized in the experiments. Next, in Section 4 we report results related to the performance of various recent and traditional speech features, as well as to the ranking of a variety of fusion schemes for phonetic boundary predictions. Finally, in Section 5 we conclude this work.

2. Regression Fusion of Multiple Phonetic Boundary Predictions

The block diagram of the proposed regression fusion scheme for combining multiple different segmentation engines is presented in Figure 2. This general fusion scheme covers both the linear and non-linear fusion cases and is independent from the implementation of the individual segmentation engines as well as from their number.

Figure 2

Let us define a set of N phone transition position predictions S_i, with $1 \le i \le N$, as the outcome of N different segmentation engines. These engines, which in the rest of this paper will be referred to as baseline segmentation engines (BSEs), produce phonetic boundary predictions that are independent to each other. The predictions are combined with the use of a regression fusion function f to create a new phone transition position prediction $S_{pred} = f\left(S_1, S_2, ..S_N, p(b)\right)$, where $p(b)$ defines the parameters of the fusion function, and b defines the phonetic boundary type. The parameters of the fusion function, $p(b)$, are adjusted by minimizing an error function $\varepsilon_f\left(S_{real}, S_{pred}\right)$, which is specific for each fusion function and expresses the misalignment between the real and predicted phone transition positions on a training bootstrap set *D*(*b*), i.e. $\underset{D(b),\,p(b)}{\arg\min}\left(\varepsilon_f\right)$. For the real phone transition positions, S_{real}, we consider the manually annotated labels of the phonetic boundaries available in the speech database.

Since different BSEs offer different performance at specific boundary types (Jarifi et al., 2008), we hypothesize that an appropriate combination of them could increase the overall performance. Furthermore, intuitively we assume that specific boundary fusion techniques would be more successful on the given task than other techniques. For that purpose we are interested in investigating the performance of various fusion functions and evaluating their applicability for the present problem. Specifically, we consider fusion approaches that have already been studied in the literature such as the simple average (Kominek and Black, 2004), the best-only selection (Park and Kim, 2006), the linear combination of (Jarifi et al., 2008) but more importantly the linear regression, multilayer perceptron neural networks, support vector regression and model trees, whose performance haven't been investigated on the specific task, yet.

For the purpose of comprehensiveness in the following subsections we review the regression techniques of interest.

2.1. Linear Regression: LR(AIC)

In linear regression (LR) all boundary predictions are weighted and summed, i.e. the fusion function f takes the form

$$S_{pred} = w_0(b) + \sum_{i=1}^{N} w_i(b) S_i \ . \quad (1)$$

The attribute weights $w_i(b)$, for each boundary type *b*, are computed by applying the least-squares criterion over the training data,

$$\arg\min_{w_i(b)} \left\{ \sum_{i=1}^{D(b)} \left(S_{real}(i) - w_0(b) - \sum_{j=1}^{N} w_j(b) S_j(i) \right)^2 \right\}, \quad (2)$$

where $S_j(i)$ is the position prediction of the *j*-th BSE for the *i*-th boundary, $D(b)$ is the size of the training data for the corresponding boundary type and $w_0(b)$ stands for the bias.

In the case of *average fusion* the weights are $w_i(b) = 1/N$, for $1 \le i \le N$. As for the *best prediction selection* case for boundary type *b*, the weights are $w_i(b) = 1$, for $i = best$ and $w_i(b) = 0$, for $i \ne best$.

Instead of using all attributes, M5' decision trees (refer to Section 2.4) can be applied for feature selection (Wang and Witten, 1997). During feature selection the attribute with the smallest standardized coefficient is iteratively removed until no improvement is observed in the error estimation. The error estimation is given by the Akaike information criterion (Akaike, 1974) as:

$$AIC = 2k + D(b)\left(\ln\left(\frac{2\pi R_S}{D(b)} \right) + 1 \right), \quad (3)$$

where k is the number of parameters in the statistic model and R_S is the residual sum of squares:

$$R_S = \sum_{i=1}^{D(b)} \left(S_{real} - S_{pred} \right)^2 . \quad (4)$$

Here R_S indicates the cumulative squared error with respect to the real boundaries, and a smaller value of the AIC indicates for a better model.

2.2. Multilayer Perceptron Neural Networks: MLP NN

Neural networks (NNs) with three layers have been proved capable for numerical predictions (Chester, 1990), since neurons are isolated and region approximations can be adjusted independently to each other. In detail, the output z_j of the j th neuron in the hidden layer of a multilayer perceptron (MLP) NN is defined as:

$$z_j = f\left(\sum_{i=1}^{N} w_{ji}^{(1)}(b) S_i + w_{j0}^{(1)}(b) \right), \quad j = 1, 2, ..., M, \quad (5)$$

where $f(x) = \left(1 + e^{-x}\right)^{-1}$ is the sigmoid activation function, M is the total number of neurons in the hidden layer, and $w_{ji}^{(1)}(b)$ and $w_{j0}^{(1)}(b)$ are the weight and bias terms, respectively. In the present work the output layer of the MLP NN consists of a single unthresholded linear unit, and the network output, S_{pred}, is defined as:

$$S_{pred} = \sum_{j=1}^{M} w_j^{(2)}(b)\ z_j + w_0^{(2)}(b) \ . \quad (6)$$

All weights are adjusted during the training through the back propagation algorithm.

2.3. Support Vector Regression: SVR

For the non-linear case of support vector regression (SVR) the two most widely used algorithms are the ε-SVR (Vapnik, 1998) and the ν-SVR (Scholkopf et al., 2000). Here we utilize the ν-SVR because of its ability to automatically adjust the ε insensitive cost parameter. Given the set of training data $\{\mathbf{x}_i, S_{real}(i)\}$ for the boundary type *b*, with $\mathbf{x}_i = \left[S_1(i), ..., S_N(i) \right]^T$ and $1 \le i \le D(b)$, a function ϕ maps the attributes to a higher dimensional space. The primal problem of ν-SVR,

$$\arg\min_{\mathbf{w}, \varepsilon, \xi_i, \xi_i^*} \left\{ \frac{1}{2} \mathbf{w}^T \mathbf{w} + C\left(\nu\varepsilon + \frac{1}{k} \sum_{i=1}^{k} (\xi_i + \xi_i^*) \right) \right\}, \quad (7)$$

is subject to the following restrictions: $\left(\mathbf{w}^T\phi(x_i)+\beta\right)-S_{real}(i)\le\varepsilon+\xi_i$, $S_{real}(i)-\left(\mathbf{w}^T\phi(x_i)+\beta\right)\le\varepsilon+\xi_i^*$, $\xi_i,\ \xi_i^*\ge 0$, with $\mathbf{w}\in\mathbb{R}^N$, $\beta\in\mathbb{R}$, $i\in[0,\ \mathrm{N}]$ and $\varepsilon\ge 0$. Here, ξ_i and ξ_i^* are the slack variables for exceeding the target value more or less than ε, respectively, and C is the penalty parameter. The kernel function is $K(\cdot,\cdot)=\phi(x)^T\phi(x)$. The value of v affects the number of support vectors and training errors.

Here we consider the radial basis kernel function $K(x_i,x_j)=\exp\left(-\gamma\left\|x_i-x_j\right\|^2\right)$.

