"BLIND" SPEECH SEGMENTATION: AUTOMATIC SEGMENTATION OF SPEECH WITHOUT LINGUISTIC KNOWLEDGE

Manish Sharma

SpeakEZ, Inc., a subsidiary of T-NETIX, Inc.
371 Hoes Lane, Piscataway, NJ 08854
msharma@tnetix.com

Richard Mammone

CAIP Center, Rutgers University,
P O Box 1390, Piscataway, NJ 08855
mammone@caip.rutgers.edu

ABSTRACT

A new automatic speech segmentation procedure, called the "Blind" speech segmentation, is presented. This procedure allows a speech sample to be segmented into sub-word units without the knowledge of any linguistic information (such as, orthographic or phonetic transcription). Hence, this procedure involves finding the *optimal* number of sub-word segments in the given speech sample, before locating the sub-word segment boundaries.

1. INTRODUCTION

Segmentation and labeling of speech material according to phonetic or similar linguistic rules is a fundamental task in speech processing. A vast majority of the currently available speech processing systems, including, medium to large vocabulary speech recognition systems [1, 2], speaker recognition systems [3, 4, 5], and language identification systems [6], are designed based on subword acoustic units. Traditionally, the segmentation and labeling of speech data was done manually by a trained phonetician using the listening and visual cues. However, this manual procedure can be tedious, time consuming, subjective and error prone. Therefore, automatic speech segmentation procedures have been favored and used extensively in speech processing systems. Most automatic segmentation and labeling procedures use the associated linguistic knowledge, such as the spoken text and/or phonetic string.

The automatic speech segmentation algorithms found in the literature can be divided into two broad categories: Hierarchical and Non-hierarchical. The Hierarchical speech segmentation procedures involve a multi-level, fine-to-coarse, segmentation description; sometimes displayed in a tree-like fashion called dendogram. The best segmentation is then generally found as the best-path finding problem in the multi-level segmentation search space [7]. The Non-hierarchical speech segmentation algorithms attempt to locate the optimal segment boundaries by using Knowledge Engineering-based rule set [8, 9], or by minimizing a distortion metric using Dynamic programming-based methods [10], or by maximizing the score metric of acoustic models [11, 12].

The problem of automatic speech segmentation can be posed under two different scenarios. In the first kind, the phonetic transcription of the given speech, or alternately the number of phonemes present in it, is known. The segmentation algorithm is just required to optimally locate the sub-word segment boundaries. The algorithms cited in the above paragraph may be used to perform this type of speech segmentation. However, in the other type of speech segmentation, there is no linguistic knowledge about the given speech data. The segmentation algorithm, therefore, is required to determine the optimal number of sub-word units present in the given speech sample, as well as their boundary locations, based on the acoustic cues only. This type of segmentation will be referred to as the "blind" speech segmentation procedure. A novel "blind" speech segmentation algorithm is presented in detail in the following section.

2. "BLIND" SPEECH SEGMENTATION

A "Blind" speech segmentation procedure allows a speech sample to be segmented into sub-word units without the knowledge of any linguistic information (such as, orthographic or phonetic transcription). Hence, this procedure involves finding the *optimal* number of sub-word segments in the given speech sample, before optimally locating the sub-word segment boundaries.

Some applications of the "Blind" speech segmentation include:
(a) *Speaker Verification systems:* To achieve a sub-word level segmentation (without orthographic information) of a user-selectable password in a text-dependent Speaker Verification system.
(b) *Speech Recognition systems:* To obtain sub-word level segmentation (for sub-word level modeling) in a low-to-medium size vocabulary speech recognition systems, with user-defined vocabulary (such as, in Voice Dialing application).
(c) *Language Identification systems:* To obtain sub-word level segmentation of untranscribed, multi-language speech corpora for Automatic Language Identification.
(d) *Speech corpus segmentation & labeling:* To obtain a *"coarse"*, sub-word level segmentation on newly acquired speech corpora. This can be used as seed values to aid the subsequent manual process of phonetic transcription.

