EXPLICIT SEGMENTATION OF SPEECH SIGNALS USING BACH FILTER-BANKS

Ranjani H G[1], Ananthakrishnan G[2], A G Ramakrishnan[3]

Department of Electrical Engineering
Indian Institute of Science
Bangalore – 560012, INDIA
{ranjani[1], ramkiag[3]} @ee.iisc.ernet.in, gananth[2]@gmail.com

ABSTRACT

For synthesizing high quality speech, a concatenative Text-To-Speech system requires a large number of well annotated segments at the phone level. Manual segmentation, though reliable, is tedious, time consuming and can be inconsistent. This correspondence presents an automated phone segmentation algorithm that force aligns the phonetic transcriptions with the utterances of the corresponding Indian language sentences. The algorithm uses the distance function obtained from the output of the recently proposed Bach scale filter bank and the statistical knowledge of the lengths of the phones to force align the boundaries between successive stop consonants. Preliminary results for Hindi database shows that 85.2% of the boundaries detected by the algorithm are well within 20 ms of the manually segmented boundaries. The misclassified frames (20 ms) per sentence or the Frame Error Rate is 20.4%.

1. INTRODUCTION

Accurate time markers indicating the beginning and ending times of a speech sound (phone) in a spoken sentence are crucial for building a high quality Text to Speech (TTS) system. Segmentation is the process of getting these time markers. Such segmented and labeled phones are used to create a new unit inventory meant for concatenative speech synthesis and also for prosody modeling. The quality of segmentation is critical because an error in either segmentation or labeling can give rise to an audible error in the synthesized speech.

Manual segmentation is a conventional technique for segmenting speech. However, it turns out to be monotonous, time consuming and at times inconsistent. To circumvent these drawbacks, it becomes necessary to automate the process of segmentation.

To develop a concatenative TTS system for any Indian language, a speech corpus is created by recording from a single speaker, utterance of a large number of sentences covering various acoustic and phonetic contexts. The phonetic transcription is obtained by mapping the graphemes to the corresponding phonemes using a grapheme to phoneme (G2P) converter. Thus, the phonetic labels of segmented speech are obtained from the phonetic transcription. The task therefore is to align these phonetic transcriptions to the actual boundaries in the corresponding speech utterances. This is an explicit segmentation problem, which differs from implicit automated segmentation where there is no a priori knowledge of the phonetic transcription, thus potentially increasing the number of "inserted" and "deleted" boundaries.

As far as the work on automated explicit segmentation is concerned, Abhinav et al proposed refining context dependent phone based HMM (CDHMM) giving good boundary accuracy [1]. Neural network trees with known number of sub-word units have also been used for segmentation [2]. However, the need to develop TTS in multiple Indian languages and the non-availability of large speech corpora for Indian languages are the major constraints, which limit the use of these training based segmentation techniques.

A recent segmentation work using the Bach scale filter bank has the advantage of being language independent and training free [3], [4]. We have extended the ideas of this work for our explicit segmentation algorithm.

2. SEGMENTATION USING BACH FILTER BANK

In this method, speech signal is treated as non-stationary. A constant Q filter bank is formulated, motivated by the perception of music. This bank has 12 filters in every octave, wherein the centers of successive filters are separated by a ratio of $2^{(1/12)}$.

Speech signal sampled at 16 kHz is passed through this filter bank and the set of outputs of the bank at any instant of time is treated as the feature vector. Here, speech is not presented to the filter banks as short segments (frames) as in the usual framework of quasi-stationary signal. Rather, we get feature vectors for every instant of time. Now, the mean of the log of the feature vectors in each 15 ms window is taken and the Euclidean distance between successive means is calculated. Seen as a 2-class problem, the distance between the means should peak if the feature vectors in the adjacent windows belong to different phoneme classes. The

distance measure used is referred to as the Euclidean Distance between Mean Log (EDML) feature vectors.

Figure 1 displays the plot of the values of the feature EDML as a function of time for part of a Hindi word utterance. We can see that the peaks of this function either coincide with or are close to the manually marked phone boundaries of the uttered word.