2.4. Model Trees: M5'

Here we consider the M5' model tree algorithm proposed by (Wang and Witten, 1997), which is a rational reconstruction of M5 method developed by (Quinlan, 1992). In tree structures, leaves represent classifications and branches represent conjunctions of attributes. The M5' tree is a binary decision tree constructed in two steps, namely the splitting and the pruning phase. During splitting, for each node the algorithm computes the best attribute to split the T subset of data that reaches the node. The error criterion is the standard deviation of each class value that reaches each node. The attribute i with the maximum standard deviation reduction $\hat{\sigma}$ is selected for splitting that node, i.e.

$$\arg\max_i\left\{\hat{\sigma}=\sigma(T)-\sum_j\frac{|T_{ij}|}{|T|}\times\sigma(T_{ij})\right\} \qquad (8)$$

where T_{ij} are the subsets that result from splitting the node according to the chosen attribute i, with $1\le i\le N$. The splitting process, which results to child nodes with smaller standard deviation, terminates when class values of the instances that reach a node have standard deviation equal to a small fraction of the original instance set, or if only few instances remain. When splitting is completed a large tree structure will be constructed. For each node one linear regression model is calculated and simplified by dropping the attributes that do not reduce the expected error. The error for each node is the averaged difference between the predicted and the actual value of each instance of the training set that reaches the node. The computed error is weighted by the factor $(n+v)/(n-v)$, where n is the number of instances that reach that node and v is the number of parameters in the linear model that give the class value at that node. This process is repeated until all the examples are covered by one or more rules. During the pruning phase, sub-trees are pruned if the estimated error for the linear model at the root of a sub-tree is smaller or equal to the expected error for the sub-tree.

3. Experimental Setup

The regression fusion scheme, shown in Figure 2, employs multiple phonetic boundary position predictions that are obtained from independent segmentation engines (refer to Section 2). In the present work these BSEs were implemented as HMM-based segmentation engines, utilizing the HTK toolkit (Young et al., 2006), and differ in the speech parameters fed on their input and/or in the settings of the HMM engine itself. Several factors such as the number of HMM states, the number of Gaussian mixtures per state, the frame shift and length, and the context dependency of the phone models can affect the segmentation performance. Although each combination of these factors would result to a different BSE, here we restrict the evaluation to 112 different BSEs. Their settings were chosen based on practical considerations and findings of previous research on speech segmentation (Brugnara et al., 1993; Pellom and Hansen, 1998; Park and Kim, 2006; Park and Kim, 2007).

Specifically, an experimental setup similar to (Brugnara et al., 1993) was followed here. In particular, each BSE utilized 3-state and 4-state left-to-right HMMs, without skipping transitions to train one model for each phone. Both context-independent (CI) and context-dependent (CD) HMM models were trained. Every HMM state was modelled by 1, 2, 4 and 6 linear combinations of continuous Gaussian densities with diagonal covariance matrix. For the case of CD phone models, similar HMM states were tied, with outlier threshold (parameter RO in HTK) equal to 100 and cluster log-likelihood threshold (parameter TB in HTK) equal to 350. In both CI and CD cases, speaker-independent models were trained.

3.1 Speech Pre-processing and Speech Parameterization

It has been shown in the literature (Pauws et al., 1996; Paulo and Oliveira, 2003) that some speech features present significantly better ability to detect certain types of phonetic transitions compared to others. Since different speech parameterization techniques lead to somehow different boundary position predictions, which for specific transitions are more accurate than others, we hypothesized that if such predictions are combined in a reasonable manner the outcome of their fusion might turn out to be beneficial in terms of accuracy. Based on this assumption and on the idea of performing fusion per boundary type, in the present work we implemented seven speech parameterization techniques, which feed multiple parallel BSEs, whose outputs are combined (Figure 2).

In brief, the seven speech parameterizations implemented here utilize the standardized speech processing procedure (ETSI, 2000; ETSI, 2007). In that way, a number of setup dependent parameters (e.g. sampling frequency, frequency bandwidth of speech signal, etc) that were disparate in the original studies, where these speech parameterizations were proposed initially, were unified. Specifically, assuming speech signal sampled at 16 kHz, we adapted all speech parameterization techniques to frequency bandwidth [100, 7000] Hz. Moreover, according to the ETSI procedures, uniform pre-processing, consisting of pre-emphasis with factor a=0.97, frame blocking and windowing of the speech signal were carried out. Speech waveforms were frame blocked every 5 milliseconds as in (Brugnara et al., 1993; Pellom and Hansen, 1998; Jarifi et al., 2008; Park and Kim, 2007), using a 16 millisecond window. Here we do not make use of the 20 millisecond window length as in (Brugnara et al., 1993) due to the restriction of the discrete wavelet packet transform (DWPT), on which all wavelet-based features rely on, to be applied on a number of samples which is a power of two.

After the pre-processing of the speech signal, the feature extraction was performed following the particular speech parameterization procedure, as it was introduced by the original authors, except that we adapted the frequency range of all filter-banks to the desired bandwidth. In the following paragraphs we summarize these changes:

***Mel-Frequency Cepstral Coefficients* (*MFCC*):** The MFCC implementation of (Slaney, 1998) utilized a filter-bank of forty equal-area filters, which covers the frequency range [133, 6855] Hz. The first 13 filters in the filter-bank are with linearly spaced centre frequencies in the range [200, 1000] Hz, and the next 27 have their centres logarithmically spaced in the range [1071, 6400] Hz, with logarithmic factor 1.0711703.

***Linear Frequency Cepstral Coefficients* (*LFCC*):** The LFCC parameterization as in (Davis and Mermelstein, 1980) was adapted by implementing a filter-bank of forty equal-width equal-height filters, each one with pass-band of 164 Hz. This resulted in filter-bank that covers the frequency range [133, 6857] Hz.

***Human Factor Cepstral Coefficients* (*HFCC-E*):** The HFCC filter-bank of (Skowronski and Harris, 2004) that has twenty-nine filters covering bandwidth [0, 6250] Hz, was adapted by discarding the two filters with lowest centre frequencies and adding a new one at the high-frequency end of the filter-bank. This resulted in a filter-bank that covers the frequency range [125, 6844] Hz with twenty-eight filters. The filter-bank was designed for E-factor equal to one.

***Perceptual Linear Prediction* (*PLP*):** The eighteen-filter Bark-spaced filter-bank utilized in the PLP (Hermansky, 1990) covering the frequency range [0, 5000] Hz was adapted by discarding the lowest-frequency filter and adding two new high-frequency filters with Bark-spacing. This led to a filter-bank of nineteen filters that cover the frequency range [100, 6400] Hz, which is the closest feasible implementation.

***Wavelet-Packet Features* (*WPF*):** In the WPF (Farooq and Datta, 2001) the twenty-four frequency subbands approximating the Mel-scale in the frequency range [0, 8000] Hz were reduced to twenty-two by eliminating the lowest and highest frequency subbands. This way the frequency range [125, 7000] Hz is covered. The WPF utilize wavelet packet decomposition (WPD) based on the Daubechies wavelet of order 12.

***Subband-Based Cepstral parameters* (*SBC*):** In the SBC (Sarikaya and Hansen, 2000) the authors used twenty-four Mel-spaced subbands to cover the frequency range [0, 4000] Hz. We adjusted this frequency division to the desired frequency range by discarding the two lowest subbands and adding at the high-frequency end six new subbands of 500 Hz each. This resulted in Mel-scale frequency warping with twenty-eight subbands that cover the frequency range [125, 7000] Hz. The SBC utilize WPD based on the Daubechies wavelet of order 32.