The pseudo code for a generic *blind* speech segmentation algorithm can be given as follows:

(1) Estimate the range of the number of clusters (K_{min},K_{max})
(2) **for** K = K_{min} to K_{max}
(3) compute a K-cluster optimality criterion Q_K
(4) **end-for**
(5) Find the optimal number of clusters K_0
(6) $K_0 = \arg\max_{K=K_{min}}^{K=K_{max}} Q_K$
(7) Do segmentation for K_0 clusters

More specifically, the proposed *blind* segmentation algorithm can be outlined as follows:

(1) Estimate K_{min} using the number of syllables *(Convex-hull method)*
(2) Estimate K_{max} using a Spectral Variation Function (SVF)
(3) **for** K = K_{min} to K_{max}
(4) do *Level Building Dynamic Programming (LBDP)*-based segmentation for K-segments
(5) compute K-cluster optimality criterion Q_K using *Normal Decomposition method*
(6) **end-for**
(7) Optimal number of subwords K_0 is found as:
(8) $K_0 = \arg\max_{K=K_{min}}^{K=K_{max}} Q_K$
(9) Do DP-based segmentation for K_0 subwords.

The main steps in the above pseudo code are explained in detail below.

2.1. Estimating the range of the number of segments [K_{min}, K_{max}]

Estimating a reasonable limit on the expected number of segments in the given speech sample may be vital for faster and successful execution of the algorithm.

Estimating the minimum number of segments K_{min}

In the proposed algorithm, the lower limit on the number of segments is found by estimating the number of syllables in the given speech sample. It is assumed that the desired segmentation is at the acoustic-phonetic level. Since the linguistic unit of a syllable encompasses a phoneme unit, the number of syllables in a given speech sample can logically serve as the lower limit on its number of segments. The number of syllables in speech can be found by using the **Convex Hull** method [13]. The algorithm works on a subjective loudness function (such as, temporally smoothed log-energy of speech frames) to find the number of syllabic units.

Estimating the maximum number of segments K_{max}

The maximum number of segments in the given speech sample is estimated by using a spectral variation function (SVF). The proposed SVF is based on the Euclidean norm of the delta cepstral coefficients. The cepstral coefficients represent the log of the time-varying, short-time spectrum of speech. The time derivative of these cepstral coefficients will, therefore, represent the variation of speech spectrum with time. The frame-to-frame spectral variation function (SVF) can be approximated by a parameter generated by computing the Euclidean norm of delta cepstral coefficients:
$SVF_{\Delta cep}(n) = [\sum_{m=1}^{p} [\Delta c_n(m)]^2]^{\frac{1}{2}}$
where, p is the order of cepstral coefficient vector, and $\Delta c_n(m)$ is the m^{th} delta cepstral coefficient at time index n, obtained by fitting an orthogonal polynomial to the cepstral coefficient trajectory.

The function $SVF_{\Delta cep}(n)$ generally exhibits peak at boundaries between speech sounds (phonemes), where the spectral characteristics of speech change rapidly. The speech sound boundaries can then be located by picking local peaks in the trajectory of $SVF_{\Delta cep}(n)$ over time. Depending on the criterion used to define a local peak in the $SVF_{\Delta cep}$ trajectory, the algorithm may locate small and spurious peaks. This will give rise to more segment boundaries than are actually present in the speech sample. This is, however, utilized to set a limit on the maximum number of segments K_{max}.

The number of syllabic units and the number of peaks in the spectral variation function may be modified by some empirical rules before being actually used as the values for K_{min} and K_{max} respectively.

2.2. Level Building Dynamic Programming (LBDP)-based speech segmentation

A Level Building Dynamic Programming-based speech segmentation is a dynamic programming (DP)-based algorithm to optimally locate the sub-word boundaries by minimizing distortion metric. For every K in the range (K_{min}, K_{max}), a LBDP-based speech segmentation is used to segment the given speech sample into K sub-word speech segments.