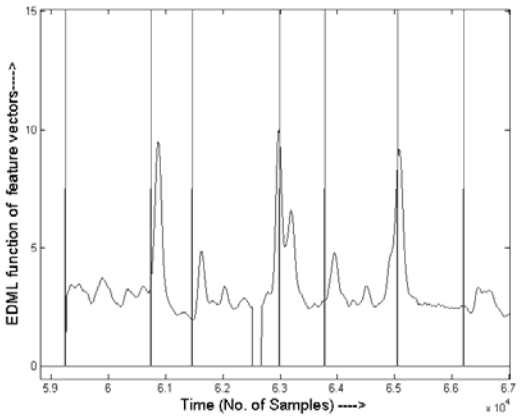

Figure 1. The plot of EDML against time for a portion of Hindi utterance ("satypar"). The vertical lines denote the manually segmented phone boundaries

This method gives 86.4% accuracy (automated boundary within 20 ms of a manual boundary), 21.4% insertions and 3.2% deletions for Hindi database, 81.9% accuracy, 15.3% deletions and 23.7% insertions for Tamil database, 82.5% accuracy, 22.3% deletions and 18.9% insertions for TIMIT database [5].

3. EXPLICIT SEGMENTATION

The proposed algorithm makes use of the statistical knowledge of the durations of the phones. The major disadvantage of forcing boundary alignments on the entire speech waveform is that the boundary error gets accumulated. To avoid propagating the boundary errors from the start of the sentence to the end of the same, we can force boundaries for the phones between two phone classes at a time. Armed with the phonetic transcription, the first stage of the algorithm detects a phone class but is constrained to be a training free algorithm.

In order to be detected, fricatives, vowels, nasals, diphthongs, nasal vowels and glides need some stored form of features. Also, different phones in each class require different features. However, as described below, stop consonants can be detected without storing any features. Thus, we follow hierarchical segmentation, where the stop consonants in a sentence are first located, and then the phones occurring between successive stop consonants are segmented.

The first frame (10 ms) of any speech sentence is predominantly silence. However, stop consonants can either be voiced or unvoiced. To remove the low frequency components that are present in the closure region of a voiced stop consonant, the speech signal is high-pass filtered with a Bessel filter with the lower cutoff frequency of 400 Hz (the voice bar of voiced stops extends roughly till 400 Hz). Now, MFCCs of all the frames of the filtered speech are calculated. The Euclidean distance is computed between the MFCC of the first frame of the sentence and the MFCC of every other frame. If this distance drops below a threshold value for a minimum of 3 consecutive frames (the minimum duration of a stop consonant is assumed to be roughly 30 ms), then it implies that the corresponding region may contain the silence part of a stop consonant or a silence region of speech or a combination of both. The frame within this region having the minimum distance from the first silence frame is surely a stop consonant (or silence or both) frame. Preliminary tests on 100 sentences from Hindi database give a stop consonant detection accuracy of 87% with 20% insertions.

The number of regions involving actual silence between words and the closure regions of the stop consonants can be known from the phonetic transcription. Using this, the number of silence regions to be detected can be forced. In this case, the equal error rate (i.e., the number of insertions equals number of deletions) is 11.3% for 100 sentences in Hindi and 15% for 50 sentences of TIMIT database. This performance is of the same order as the stop detection accuracy proposed in [6], [7]. Figure 2 illustrates the stop consonants (silence regions) detected by the above algorithm in a portion of a Hindi utterance.

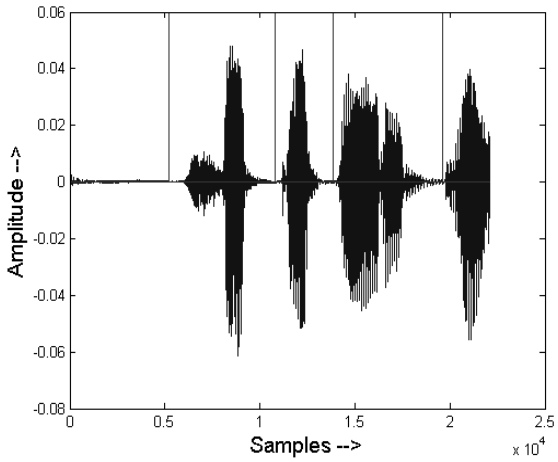

Figure 2. Portion of a Hindi speech utterance – "satyapardriRh". The utterance has a silence region at the start of the sentence and 3 stop consonants (/t/, /p/ and /d/). The vertical lines denote the start of the frame classified as sure stop consonants by the proposed algorithm.