***Mixed Wavelet Packet Advanced Combinational Encoder* (*MWP-ACE*):** The MWP-ACE speech features (Nogueira et al., 2006) utilize twenty frequency subbands to cover the frequency range [0, 8000] Hz. In our implementation, we discarded the lowest and highest subbands, which resulted in a total of eighteen subbands that cover the frequency range [125, 7000] Hz. The MWP-ACE features utilize WPD based on the Symlets family with the Symlets wavelet of order 6 on the first level, Symlets 5 on the second, etc.

A comprehensive description of the different speech parameterizations utilized here can be found in the corresponding references. In all speech parameterization schemes we computed only the first thirteen cepstral coefficients. Before training the HMM models, feature vectors composed of the static speech features and their delta coefficients were composed, resulting to a 26 dimensional parametric vector.

3.2 Fusion Scheme

We utilized the Weka (Witten and Frank, 2005) and LibSVM (Chang and Lin, 2002) implementations of the regression algorithms described in Section 2. The MLP neural network consisted of three layers. The number of input nodes was equal to the number of BSEs while the number of hidden nodes was empirically set equal to 65. The output layer of the MLP NN contains a single neuron. In the case of SVR, the v parameter was empirically set to 0.5, while the C and γ parameters, which were set equal to 2^{1} and 2^{-9} respectively, were determined by grid search ($C=\{2^{-5}, 2^{-2}, \ldots, 2^{4}\}$, $\gamma=\{2^{-15}, 2^{-12}, \ldots, 2^{3}\}$) on a randomly selected bootstrap subset, consisting of approximately 1/4 of the training data.

3.3 Evaluation Database

The performance of each regression algorithm was evaluated on TIMIT database (Garofolo, 1988). TIMIT is the most widely used corpus for phone segmentation and has been established for this task (Brugnara et al., 1993; Wightman and Talkin, 1997; Keshet et al., 2007). Briefly, it consists of microphone quality recordings of 630 American-English speakers (10 sentences per speaker), with sampling frequency 16 kHz and resolution 16-bit.

Here, we rely on the standard train/test subset division of the database, i.e. the train subset was utilized for the training of both the HMM phone models and the fusion models, while the segmentation accuracy was measured on the test subset. The SA sentences, which are common for all speakers, were excluded from the evaluation. This resulted to eight sentences per speaker, i.e. 3696 and 1344 sentences in the train and test subsets, respectively. We utilized the well established for American-English set of 48 phones, proposed by (Lee and Hon, 1989). Successive occurrences of the same phone were merged to one single occurrence as in (Brugnara et al., 1993; Pellom and Hansen, 1998).

The phonetic clustering defined in the TIMIT documentation was used: affricates (AFF), fricatives (FRI), nasals (NAS), semivowels and glides (GLI), stops (STO), vowels (VOW) and silence (SIL).

In the present work the segmentation accuracy was measured in terms of the percentage of predicted boundaries within a tolerance of t milliseconds from the manually annotated boundary labels, which is the most commonly used figure of merit. Furthermore, we also present the performances in terms of mean absolute errors (MAEs) and root mean squared errors (RMSEs).

4. Experimental Results

We firstly investigated the performance of the BSEs described in Section 3, on the phonetic segmentation task. The predictions of these engines per phonetic transition type are further utilized to perform the regression fusion scheme shown in Figure 2 for several regression algorithms.

4.1 Results for the Baseline Segmentation Engines

Specifically, first of all, we computed the segmentation accuracy for each BSE separately. The performance results, i.e. the amount of correctly detected phonetic boundaries in percentages, for different number of HMM states, s, and Gaussian mixtures, m, for CI and CD phone models are shown in Tables 1 and 2, respectively. In the tables, the setup of each BSE is denoted in brackets as [*m*-*s*-CI/CD]. The best performance for each tolerance interval is tabulated in bold. The last two columns in Tables 1 and 2 show the performance of each BSE in terms of MAE and RMSE.

Table 1

Table 2

As can be seen in Tables 1 and 2, the best performance for all examined tolerances was achieved by the HFCC-E speech parameters. In detail, in the [1-4-CI] setup the HFCC-E showed the best performance for the tolerances 10 and 30 milliseconds. In contrast to the most widely used MFCC features, where the best performance on TIMIT for the tolerance area of 15-25 milliseconds is achieved for the setup [1-4-CI], the HFCC-E features demonstrated the best performance for the setup [2-3-CI]. The differentiation in the segmentation ability between the speech features is owed to the dissimilar implementation of the filter-banks among the different speech parameterization methods. For instance, while the MFCC filter-bank is based on the Mel-scale, the HFCC-E filter-bank is derived from the equivalent rectangular bandwidth (ERB) introduced by (Moore and Glasberg, 1983). Furthermore, in the HFCC-E filter-bank the filter bandwidth is decoupled from the filter spacing, which results to a smaller overlap among the filters with low centre frequencies and a bigger overlap among the filters with high centre frequency, when compared to the filter-bank of the MFCC.

As presented in the tables, in most of the cases the CI models outperformed CD models. This is in agreement with (Toledano et al., 2003), where it was shown that CI phone models present, in average, higher segmentation scores than the CD ones, since the latter tend to lose the alignment with the boundaries during training.

The experimental results presented in Tables 1 and 2 show that the modelling of the HMM states with more than two Gaussian components generally reduces the phonetic segmentation accuracy of the BSEs. This is due to the inherent variance of the spectrum in the vicinity of a phonetic transition, which could make a simpler model more adequate (Toledano et al., 2003). The superiority of HMMs modelled with fewer Gaussians is more intense for small tolerances, while for intermediate and large tolerances this tendency is weakened or even inverted, as reported in (Toledano et al., 2003) for tolerance equal to 50 milliseconds. Another explanation could be the amount of data in the training subset of TIMIT, which might not be sufficient to successfully train with many mixtures the HMM states of the phones with few occurrences in the database.

The experimental results point out that the best segmentation performance was achieved for the HFCC-E BSE, for all the examined tolerances and parameter setups, followed by the PLP and MFCC segmentation engines. The advantageous performance of the HFCC-E speech parameters is due to the better frequency resolution of their filter-bank at low frequencies. This is in accordance with recent insights, which suggest that the human auditory system is capable of a better frequency resolution at low frequencies, in comparison to the one incorporated in the Mel-scale, and that resolution continues to improve with the decrease of the frequency (Moore, 2003). Indeed, the Mel-scale had been approximated with uniform frequency resolution at low frequencies, since at the time it was proposed there were only few measurements about the frequency resolution of the human auditory system at low frequencies, i.e. [0-500] Hz. Despite the fact that the increasing frequency resolution at low frequencies is accounted in some MFCC implementations (Young et al., 2006) and in the implementation of the PLP cepstral coefficients, the frequency resolution computed by the ERB is finer and as shown in (Moore 2003) is a closer match to the one of the human auditory system.