A pattern matching approach to connected word recognition using Level Building Dynamic Programming (LBDP) was proposed in [14]. Given a fluently spoken word string and the reference templates of the vocabulary words, the connected word recognition problem requires to find the optimum match by concatenating these reference patterns. In the connected word recognition, it is desired to locate the word boundaries within the spoken string and to find the best sequence of spoken words. The speech segmentation problem at hand also requires to locate the optimal segment boundary points, so as to minimize the overall intra (within)-segment distortion. Therefore, the speech segmentation problem for a given number of subwords K, can also be formulated in a fashion similar to connected word recognition problem [10]. There are, however, some basic differences between these two problems. Firstly, there are no stored reference templates for the speech segments in speech segmentation. These reference templates are built *dynamically* during the search (for example, by averaging the vectors in the segment). Secondly, in speech segmentation it is not desired to find the optimal sequence of segments. The sequence of segments is already known (segment-1, segment-2, $\cdots$, segment-L). It is only desired to find the optimal boundary points between these segments. Due to these differences in the problem formulation, the original LBDP algorithm is modified here to suit the current needs.

The notation which shall be used in the following discussion of the algorithm is given below:

L total number of segments (equivalent to number of levels in level building algorithm)

l	segment (level) counter
N	total number of spectral frames in given speech sample
n,i	spectral frame counters
$\mathbf{x}_i$	speech spectral vector
$d_l(i,n)$	local distance at l^{th} level when frames i through n are classified in segment l
$D_l(n)$	accumulated distance at the l^{th} level and n^{th} frame; corresponds to the minimum accumulated distance for l segments in n frames
$B_l(n)$	backtrack pointer at the l^{th} level and n^{th} frame; corresponds to the best path for locating the optimal l segment boundaries ending at frame n.

If the local distance within a segment is computed using distance from the cluster mean, then
$d_l(i,n) = \sum_{j=i}^{j=n} \parallel \mathbf{x}_j - \mathbf{M}_{i,n} \parallel^2 \qquad (i < n)$
where,
$\parallel \mathbf{x} \parallel$ stands for the Eucledian norm of vector $\mathbf{x}$, and
$\mathbf{M}_{i,n}$ is the mean vector of frames i through n, given by $\mathbf{M}_{i,n} = \frac{1}{n-i+1} \sum_{j=i}^{j=n} \mathbf{x}_j$.

The Level Building Dynamic Programming-based speech segmentation is presented in the form of a pseudo-code as follows:

```
Level l = 1
        D_l(n) = d_l(1,n)              n = 1 to (N - L + 1)
        B_l(n) = 1

Levels l = 2 to L - 1
        D_l(n) = min_i{d_l(i+1,n) + D_{l-1}(i)}
             (i < n)      n = l to (N - L + l)
        B_l(n) = arg min_i{d_l(i+1,n) + D_{l-1}(i)}

Level l = L
        D_l(N) = min_i{d_l(i+1,N) + D_{l-1}(i)}
                 (i < N)
        B_l(n) = arg min_i{d_l(i+1,N) + D_{l-1}(i)}

Backtrack:
        Find the best path using the
        backtracking pointers
```

2.3. Normal Decomposition method

"Blind" segmentation also involves estimating the number of segments $K = K_0$, which will be optimal in some sense, for the given set of observation vectors $\{\mathbf{x_j}\}$. This is achieved by using the *Normal Decomposition* method ([15], Chapter 11). It is assumed that the complex distribution of all the input observation vectors can be approximated by the summation of several normal-like distributions. Consider that the distribution $p(\mathbf{x})$ of given input vectors consists of K normal distributions

$$p(\mathbf{x}) = \sum_{i=1}^{i=K} P_i p_i(\mathbf{x}) \tag{1}$$

where,
P_i is the prior probability, and,
$p_i(\mathbf{x}) = \mathcal{N}(\mathbf{x}, \mathbf{M}_i, \Sigma_i)$ is normal with expected vector $\mathbf{M}_i$ and covariance matrix Σ_i.
The *maximum likelihood* (ML) estimates of the parameters may be obtained by maximizing $\prod_{j=1}^{N} p(\mathbf{x}_j)$ with respect to P_i, $\mathbf{M}_i$ and Σ_i, under the constraint $\sum_{i=1}^{K} P_i = 1$ ([15]). The ML estimation formulas obtained are listed below:

$$P_i = \frac{1}{N} \sum_{j=1}^{N} q_i(\mathbf{x}_j) \quad (1 \le i \le K) \tag{2}$$

$$\mathbf{M}_i = \frac{1}{N_i} \sum_{j=1}^{N} q_i(\mathbf{x}_j)\mathbf{x}_j \tag{3}$$

$$\Sigma_i = \frac{1}{N_i} \sum_{j=1}^{N} q_i(\mathbf{x}_j)(\mathbf{x}_j - \mathbf{M}_i)(\mathbf{x}_j - \mathbf{M}_i)^T \tag{4}$$

where, $q_i(\mathbf{x}) = \frac{P_i p_i(\mathbf{x})}{p(\mathbf{x})}$ is the a posteriori probability of i^{th} normal mixture.

Using the segmentation boundaries obtained from the LBDP-based method (section 2.2), the ML estimates of P_i, $\mathbf{M_i}$ and Σ_i are computed using equations (2), (3) and (4) respectively. To compute the ML estimates using these equations, some initial values of the parameters P_i, $\mathbf{M_i}$ and Σ_i is necessary. The speech data vectors belonging to the i^{th}-segment are used to estimate the initial parameters of the i^{th} normal distribution.

A log likelihood criterion defined by:

$$Q_K = \sum_{j=1}^{j=N} \ln p(\mathbf{x}_j) \tag{5}$$

is used to determine the optimal number of clusters (segments). The maximized criterion value Q_K is obtained for a given number of clusters K.

This procedure is repeated for various values of K. The criterion Q_K tends to increase with increasing K, and reach a flat plateau at some optimal number K_0 (or even decrease beyond K_0 due to estimation errors). This means that even if we use more normal distributions (equivalently, more clusters or segments), the mixture distribution cannot be better approximated. Therefore, K_0 is used as the optimal number of segments for the given speech vectors. The plots of the optimality criterion Q_K with varying number of segments is shown in figure 1. The figure shows an ideal plot as well as two of the actual plots obtained for illustration.

Once the optimal number of segments $K = K_0$ in the given speech sample is known, the segmentation provided by the dynamic programming-based method is used for the location of the optimal segment boundaries.

3. EXPERIMENTAL RESULTS

3.1. "Blind" speech segmentation : an Illustration

In order to illustrate how the above *Blind* speech segmentation algorithm works, an example is presented here. Three repetitions of the word *"Manish"* were digitally recorded and run through the proposed *Blind* speech segmentation algorithm. The algorithm found the optimal number of segments to be six(6). The boundary location of the segments

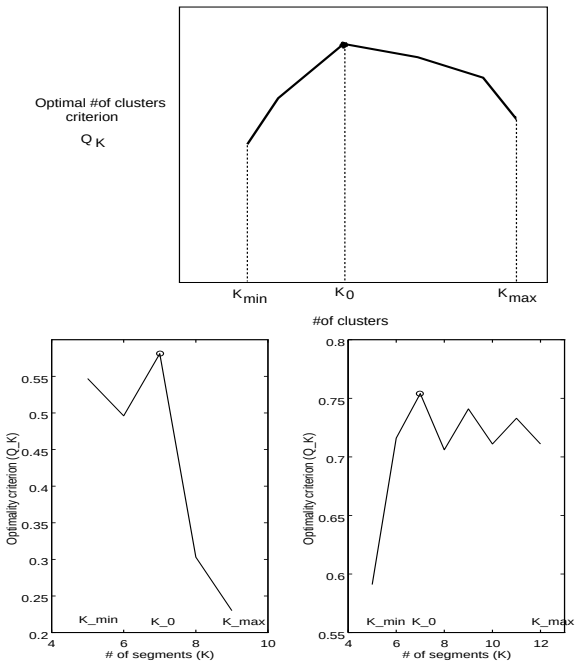

Figure 1: Plots of the optimality criterion Q_K for varying number of segments. (a) (Upper) Ideal curve (b) (Lower) Actual curves

as shown in figure 2, seem to correspond well to the actual phoneme boundary locations.

