Query-by-Example Spoken Term Detection using Frequency Domain Linear Prediction and Non-Segmental Dynamic Time Warping

Gautam Mantena, Sivanand Achanta, and Kishore Prahallad

***Abstract*—The task of query-by-example spoken term detection (QbE-STD) is to find a spoken query within spoken audio data. Current state-of-the-art techniques assume zero prior knowledge about the language of the audio data, and thus explore dynamic time warping (DTW) based techniques for the QbE-STD task. In this paper, we use a variant of DTW based algorithm referred to as non-segmental DTW (NS-DTW), with a computational upper bound of $O\left(mn\right)$ and analyze the performance of QbE-STD with Gaussian posteriorgrams obtained from spectral and temporal features of the speech signal. The results show that frequency domain linear prediction cepstral coefficients, which capture the temporal dynamics of the speech signal, can be used as an alternative to traditional spectral parameters such as linear prediction cepstral coefficients, perceptual linear prediction cepstral coefficients and Mel-frequency cepstral coefficients. We also introduce another variant of NS-DTW called fast NS-DTW (FNS-DTW) which uses reduced feature vectors for search. With a reduction factor of $\alpha \in \mathbb{N}$, we show that the computational upper bound for FNS-DTW is $O(\frac{mn}{\alpha^2})$ which is faster than NS-DTW.**

***Index Terms*—Dynamic time warping, fast search, frequency domain linear prediction, query-by-example spoken term detection.**

I. Introduction

THE task of query-by-example spoken term detection (QbE-STD) is to find a spoken query within spoken audio. A key aspect of QbE-STD is to enable searching in multi-lingual and multi-speaker audio data. A traditional QbE-STD approach is to convert spoken audio into a sequence of symbols and then perform text based search. In [1]–[3], the audio is first converted to a sequence of symbols using automatic speech recognition (ASR) and then lattice based search techniques are incorporated.

ASR based techniques assume the availability of labelled data for training the acoustic and language models. Such approaches are not scalable for languages where there is no availability or the resources to build an ASR. To overcome this limitation, zero prior knowledge is assumed about the language of the spoken audio, and thus dynamic time warping (DTW) based techniques are exploited for QbE-STD [4]–[9]. One of the popular DTW based techniques is the segmental DTW (S-DTW) [4], which uses a windowed (or segmental) type of approach to search a spoken query within spoken audio. In this paper, we use a variant of DTW referred to as non-segmental DTW (NS-DTW) which has been applied for segmentation of large speech files [10], [11] and also for QbE-STD tasks [6], [8], [9]. In [12], the NS-DTW is referred to as subsequence DTW.

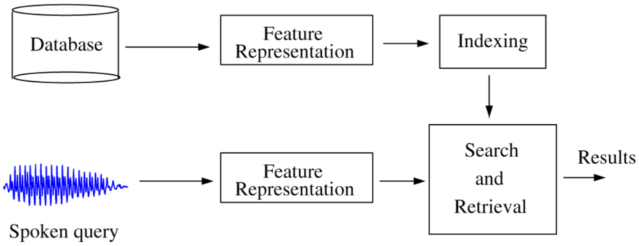


Fig. 1. A general architecture for a QbE-STD system.

Fig. 1 shows a general architecture of a QbE-STD system. Speech features are extracted from the audio database and are indexed for quick retrieval during the search process. In [4], [8], [13], Gaussian posteriorgrams are shown to be a good feature representation to suppress speaker characteristics and to perform search across multi-lingual data.

In general, Gaussian posteriorgrams used for QbE-STD are computed from short-time spectral parameters such as Mel-frequency cepstral coefficients. In [14], [15], it is shown that frequency domain linear prediction cepstral coefficients (FDLP) perform better than the short-time spectral parameters for speech recognition in noisy environments. In FDLP, the temporal dynamics of the speech signal are captured by applying an all-pole model in the spectral domain. Athineos *et al.* [16] provides a detailed mathematical analysis of extracting the temporal envelope of the signal using autoregressive modelling. In this paper, we show that Gaussian posteriorgrams computed from FDLP, which capture the temporal dynamics of the speech signal, can be used as an alternative to traditional spectral parameters such as linear prediction cepstral coefficients (LPCC), perceptual linear prediction cepstral coefficients (PLP) and Mel-frequency cepstral coefficients (MFCC).