As Tables 1 and 2 show, the best performing BSEs in terms of MAE and RMSE are the HFCC-E [1-4-CI] and HFCC-E [6-3-CI] respectively. While the MAE and RMSE statistics generally vary in unison, here the presence of outliers in the error distribution generates large values of RMSEs for some of the BSEs. The segmentation accuracy shown in Tables 1 and 2 indicates only the averaged performance across all the phonetic boundary types in the test subset of TIMIT. A further analysis showed that neither the HFCC-E based speech segmentation engine nor the [2-3-CI] setup are the best for *every* phonetic class transition type, but only in average among all phone boundary types. Table 3 shows the best BSE for each boundary type for the most commonly used tolerance of 20 milliseconds.

Table 3

As can be seen in Table 3, despite the overall performance results shown above, there are phonetic boundary types, where other parameterization techniques and setups with higher number of mixtures per HMM state and/or context dependent models present superior accuracy. This is due to the fact that close to the area of a phone boundary, class-specific characteristics such as continuant/non-continuant, periodic/non-periodic, short/long duration (Deller et al., 1993) are transited from one target articulation area to another. Thus, different speech parameterization techniques and BSE setups, with different time-frequency resolution, offer different ability to capture the position of specific boundary types, even when the same segmentation method is considered (here HMM-based).

The evaluation of the BSEs on TIMIT database indicated that the best performing speech features are the HFCC-E, which significantly outperformed the widely-used MFCC both in terms of segmentation accuracy and in terms of mean absolute error. The superiority of HFCC-E was observed across all examined error tolerances. However, since none of the speech features offers advantage for all boundary types, a collaborative scheme that exploits the complementary information provided by the segmentation engines, employing dissimilar speech features, would contribute to a further improvement of the phonetic segmentation accuracy.

4.2 Results for the Fusion Schemes

The experimental results shown in Table 3 are a clear indication that in order to achieve optimal accuracy on the phonetic segmentation task either boundary-specific speech features and BSE setups or appropriate fusion schemes, which learn the proper combination function from a representative training dataset, have to be employed.

Table 4 shows the results obtained after combining the 112 BSEs shown in Tables 1 and 2, with the use of the regression fusion algorithms described in Section 2. These algorithms are the support vector regression (SVR), the linear regression with the Akaike information criterion, LR(AIC), the three-layer multilayer perceptron (MLP) neural network and the model trees (M5'). In addition, we present results for three formerly proposed fusion methods: the per boundary type best-only engine (BEST) of (Park and Kim, 2006), the average of all predictions (AVE) presented in (Kominek and Black, 2004), and the general fusion technique (GFT) proposed in (Jarifi et al., 2008) for the best performing case, i.e. the soft supervision case with weighting functions $f(x)=x$ and $f(x)=1/(1-x)$. For the purpose of direct comparison with the evaluated fusion algorithms, the best segmentation accuracy among the individual BSEs shown in Tables 1 and 2 for each tolerance, MAE and RMSE are duplicated in the last row of the table denoted as "No Fusion". The last two columns present the MAE and RMSE values. For the GFT and BEST fusion methods the MAE and RMSE values correspond to tolerance 20 milliseconds, as these methods use different fusion function and compute different boundary predictions for each tolerance.

In order to investigate which fusion techniques offer results that are statistically different, in terms of MAE, we performed paired *t-test* between all pairs. The *t-test* has also been utilized for the task of phonetic segmentation in (Park and Kim, 2007). In Table 4, the similarly coloured cells correspond to statistically equivalent results. Finally, the best segmentation accuracy for each tolerance of interest is indicated in bold.

Table 4

As showed in Table 4, the SVR followed by the LR(AIC) algorithm present notably better segmentation accuracy, when compared to the other fusion methods evaluated here. In particular, for the tolerance of 15-20 milliseconds, which is considered an acceptable limit for producing good quality synthetic speech (Matousek et al., 2003; Wang et al., 2004), the SVR method improved the overall segmentation accuracy by approximately 9% in terms of absolute performance. For small tolerances, the SVR fusion method offers results which improve the overall segmentation accuracy by more than 15%, when compared to the best performing BSE, i.e. HFCC-E with setup [1-4-CI]. For large tolerances, i.e. about ±30 milliseconds, the SVR and LR(AIC) methods improved the absolute segmentation accuracy by approximately 5%.

The M5' model trees and the MLP NN fusion improved the overall segmentation accuracy for all tolerances of interest, with the MLP NN presenting high RMSE, i.e. many large errors, comparing to SVR, LR(AIC) and M5' algorithms. On the contrary, the best-only selection, BEST, the averaging of the predictions, AVE, and the general fusion technique, GFT, did not improve the segmentation accuracy over the one of the best-performing BSEs.

All fusion methods were better or similar to the best performing HFCC-E segmentation engine for tolerances larger than 20 milliseconds. In this area big misalignments have been obliterated, which is in agreement with (Kominek and Black, 2004).

Although in earlier studies (Jarifi et al., 2008; Park and Kim, 2006) the GFT and BEST linear fusion methods were found to improve the segmentation accuracy on the single-speaker speech segmentation task, the experimental results obtained on the TIMIT database demonstrated that in the case of multiple speakers these methods do not offer improvement over the accuracy of the best-performing BSE. This is mainly owed to the mismatch between the train and test subsets of TIMIT (there is no speaker overlap between the train and test subsets), and the variations of the spectral characteristics of phones among the 630 speakers. These variations are both in the central areas of the phones and in the transitions between the phones. This mismatch between train and test data results in different performance of each BSE on the train and test subsets. Thus, the criterion for adjustment of the fusion parameters $\boldsymbol{p}(b)$ that is based on the BSEs' segmentation accuracy over the training data, which was used in these earlier studies is not a successful strategy when segmentation of speech recordings from multiple speakers is needed. Moreover, the use of hard decisions, as in the BEST fusion method, eliminates the predictions of the BSEs with worse segmentation accuracy. However these predictions still include complementary information that can be exploited for improving the overall segmentation accuracy.

As it was demonstrated by the results presented in this section, when phonetic segmentation has to be performed on speech recordings that include different genders, dialects, multiple speakers, etc, such as in TIMIT database, the adjustment of the fusion parameters $\boldsymbol{p}(b)$ is more efficient, when it is based directly on utilizing the boundary predictions on training instances, rather than based on the accuracy of each BSE on that training data. Furthermore, the experimental results on the TIMIT data indicated that the SVR and LR(AIC) fusion methods offer a significant advantage over the linear fusion methods used so far, as well as over some non-linear regression algorithms, and significantly improve the overall phonetic segmentation accuracy. This improvement derives from the ability of the regression algorithms to capture biases between the real and the predicted from the BSEs boundary positions, to learn systematic errors of each BSE in specific phonetic transition types and finally, to better model systematic misalignments in boundary position predictions between different BSEs.

5. Conclusion

In this article we proposed the use of a fusion scheme, based on regression analysis, for the task of phonetic segmentation of speech waveforms. This scheme utilizes numerous independent HMM-based segmentation engines, with different speech parameterizations, different number of HMM states, different number of Gaussian mixtures per state, and context dependent and independent models, to produce multiple predictions of boundary positions. These predictions were utilized as input to the proposed fusion scheme.