3.2. "Blind" speech segmentation : an Application

The proposed *Blind* speech segmentation procedure was incorporated in a subword-based text dependent speaker verification system, with user-selectable passwords [5, 16]. The spoken password of the speakers was segmented into subword units, without the knowledge of its text, using the *blind* speech segmentation procedure. The system with the *blind* speech segmentation procedure performed favorably to the system where the speech segmentation was based on the number of subword units obtained from the *"phonetic spelling"* of the password [5].

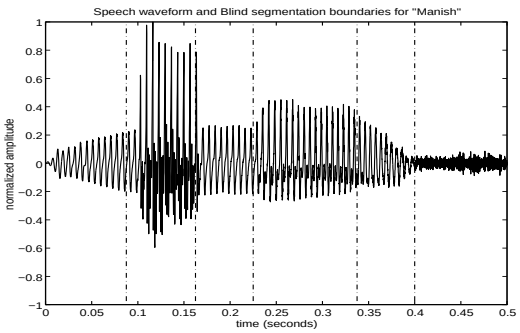

Figure 2: Sample output of the proposed Blind speech segmentation algorithm using speech file of the text *"Manish"*

	Segmentation using transcription	"Blind" segmentation
Equal Error Rate (EER)	1.37 %	0.93 %

4. CONCLUSION

A novel "Blind" speech segmentation procedure to automatically segment speech without the knowledge of any linguistic information was presented. This procedure finds the optimal number of subword segments in a given speech sample, and also locates the boundaries for these subword segments.

5. REFERENCES

[1] K.F. Lee. *Automatic Speech Recognition - The Development of the SPHINX system.* Kluwer Academic, Boston, 1989.

[2] B-H Juang and L.R. Rabiner. *Fundamentals of Speech Recognition.* Prentice-Hall, NJ, 1992.

[3] A.E. Rosenberg, C-H Lee, and F.K. Soong. Sub-word unit Talker Verification using Hidden Markov models. In *Proceedings of ICASSP*, pages 269–272, 1990.

[4] T. Matsui and S. Furui. Concatenated phoneme models for text-variable Speaker Recognition. In *Proceedings of ICASSP*, pages II 391–394, 1994.

[5] M. Sharma and R. Mammone. Subword-based text dependent speaker verification system with user-selectable passwords. In *Proceedings of ICASSP*, pages 93–96, 1996.

[6] Y.K. Muthusamy, E. Barnard, and R.A. Cole. Reviewing Automatic Language Identification. *IEEE Signal Processing magazine*, 11(4):33–41, October 1994.

[7] James R. Glass and Victor W. Zue. Multi-level acoustic segmentation of continuous speech. In *Proceedings of ICASSP*, pages 429–432, 1988.

[8] Ronald Cole and Lily Hou. Segmentation and broad classification of continuous speech. In *Proceedings of ICASSP*, pages 453–456, 1988.

[9] Kaichiro Hatazaki, Yasuhiro Komori, Takeshi Kawabata, and Kiyohiro Shikano. Phoneme segmentation using spectogram reading knowledge. In *Proceedings of ICASSP*, pages 393–396, 1989.

[10] T. Svendsen and F. Soong. On the automatic segmentation of speech signals. In *Proceedings of ICASSP*, pages 3.4.1–3.4.4, 1987.

[11] B-H Juang and L.R. Rabiner. The Segmental K-means algorithm for estimating parameters of Hidden Markov models. *IEEE Trans. on Acoustics, Speech and Signal proc.*, 38(9):1639–1641, September 1990.

[12] A. Ljolje and M.D. Riley. Automatic segmentation and labeling of speech. In *Proceedings of ICASSP*, pages 473–476, 1991.

[13] Paul Mermelstein. Automatic segmentation of speech into syllabic units. *Journal of Acoustical Society of America*, 58(4):880–883, October 1975.

[14] C.S. Myers and L.R. Rabiner. A Level Building Dynamic Time Warping algorithm for connected word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29:284–297, April 1981.

[15] K. Fukunaga. *Introduction of Statistical Pattern Recognition.* Academic Press, San Diego, 1990.

[16] M. Sharma. *Subword-based Text Dependent Speaker Verification System with User-selectable passwords.* PhD thesis, Rutgers University, May 1996.