In [7], [17], indexing based approaches such as locality sensitive hashing and hierarchical clustering are used to build sparse

Manuscript received March 24, 2013; revised September 23, 2013; accepted February 25, 2014. Date of publication March 11, 2014; date of current version April 04, 2014. The work of G. Mantena was supported in part by Tata Consultancy Services (TCS) through a Ph.D. fellowship at IIIT-Hyderabad, India. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. James Glass.

The authors are with International Institute of Information Technology (IIIT-H), Hyderabad 500032, India (e-mail: gautam.mantena@gmail.com; siva.kits@gmail.com; kishore@iiit.ac.in).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2014.2311322

TABLE I
STATISTICS OF MEDIAEVAL 2012 DATA

Data	Utts	Total(min)	Average(sec)
dev reference	1580	221.9	8.42
dev query	100	2.4	1.44
eval reference	1660	232.5	8.40
eval query	100	2.5	1.50

similarity matrices for searching the spoken query. Use of indexing techniques is not in the scope of this work. Thus a spoken utterance is represented by a sequence of Gaussian posteriorgrams and a full similarity matrix is used for searching a spoken query.

The brief set of contributions of our work is as follows:

- We provide a comparison of time complexity of NS-DTW and S-DTW [4]. We experiment with different local constraints in NS-DTW based method, and report results on the common MediaEval 2012 dataset [18].
- In this work, we introduce a faster method of searching a spoken query. This method exploits the redundancy in speech signal, and averages the successive Gaussian posteriorgrams to reduce the length of the spoken audio and the spoken query. However with such an approach there is a trade-off between search performance and accuracy and these results are reported. We show that the search time of the proposed fast NS-DTW is lower than that of the randomized acoustic indexing method described in [19].
- We provide experimental results to show that the Gaussian posteriorgrams obtained from FDLP can be used for QbE-STD as an alternative to other short-time spectral parameters such as MFCC.

II. DATABASE

The experiments conducted in this work use MediaEval 2012 data which is a subset of Lwazi database [18]. The data consists of audio recorded via telephone in 4 of 11 South African languages. We have considered two data sets, development (dev) and evaluation (eval) which contain spoken audio (reference) and spoken query data. The statistics of the audio data is shown in Table I.

III. FEATURE REPRESENTATION FOR SPEECH

Feature representation of the speech signal was obtained by a two step process. In the first step, parameters were extracted from the speech signal. In the second step, Gaussian posteriorgrams were computed from these parameters. The different parameters extracted from the speech signal were as follows: (a) Linear prediction cepstral coefficients (LPCC), (b) Mel-frequency cepstral coefficients (MFCC), (c) Perceptual linear prediction cepstral coefficients (PLP) [20] and (d) Frequency domain linear prediction cepstral coefficients (FDLP).

$$X[k] = a[k] \sum_{n=0}^{N-1} x[n] \cos\left(\frac{(2n+1)\pi k}{2N}\right)$$
$$k = 0, 1, \ldots, N-1$$

where:

$$a[k] = \begin{cases} \frac{1}{\sqrt{N}} & k = 0 \\ \sqrt{\frac{2}{N}} & k = 1, 2, \ldots, N-1 \end{cases} \tag{1}$$

In linear prediction (LP) analysis of speech an all-pole model was used to approximate the vocal tract spectral envelope [21]. MFCC, PLP and FDLP use a series of band pass filters to capture speech specific characteristics. To compute MFCC, signal was passed through a bank of filters to compute the energy of the signal in each of the bands. This energy from each band is referred to as Mel-spectrum. Cepstral coefficients were then computed by performing DCT on these sub-band energies. In PLP analysis of speech, the power spectrum was modified before applying the LP all-pole model. The modified spectrum was obtained as follows [20]: (a) speech signal is first passed through the filter banks, (b) pre-emphasis by an equal loudness curve on the filtered signal and (c) cubic compression of the spectrum.

In LPCC, MFCC and PLP the short-time spectral properties of the speech signal are captured. In order to capture the temporal dynamics of the speech signal, frequency domain linear prediction (FDLP) was developed [14]–[16]. FDLP technique relies on all-pole modeling in the spectral domain to characterize the temporal dynamics of the frequency components. In [15], the performance of FDLP parameters for phoneme recognition was evaluated in noise conditions such as additive noise, convolutive noise and telephone channel. It was shown that, in such noise conditions, FDLP was performing better as compared to other parameters such as PLP. This motivated us to explore FDLP based features for QbE-STD.