Various regression algorithms were evaluated with respect to their capability to provide precise estimations of the phonetic transition positions. The experimental results demonstrated significant improvement in the absolute segmentation accuracy for the support vector regression method, when compared to the best performing baseline segmentation engine. Specifically, in all the evaluated tolerances the segmentation accuracy was significantly improved, while in the most widely used tolerance of 20 milliseconds the performance was improved by approximately 9% in terms of absolute segmentation accuracy. In addition, the mean absolute error was decreased by approximately 33% while the root mean squared error was reduced by 27%. The linear regression method was found out to perform slightly worse than the support vector regression method, but also improved the overall performance by approximately 8%. The experimental results demonstrated that, in the multiple speaker case, the direct use of the boundary prediction instances resulting from individual segmentation engines on a training dataset is a better criterion than using the accuracies of the segmentation engines on the training dataset for adjusting the parameters of the fusion scheme.

Finally, the support vector regression fusion approach proved to combine segmentation predictions more successfully, i.e. to provide more precise phonemic boundary position predictions, when compared to various linear methods reported so far.

6. Acknowledgement

The authors would like to thank the anonymous reviewers for their creative comments and suggestions on earlier version of this manuscript that assisted us to significantly improve the quality of this study.

7. References

Acero A. The role of phoneticians in speech technology. In: Bloothooft G, Hazan V, Huber D, Llisterri J, editors. European Studies in Phonetics and Speech Communication, OTS Publications; 1995.

Adami AG, Hermansky H. Segmentation of speech for speaker and language recognition. In: Proc. 8th European Conf. on Speech Communication and Technology (EUROSPEECH 2003); 2003. p. 841-844.

Adell J, Bonafonte A, Gomez JA, Castro MJ. 2005, Comparative study of automatic phone segmentation methods for TTS, In: Proc. 2005 IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2005); 2005. p. 309-312.

Akaike H. A new look at the statistical model identification, IEEE Transactions on Automatic Control 1974;19(6): 716-723.

Aversano G, Esposito A, Esposito A, Marinaro M. A new text-independent method for phoneme segmentation. In: Proc. 44th IEEE Midwest Symposium on Circuits and Systems; 2001. Vol. 2, p. 516-519.

Bajwa RS, Owens RM, Kelliher TP. Simultaneous speech segmentation and phoneme recognition using dynamic programming. In: Proc. 1996 IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1996); 1996. Vol. 6, p. 3213-3216.

Baum LE, Petrie T, Soules G, Weiss N. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. Annals of Mathematical Statistics 1970;41(1):164-171.

Brugnara F, Falavigna D, Omologo M. Automatic segmentation and labeling of speech based on hidden Markov models. Speech Communication 1993; 12:357-370.

Campbell N, Black AW. Prosody and the selection of source units for concatenative synthesis. In: Van Santen JPH, Sproat RW, Olive JP, Hirschberg J, editors. Progress in speech synthesis. Springer; 1997. p. 279-292.

Chang CC, Lin CJ. Training v-support vector regression: theory and algorithms. Neural Computation 2002;14(8):1959-1977.

Chester DL. Why Two Hidden Layers are Better than One. In: Proc. International Joint Conference on Neural Networks; 1990. Vol. 1, p. 265-268.

Dalsgaard P, Andersen O, Barry W. Multi-lingual label alignment using acoustic-phonetic features derived by neural-network technique. In: Proc. 1991 IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1991); 1991. Vol. 1, p. 197-200.

Davis SB, Mermelstein P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE T Acoust., Speech Signal P. 1980;28(4):357-366.

Deller J, Hansen J, Proakis J. Discrete-time processing of speech signals. New York: Macmillan Publishing; 1993.

Dutoit T. An introduction to text-to-speech synthesis. Kluwer Academic Publishers; 1997.

ETSI. ETSI ES 201 108, V1.1.2 (2000-4). ETSI Standard: Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Extended advanced front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction algorithm; April 2000. Chapter 4, p. 8-11.

ETSI. ETSI ES 202 050, V1.1.5 (2007-1). ETSI Standard: Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Extended advanced front-end feature extraction algorithm; Compression algorithms; Back-end speech reconstruction algorithm; January 2007. Section 5.3, p. 21-24.

Farooq O, Datta S. Mel filter-like admissible wavelet packet structure for speech recognition. IEEE Signal Proc. Let. 2001;8(7):196-198.

Garofolo J. Getting Started with the DARPA-TIMIT CD-ROM: An acoustic phonetic continuous speech database. National Institute of Standards and Technology (NIST), Gaithersburgh, MD, USA; 1988.

Grayden DB, Scordilis MS. Phonemic segmentation of fluent speech. In: Proc. 1994 IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1994); 1994. Vol. 1, p. I/73-I/76.

Hermansky H. Perceptual linear predictive (PLP) analysis for speech. J. Acoust. Soc. Am. 1990, 87(4):1738-1752.

Iwano K, Yamada M, Togawa T, Furui S. Prosody control for HMM-based Japanese TTS. In: Narayanan S, Alwan A, editors: Text to speech synthesis: new paradigms and advances. New Jersey: Prentice Hall; 2004. p.155-173.

Jarifi S, Pastor D, Rosec O. A fusion approach for automatic speech segmentation of large corpora with application to speech synthesis. Speech Communication 2008;50:67-80.

Keshet J, Shalev-Shwartz S, Singer Y, Chazan D. A large margin algorithm for speech-to-phoneme and music-to-score alignment. IEEE Trans. Audio, Speech, and Language Processing 2007;15(8): 2373-2382.

Kim YJ, Conkie A. Automatic segmentation combining an HMM-based approach and spectral boundary correction. In: Proc. 7th Internat. Conf. on Spoken Language Processing (ICSLP 2002); 2002. p. 145-148.

Kominek J, Bennet C, Black AW. Evaluating and correcting phoneme segmentation for unit selection synthesis. In Proc. Eurospeech 2003; 2003. p. 313-316.

Kominek J, Black A. A family-of-models approach to HMM-based segmentation for unit selection speech synthesis. In: Proc. 8th Internat. Conf. on Spoken Language Processing (ICSLP 2004); 2004. p. 1385-1388.

Lee KF, Hon HW. Speaker-independent phone recognition using hidden Markov models. IEEE Trans. Acoustics, Speech and Signal Processing 1989;37(11):1641-1648.

Lee KS. MLP-based phone boundary refinement for a tts database, IEEE Trans. Acoustics, Speech, and Language Processing 2006;14(3):981-989.

Lin CY, Roger Jang JS. Automatic phonetic segmentation by score predictive model for the corpora of mandarin singing voices, IEEE Trans. Audio, Speech, and Language Processing 2007;15(7):2151-2159.

Ljolje A, Hirschberg J, Van Santen JPH. Automatic speech segmentation for concatenative inventory selection. In: Van Santen JPH, Sproat RW, Olive JP, Hirschberg J, editors. Progress in speech synthesis. Springer; 1997. p. 304-311.

Ljolje A, Riley MD. Automatic segmentation and labeling of speech. In: Proc. 1991 IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1991); 1991. Vol. 1, p. 473-476.

Lo HY, Wang HM. Phonetic boundary refinement using support vector machine. In: Proc. 2007 IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2007); 2007. p. IV-933-IV-936.

Malfrere F, Deroo O, Dutoit T, Ris C. Phonetic alignment: speech synthesis-based vs. Viterbi-based. Speech Communication 2003;40:503-515.

Matousek J, Tihelka D, Psutka J. Automatic segmentation for Czech concatenative speech synthesis using statistical approach with boundary-specific correction. In: Proc. 8th European Conf. on Speech Communication and Technology (EUROSPEECH 2003); 2003. p. 301-304.