The rest of the discussion in this work, assumes an error free stop consonant detection, and attempts to segment

the individual phones between 2 successive correctly detected stop consonants.

Using the Bach scale filter bank, the EDML function is calculated for the speech signal between every successive pair of stop consonants. By incorporating the knowledge of the regions of stop consonants, the end of the first stop consonant (b1) and the start of the next stop consonant (b2) can be found out using the energy change in the signal. This is illustrated for a portion of Hindi speech waveform in Figure 3.

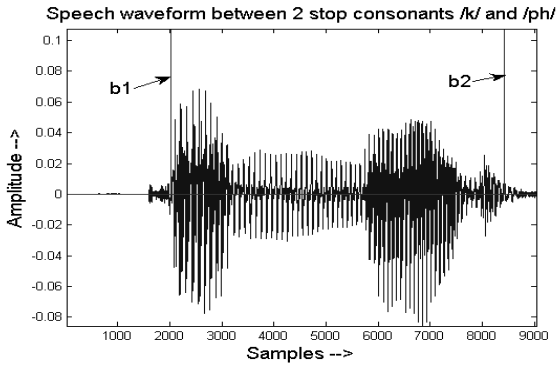

Figure 3. Identification of the region between the end of one stop consonant and the beginning of the next, in a portion of a Hindi speech waveform for the phone sequence /k/,/a/,/nl/,/a/,/r/,/ph/.

Consider a rectangular window of size δ times the standard deviation of the duration of the next phone, centered at b1+x where x is the mean duration of the next phone. Within this window, the maximum of the EDML function, $EDML_{max}$ is computed and all the peaks greater than $\alpha*EDML_{max}$ are detected. If the number of such peaks exceeds λ_1 or is less than λ_2 (where $\lambda_1 > \lambda_2$), then the best λ_1 possible peaks within that window are chosen. Again, another rectangular window, of size δ times the standard deviation of duration of the next phone, is centered at b1+x+y where y is the mean duration of the next phone. The same peak finding process is repeated for all the phones within the successive stop consonants. Hence the possible choices are $\geq \lambda_2$ and $\leq \lambda_1$ for every boundary to be detected between two successive stop consonants.

Figure 4 shows the EDML contour for the speech waveform in Figure 3. Also shown are the peaks detected for λ_2 chosen as 5.

Assuming a Gaussian PDF for the duration of the phones, the probability of transition to the next boundary is found out for each of these possible choices.

Now, the problem can be stated as: Find the best possible boundaries such that the product of the transition probabilities in that path is maximized. Equivalently, the sum of the negative log of the transition probabilities is minimized.

We have employed a graph theoretic approach to the problem, wherein each possible choice for a boundary is a node and the transition probability is the weight of an edge. This is illustrated in Figure 5.

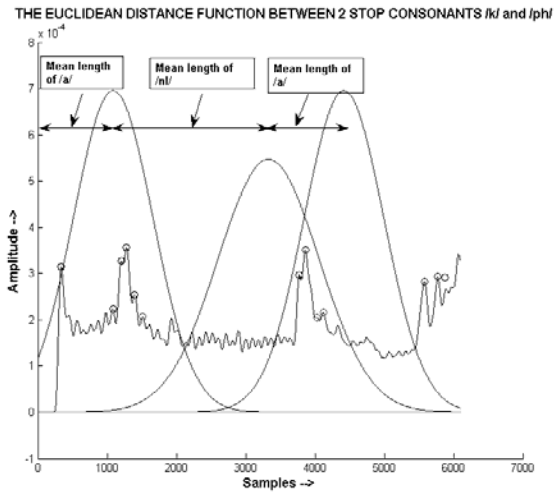

Figure 4. Contour of EDML values of the speech waveform shown in Figure 3. A window is centered at the point away from b1 by the mean duration of phone /a/. The circles indicate the peaks chosen. It can be seen that some peaks are common choices for both the successive phones.

Now, the best path that minimizes the cost of transition can be found. Since the start and end nodes (b1 and b2) are known, we can use Dijkstra's greedy algorithm. The best choices of nodes obtained from this algorithm are taken as the best possible boundaries within the 2 stop consonants. Figure 6 shows the best possible boundaries obtained using the above algorithm as against the manual boundaries.