Following the work in [22], FDLP parameters were computed as follows–(a) DCT was computed over the entire signal using Eq. (1), (b) Filter bank analysis was performed on the DCT output. (c) An all-pole model was applied on the spectral components in each sub-band, (d) For each sub-band, time domain envelope was computed by taking the frequency response of the all-pole model, (e) Short-time analysis was performed on the envelopes from each of the sub-bands to compute the FDLP spectrum, and (f) DCT was then applied on the FDLP spectrum to obtain cepstral coefficients.

A. Representation using Gaussian Posteriorgrams

Gaussian posteriorgrams were computed from the 39 dimensional LPCC, MFCC, PLP and FDLP parameters. A 25 ms window length with 10 ms shift was considered to extract 13 dimensional parameters along with delta and acceleration coefficients for all the parameters. An all-pole model of order 12 was used for LPCC, PLP and an order of 160 poles/sec for the FDLP parameters. A set of 26 filter banks were used for computing MFCC, PLP and 37 filter banks for the FDLP parameters.

Gaussian posteriorgrams were computed from these parameters as described in [8]:

1) K-means was used to initialize the means of the Gaussian mixture models (GMM). The initialization started by computing the mean μ and standard deviation σ from the entire data. Then a split operation was performed and the new

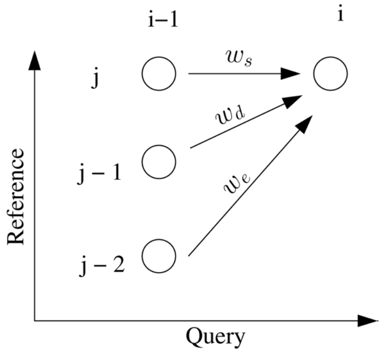


Fig. 2. A pictorial representation of the local constraints along with the weights w_s, w_d and w_e associated with each of the arcs.

centers were given by $\mu \pm 0.2\sigma$. The process of clustering and splitting continued till the required number of means were reached.

2) GMMs were trained with its centers initialized by K-means.
3) As a final step, feature vectors were pooled for each Gaussian having a maximum likelihood and the means and covariances were recomputed.

IV. Search using Non-segmental DTW

Dynamic time warping (DTW) algorithm performs a non-linear alignment of two time series. During this process, the warping constraints such as (1) start and end point, (2) monotonicity, (3) local, (4) global and (5) slope weighting are considered [23].

In segmental DTW (S-DTW), we use global constraints to restrict the alignment within a certain segment of the spoken audio. Segmenting the spoken audio using global constraints and then performing DTW is computationally expensive. As an alternative, we use non-segmental DTW (NS-DTW), where we approximate the start and end point constraints.

Let $\mathcal{Q}$ be a spoken query (or query) containing n feature vectors. Let $\mathcal{R}$ be the spoken audio (or reference) containing m feature vectors. The sequence of feature vectors are denoted as follows:

$$\mathcal{Q} = \{\mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q_i}, \ldots, \mathbf{q_n}\},$$
$$\mathcal{R} = \{\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_j}, \ldots, \mathbf{u_m}\}.$$

Each of these feature vectors represent a Gaussian posteriorgram as computed in Section III-A. The distance measure between a query vector $\mathbf{q_i}$ and a reference vector $\mathbf{u_j}$ is given by Eq. (2).

$$d(i,j) = -\log\left(\frac{\mathbf{q_i}}{\|\mathbf{q_i}\|} \cdot \frac{\mathbf{u_j}}{\|\mathbf{u_j}\|}\right) \tag{2}$$

We define the term *search hit* as the region in the reference $\mathcal{R}$ that is likely to contain the query $\mathcal{Q}$. In NS-DTW, we use only the local constraints as shown in Fig. 2 to obtain the *search hits*. The choice of these local constraints is motivated by their use in isolated word recognition [24] and in large vocabulary speech recognition [25], [26]. These local constraints are often referred as Bakis topology [25]. In Section V-D, we compare the performance of different sets of local constraints for QbE-STD tasks.