Moore BCJ, Glasberg BR. Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. Journal of the Acoustical Society of America 1983;74(3):750-753.

Moore BCJ. An introduction to the psychology of hearing. Academic Press, London, 5th Ed.; 2003.

Mporas I, Ganchev T, Fakotakis N. A hybrid architecture for automatic segmentation of speech waveforms. In: Proc. 2008 IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2008); 2008. p. 4457-4460.

Nogueira W, Giese A, Edler B, Buchner A. Wavelet Packet Filterbank for Speech Processing Strategies in Cochlear Implants. In: Proc. IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2006); 2006. Vol. 5, p. 121-124.

Park SS, Kim NS. Automatic speech segmentation based on boundary-type candidate selection. IEEE Signal Processing Letters 2006;13(10):640-643.

Park SS, Kim NS. On using multiple models for automatic speech segmentation. IEEE Trans. Audio, Speech, and Language Processing 2007;15(8):2202-2212.

Paulo S, Oliveira LC. DTW-based phonetic alignment using multiple acoustic features. In: Proc. 8th European Conf. on Speech Communication and Technology (EUROSPEECH 2003); 2003. p. 309-312.

Pauws S, Kamp Y, Willems L. A hierarchical method of automatic speech segmentation for synthesis applications. Speech Communication 1996;19:207-220.

Pellom BL, Hansen JHL. Automatic segmentation of speech recorded in unknown noisy channel characteristics. Speech Communication 1998;25:97-116.

Petek B, Andersen O, Dalsgaard P. On the robust automatic segmentation of spontaneous speech. In: Proc. 4th Internat. Conf. on Spoken Language Processing (ICSLP 1996); 1996. Vol. 2, p. 913-916.

Quilan JR. Learning with continuous classes. In Proc. 5th Australian Joint Conference on Artificial Intelligence. World Scientific; 1992. p. 343-348.

Sagisaka Y, Campbell WN, Higuchi N. Computing prosody: computational models for processing spontaneous speech. Springer-Verlag; 1997.

Sarikaya R, Hansen JHL. High resolution speech feature parameterization for monophone-based stressed speech recognition. IEEE Signal Proc. Let. 2000;7(7):182-185.

Scholkopf B, Smola A, Williamson R, Bartlett PL. New support vector algorithms. Neural Computation 2000;12(5):1207-1245.

Sethy A, Narayanan S. Refined speech segmentation for concatenative speech synthesis. In: Proc. 7th Internat. Conf. on Spoken Language Processing (ICSLP 2002); 2002. p. 149-152.

Skowronski MD, Harris JG. Exploiting independent filter bandwidth of human factor cepstral coefficients in automatic speech recognition. J. Acoust. Soc. Am. 2004;116(3):1774-1780.

Slaney M. Auditory toolbox. Version 2. Technical Report #1998-010. Interval Research Corporation; 1998.

Svendsen T, Soong FK. On the automatic segmentation of speech signals. In: Proc. 1987 IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 1987); 1987. Vol. 12, p. 77-80.

Toledano DT, Gomez LAH, Grande LV. Automatic phonetic segmentation. IEEE Trans. Speech and Audio Processing 2003;11(6):617-625.

van Hemert JP. Automatic segmentation of speech. IEEE Trans. Signal Processing 1991;39(4):1008-1012.

Vapnik V. Statistical learning theory. New York: Wiley; 1998.

Viterbi A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Information Theory 1967;13(2):260-269.

Wang L, Zhao Y, Chu M, Zhou J, Cao Z. Refining segmental boundaries for TTS database using fine contextual-dependent boundary models. In: Proc. 2004 IEEE Internat. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2004); 2004. Vol. 1, p. 641-644.

Wang Y, Witten IH. Inducing model trees for continuous classes. In Proc. 9th European Conference on Machine Learning - Poster Papers; 1997. p.128-137.

Wightman CW, Talkin DT. The aligner: text-to-speech alignment using Markov models. In: Van Santen JPH, Sproat RW, Olive JP, Hirschberg J, editors. Progress in speech synthesis. Springer; 1997. p. 313-323.

Witten IH, Frank E. Data mining: practical machine learning tools and techniques, 2nd ed. San Francisco: Morgan Kaufmann; 2005.

Young S, Evermann G, Gales M, Hain T, Kershaw D, Liu X, Moore G, Odell J, Ollason D, Povey D, Valtchev V, Woodland P. The HTK Book (for HTK Version 3.4). Cambridge University Engineering Department; 2006.

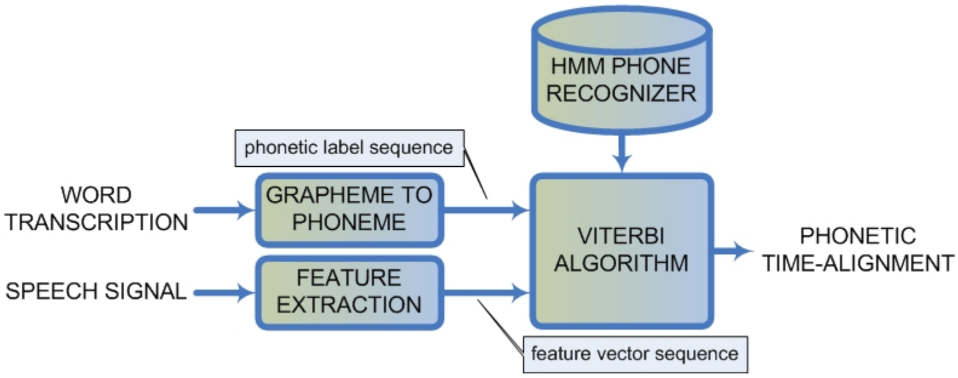

Figure 1. Block diagram of the HMM-based phonetic segmentation method.

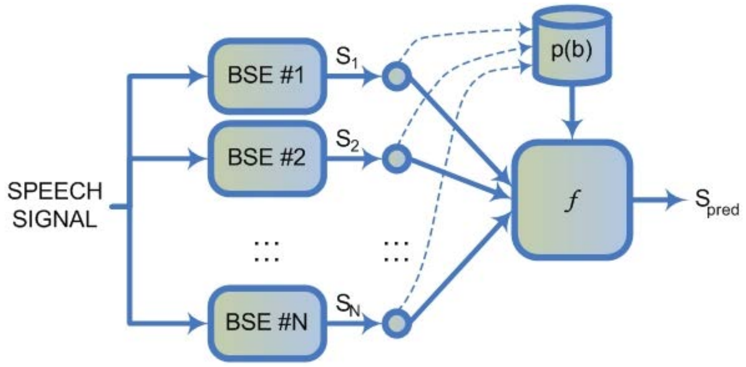

Figure 2. Block diagram of the regression fusion of multiple baseline segmentation engines (BSEs). The dashed arrows indicate the use of the BSE predictions for the computation of the $\boldsymbol{p}(b)$ parameters in the training phase.

Table 1. Segmentation accuracy (in percentages) for the evaluated CI baseline segmentation engines (BSEs). Mean absolute error (MAE) and root mean squared error (RMSE) are given in milliseconds.