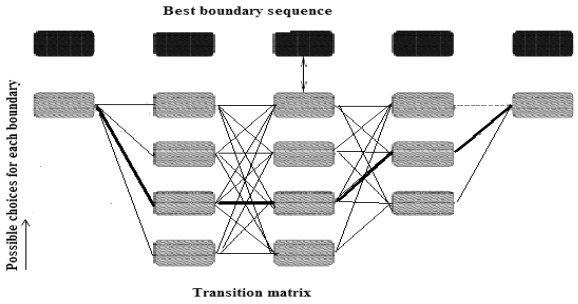

Figure 5. Nodes with transition probabilities. The first and last nodes are b1 and b2. The choices for a boundary are considered as a node. The edges indicate the negative log of transition probability.

The experiments were conducted on the Hindi database using the statistics of phone durations computed from the manually segmented database. Best results were obtained for the parameters $\delta = 8$, $\alpha = 0.1$, $\lambda_1 = 5$ and $\lambda_2 = 2$. Performance on 30 sentences from the Hindi database is 21.4% FER with a frame size of 20 ms and 20.4% FER with

a frame size of 25 ms. The experiments were repeated for statistics obtained from TIMIT database and the corresponding FER for utterances of a single speaker are 29.5% and 28.4%. The boundary error rate is 14.6% for Hindi and 19.4% for TIMIT database.

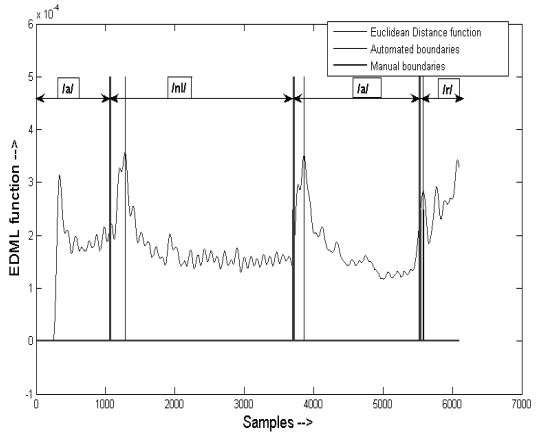

Figure 6. Boundaries between stop consonants /k/ and /ph/. The thick vertical lines are the manually marked boundaries and the thin vertical lines are the boundaries identified by the proposed algorithm.

4. CONCLUSION AND FUTURE WORK

The proposed method promises good segmentation provided the statistics of the phones are known. A final round of manual intervention is required. However, this manual intervention is now less tedious and less time consuming.

The mean durations of the phones are normalized to the speaker's rate of speech between the two stop consonants. Also, it was found that the frame error rate between the manual segmentations carried out independently by 2 trained segmenters is around 9%.

Future work can attempt using the statistics of phone durations of one language for segmenting speech of another language. Also, stop consonant detection method with a much higher accuracy needs to be developed.

5. REFERENCES

[1] Abhinav Sethy, Shrikanth Narayanan, "Refined Speech segmentation for concatenative speech synthesis", Proc. ICSLP-2002, pp 149-152.

[2] Sharma, M. Mammone, R., "Automatic speech segmentation using neural tree networks", Proc IEEE workshop on neural networks for signal processing, Sep 1995, pp 282-290.

[3] G. Ananthakrishnan, H. G. Ranjani, and A. G. Ramakrishnan, "Language independent automated segmentation of speech using Bach scale filter-banks", Proc. ICISIP –Dec. 2006, pp -115 – 120.

[4] G. Ananthakrishnan, H. G. Ranjani, and A. G. Ramakrishnan, "Comparative study of filter-bank mean-energy distance for automated segmentation of speech signals", Proc ICSCN –Feb. 2007, pp 06- 10.

[5] Ananthakrishnan G, "Music and speech analysis using the 'Bach' scale filter-bank", M.Sc (Engg) thesis, Indian Institute of Science, Apr -2007.

[6] F. Malbos, M. Baudry and S. Montresor , "Detection of stop consonants with the wavelet transform", Proc. IEEE-SP International symp. time-frequency and time-scale analysis, Oct. 1994, pp 612 – 615.

[7] P. Niyogia and M. M. Sondhi, "Detecting stop consonants in continuous speech", Proc. JASA Feb. 2002, pp 1063- 1075.

[8] L. Rabiner and B. H. Juang, "Fundamentals of speech recognition", Pearson Education Press, 1993 (AT&T).