[m-s-CI/CD]	BSE	*t*≤5ms	*t*≤10ms	*t*≤15ms	*t*≤20ms	*t*≤25ms	*t*≤30ms	MAE(ms)	RMSE(ms)
[1-3-CI]	HFCC-E	**29.85**	49.14	66.92	78.57	84.70	88.20	17.05	40.77
	LFCC	19.29	38.14	57.31	72.84	81.25	85.40	19.63	41.60
	MFCC	25.03	44.84	62.12	76.29	83.89	87.65	17.55	37.75
	MWP-ACE	18.94	36.31	55.32	70.19	80.92	85.32	19.76	36.89
	PLP	26.79	46.68	63.71	77.29	84.19	87.87	18.21	43.21
	SBC	21.65	40.73	59.99	74.07	83.49	87.38	19.48	51.27
	WPF	17.78	36.40	56.89	72.30	83.39	87.46	20.15	44.51
[1-4-CI]	HFCC-E	28.92	**50.21**	66.54	78.54	85.02	**88.74**	**14.98**	25.67
	LFCC	20.88	40.96	59.86	74.14	82.44	86.43	18.69	39.59
	MFCC	26.35	47.13	64.25	77.54	84.80	88.63	15.50	26.80
	MWP-ACE	22.16	41.26	60.30	73.33	81.94	86.01	18.50	36.88
	PLP	27.61	48.71	65.34	77.99	84.95	88.62	15.30	26.85
	SBC	21.86	42.65	62.91	76.16	84.54	88.19	16.33	26.15
	WPF	18.15	37.47	58.38	73.42	83.34	87.42	17.77	30.94
[2-3-CI]	HFCC-E	28.40	49.63	**68.38**	**79.41**	**85.22**	88.50	15.03	25.21
	LFCC	21.78	41.70	60.35	73.25	81.90	85.97	19.28	29.23
	MFCC	22.82	42.90	61.20	74.51	83.59	87.41	16.59	25.87
	MWP-ACE	21.58	40.54	59.36	72.87	82.97	86.97	19.86	33.56
	PLP	24.71	45.38	64.14	76.91	84.57	88.02	16.02	27.00
	SBC	20.53	40.12	60.42	74.27	83.31	87.19	18.92	32.23
	WPF	20.80	41.11	62.20	75.71	84.38	88.09	19.10	34.94
[2-4-CI]	HFCC-E	21.63	42.51	62.15	75.79	83.60	87.39	16.47	25.16
	LFCC	18.68	37.01	56.59	71.60	81.20	85.56	18.27	27.98
	MFCC	20.75	40.79	60.29	74.61	82.82	86.90	17.00	27.03
	MWP-ACE	21.26	40.63	59.75	73.67	82.82	86.89	17.35	27.88
	PLP	22.21	42.80	62.32	75.81	83.48	87.26	16.44	26.40
	SBC	19.99	39.10	59.92	74.55	83.48	87.54	16.84	24.74
	WPF	17.38	34.74	55.30	71.20	82.30	86.86	18.29	30.01
[4-3-CI]	HFCC-E	25.84	47.27	66.27	77.93	84.19	87.66	15.22	25.08
	LFCC	22.20	42.40	60.79	73.99	82.43	86.25	19.11	39.96
	MFCC	23.07	43.41	61.80	74.90	83.13	86.89	16.11	24.63
	MWP-ACE	21.50	40.26	58.84	72.53	82.61	86.48	17.52	31.63
	PLP	24.44	45.32	63.71	76.56	83.69	87.14	15.64	24.13
	SBC	19.81	38.96	58.84	72.83	82.41	86.40	16.97	25.83
	WPF	20.80	40.47	61.14	74.92	83.86	87.56	16.21	24.22
[4-4-CI]	HFCC-E	19.85	40.03	59.70	73.08	80.27	84.33	17.41	25.83
	LFCC	18.91	37.79	57.37	71.95	80.04	84.21	17.98	26.81
	MFCC	19.83	39.82	59.11	72.96	80.33	84.56	17.40	25.88
	MWP-ACE	20.61	39.73	59.30	73.25	82.27	86.12	17.18	26.38
	PLP	20.71	41.15	60.50	73.86	80.83	84.92	17.02	25.37
	SBC	18.70	37.64	58.55	72.96	81.96	85.99	17.59	30.86
	WPF	16.29	33.75	54.40	70.14	81.48	85.88	17.89	25.57
[6-3-CI]	HFCC-E	24.53	46.25	65.68	77.60	84.34	87.76	15.18	**23.48**
	LFCC	21.59	41.14	59.81	73.03	82.02	86.06	18.16	38.37
	MFCC	22.52	42.45	60.96	74.12	82.74	86.64	16.32	24.69
	MWP-ACE	22.20	40.57	58.77	72.57	82.69	86.79	16.88	26.01
	PLP	23.49	43.88	62.55	75.72	83.55	87.27	15.78	24.07
	SBC	20.32	39.07	58.56	72.58	82.23	86.39	16.87	24.92
	WPF	21.20	40.39	60.27	74.19	83.45	87.26	16.32	24.32
[6-4-CI]	HFCC-E	19.05	38.50	58.84	72.67	79.99	84.02	17.72	26.49
	LFCC	19.17	38.51	58.09	72.30	79.44	83.62	18.20	29.03
	MFCC	19.09	38.75	58.41	72.58	80.22	84.44	17.60	26.01
	MWP-ACE	20.36	39.19	58.83	73.11	82.26	86.23	17.66	37.00
	PLP	20.24	40.85	60.54	74.08	81.10	85.17	16.95	25.27
	SBC	17.92	36.29	57.27	72.26	81.45	85.68	17.45	25.29
	WPF	16.06	33.54	54.30	70.10	81.01	85.52	18.03	25.75

Table 2. Segmentation accuracy (in percentages) for the evaluated CD baseline segmentation engines (BSEs). Mean absolute error (MAE) and root mean squared error (RMSE) are given in milliseconds.

[m-s-CI/CD]	BSE	*t*≤5ms	*t*≤10ms	*t*≤15ms	*t*≤20ms	*t*≤25ms	*t*≤30ms	MAE(ms)	RMSE(ms)
[1-3-CD]	HFCC-E	25.29	45.04	63.33	75.97	83.27	87.41	16.81	33.13
	LFCC	18.99	36.44	54.46	69.57	79.49	84.52	18.97	29.56
	MFCC	21.68	40.09	57.99	72.41	81.42	86.22	17.47	27.45
	MWP-ACE	17.78	33.95	53.18	68.67	80.25	85.20	19.60	32.55
	PLP	23.07	42.35	60.46	73.94	81.70	86.16	17.59	32.80
	SBC	19.77	37.34	56.98	71.78	82.72	87.35	17.66	27.65
	WPF	18.08	35.04	55.10	70.29	81.90	86.88	18.34	29.20
[1-4-CD]	HFCC-E	**27.09**	**46.98**	**64.48**	**76.09**	**83.77**	**87.87**	**15.64**	**25.28**
	LFCC	19.40	38.08	56.11	70.41	80.57	85.56	19.46	39.41
	MFCC	23.16	42.75	60.33	74.16	82.72	87.26	16.59	25.97
	MWP-ACE	21.58	40.37	58.93	72.19	81.90	86.21	18.33	35.21
	PLP	25.77	45.21	61.59	74.71	82.83	87.20	16.28	25.91
	SBC	21.43	41.16	60.57	74.36	83.75	88.02	16.70	26.05
	WPF	17.56	35.03	54.77	70.16	81.51	86.63	18.34	29.00
[2-3-CD]	HFCC-E	25.25	44.43	62.49	74.30	82.51	86.89	16.51	26.69
	LFCC	19.87	37.44	54.89	68.93	79.09	84.18	19.42	35.66
	MFCC	21.63	39.28	56.36	70.07	79.74	85.00	17.88	27.12
	MWP-ACE	18.63	34.73	53.23	68.32	80.32	85.55	19.18	31.48
	PLP	23.39	42.13	59.56	72.11	80.50	85.36	17.47	27.98
	SBC	19.80	36.93	55.70	70.37	81.75	86.69	17.87	27.06
	WPF	18.14	35.01	54.05	69.11	81.42	86.37	18.56	29.65
[2-4-CD]	HFCC-E	25.32	45.04	62.57	74.59	82.55	86.88	16.39	26.97
	LFCC	18.31	35.95	54.15	68.62	79.06	84.27	18.99	30.96
	MFCC	21.74	40.67	58.25	71.60	80.99	85.88	17.37	26.51
	MWP-ACE	20.77	39.19	57.98	71.57	81.53	86.06	17.88	28.88
	PLP	24.26	43.29	59.81	72.32	81.39	86.00	16.93	26.21
	SBC	20.34	39.05	58.68	72.59	82.41	87.06	17.28	26.56
	WPF	15.78	32.10	51.68	67.58	79.38	84.77	19.32	29.47
[4-3-CD]	HFCC-E	24.52	42.80	59.75	71.76	80.33	85.02	17.21	27.89
	LFCC	19.77	36.97	53.45	67.03	77.40	82.69	19.12	29.32
	MFCC	21.24	38.37	54.82	68.20	77.85	83.25	18.34	27.61
	MWP-ACE	19.00	35.13	53.17	67.84	79.63	84.67	18.89	30.81
	PLP	22.74	40.70	57.63	70.20	78.90	83.80	17.86	27.78
	SBC	19.56	36.38	54.53	68.54	79.86	84.97	18.17	27.35
	WPF	18.34	35.04	53.37	67.84	79.69	84.86	19.03	34.14
[4-4-CD]	HFCC-E	23.89	43.06	60.67	72.74	80.82	85.31	16.83	26.88
	LFCC	17.75	34.66	52.60	66.87	77.08	82.53	19.16	28.13
	MFCC	20.66	38.60	55.84	69.03	78.10	83.27	18.21	27.80
	MWP-ACE	19.98	37.91	56.37	70.02	79.57	84.44	18.20	29.13
	PLP	22.62	41.44	58.04	70.16	78.75	83.78	17.66	27.27
	SBC	19.20	37.33	56.45	70.22	80.12	84.87	18.00	27.67
	WPF	15.62	31.56	50.55	66.17	77.94	83.42	19.44	29.11
[6-3-CD]	HFCC-E	24.03	41.87	58.77	70.98	79.73	84.54	17.47	27.88
	LFCC	19.56	36.55	52.65	66.03	76.45	81.85	20.04	39.55
	MFCC	20.97	37.60	54.14	67.52	77.25	82.74	18.68	28.37
	MWP-ACE	18.99	35.16	53.09	67.91	79.57	84.87	18.62	28.70
	PLP	22.11	40.06	56.85	69.10	78.00	83.17	18.23	28.30
	SBC	19.50	36.45	54.23	68.26	79.56	84.73	18.22	27.22
	WPF	18.12	34.71	52.80	67.51	79.41	84.78	18.96	32.63
[6-4-CD]	HFCC-E	23.40	42.60	60.45	73.05	80.52	84.97	17.10	28.70
	LFCC	17.60	34.50	52.17	66.58	76.25	81.86	19.39	28.30
	MFCC	19.84	37.77	55.12	68.27	77.44	82.76	18.59	29.00
	MWP-ACE	19.72	37.54	56.04	69.69	79.11	84.15	18.32	29.26
	PLP	22.02	40.74	57.13	69.54	78.03	83.07	17.98	27.58
	SBC	18.97	36.54	55.64	69.88	79.68	84.55	18.13	27.29
	WPF	15.65	31.70	50.51	66.22	77.72	83.21	19.46	28.95

Table 3. Best BSE per phonetic transition type for 20 milliseconds tolerance. Rows and columns indicate the left (L) and right (R) context of the phonetic boundary, respectively.

L\R	AFF	FRI	NAS	GLI	SIL	STO	VOW
AFF	MFCC[1-3-CI]	PLP[6-3-CD]	MFCC[1-3-CD]	MWP-ACE[4-4-CD]	MWP-ACE[4-4-CI]	HFCC-E[1-3-CD]	PLP[6-4-CD]
FRI	HFCC-E[1-3-CD]	HFCC-E[4-4-CI]	HFCC-E[1-4-CD]	HFCC-E[1-4-CI]	MFCC[2-3-CI]	SBC[1-3-CD]	HFCC-E[2-4-CD]
NAS	HFCC-E[1-4-CI]	HFCC-E[2-3-CD]	HFCC-E[1-4-CD]	HFCC-E[2-4-CD]	HFCC-E[4-4-CI]	PLP[4-3-CI]	PLP[2-4-CD]
GLI	WPF[1-3-CD]	WPF[2-4-CD]	MFCC[2-4-CD]	PLP[2-4-CD]	HFCC-E[4-4-CI]	WPF[2-4-CD]	PLP[6-4-CD]
SIL	HFCC-E[1-3-CI]	HFCC-E[1-3-CI]	MWP-ACE[1-4-CI]	HFCC-E[1-3-CI]	-	HFCC-E[1-3-CI]	HFCC-E[1-3-CI]
STO	MWP-ACE[1-3-CI]	HFCC-E[4-4-CD]	HFCC-E[1-4-CI]	HFCC-E[1-4-CI]	LFCC[4-3-CI]	WPF[1-4-CI]	HFCC-E[2-4-CD]
VOW	HFCC-E[1-3-CI]	MWP-ACE[1-4-CD]	PLP[1-4-CD]	PLP[1-4-CD]	SBC[6-4-CI]	HFCC-E[2-4-CI]	PLP[1-4-CI]

Table 4. Phone segmentation using regression fusion algorithms for 112 BSEs.

Fusion Method	***t*≤5ms**	***t*≤10ms**	***t*≤15ms**	***t*≤20ms**	***t*≤25ms**	***t*≤30ms**	**MAE(ms)**	**RMSE(ms)**
SVR	**45.30**	**71.43**	**82.28**	**88.18**	**91.68**	**94.01**	**10.01**	**17.15**
LR(AIC)	43.01	68.74	80.52	87.12	91.03	93.54	10.44	17.47
M5'	39.85	63.05	78.07	84.31	88.37	91.28	11.95	19.82
MLP	34.27	57.92	72.23	80.40	86.64	89.85	13.91	27.56
GFT [$f(x)$=1/(1-x)]	20.76	41.98	62.26	76.83	85.40	89.48	15.49	23.24
GFT [$f(x)$=x]	21.48	41.89	61.79	75.86	84.47	88.81	15.90	24.16
BEST	24.69	46.27	65.26	76.41	86.05	90.23	15.84	33.78
AVE	20.21	40.50	60.75	75.22	84.25	88.80	15.96	23.59
No Fusion	29.85	50.21	68.38	79.41	85.22	88.74	14.98	23.48

List of Figures

List of Tables