We compute a similarity matrix S of size $m \times n$, where m, n are the number of feature vectors of the reference and the query. Let i, j represent a column and a row index of a matrix. The query can start from any point in the reference. Initially, $S(1,j) = d(1,j)$, where $d(1,j)$ is the distance measure given by Eq. (2). The entries in the rest of the similarity matrix is given by Eq. (3) [8].

$$S(i,j) = \min\left\{\begin{array}{c} \frac{d(i,j)+S(i-1,j-2)}{T(i-1,j-2)+w_e} \\ \frac{d(i,j)+S(i-1,j-1)}{T(i-1,j-1)+w_d} \\ \frac{d(i,j)+S(i-1,j)}{T(i-1,j)+w_s} \end{array}\right\}, \tag{3}$$

where T is called the transition matrix. $T(i,j)$ represents the number of transitions required to reach i,j from a start point, and normalizes the accumulated score with the length of the aligned path. The update equation for the transition matrix T is given by Eq. (4).

$$T(i,j) = \begin{cases} T(i-1,\hat{j}) + w_e & \text{if } \hat{j} = j-2 \\ T(i-1,\hat{j}) + w_d & \text{if } \hat{j} = j-1 \\ T(i-1,\hat{j}) + w_s & \text{if } \hat{j} = j \end{cases} \tag{4}$$

where

$$\hat{j} = \underbrace{\arg\min}_{\hat{j}\in\{j,j-1,j-2\}} \left\{\begin{array}{c} \frac{d(i,j)+S(i-1,j-2)}{T(i-1,j-2)+w_e} \\ \frac{d(i,j)+S(i-1,j-1)}{T(i-1,j-1)+w_d} \\ \frac{d(i,j)+S(i-1,j)}{T(i-1,j)+w_s} \end{array}\right\}.$$

In Eq. (4), w_e, w_d, w_s are the weights associated for each transition. In Section V-A, we show the effect of weights on the search performance of NS-DTW and thereby select the optimum values for the weights.

A. Selection of Start and End Time Stamps

We use a matrix P to record the path transitions. The values in the matrix $P(i,j)$ are updated when the similarity matrix is being computed and are given by Eq. (5) and Eq. (6).

$$P(i,1) = i \text{ for } i = 1,2,3,\ldots,m \tag{5}$$
$$P(i,j) = P(i-1,\hat{j}) \tag{6}$$

In order to detect the start and end time stamps of the *search hit*, we obtain the reference index that contains the best alignment score, i.e., the end point of the *search hit* as given by $\min_j\{S(n,j)\}$ for $j = 1,2,...,m$. Once this end point is obtained, the corresponding start point could be obtained by $P(n,j)$and thus avoiding the need for a path traceback to obtain the start time stamp of the search hit.

Fig. 3(a) shows an example similarity matrix plot of a query and a reference where the dark bands represent the segments that are similar between the query and the reference. To visualize the similarity matrix, each value in the matrix is scaled using an exponential function and then each column is normalized by the maximum value of the column. Please note that a full similarity matrix is computed and the white regions (as shown in Fig. 3) does not imply that we do not compute the values of the matrix in those regions.

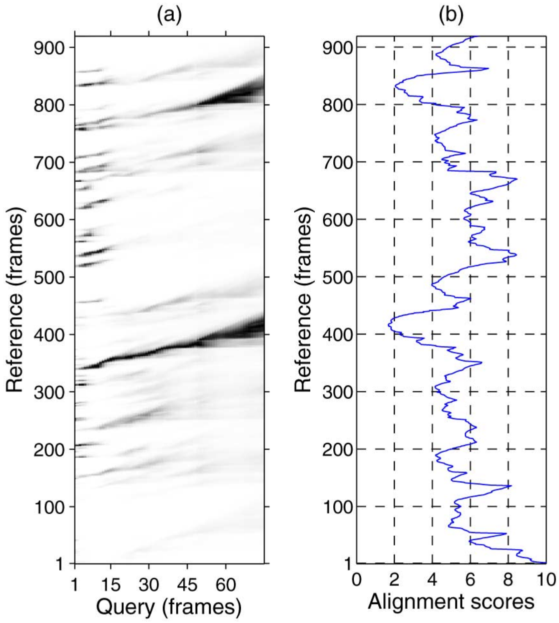


Fig. 3. (a) An example similarity matrix plot obtained using NS-DTW when a query is present in the reference and (b) A plot of the alignment scores obtained from the last column of the similarity matrix. Please note that, to visualize the similarity matrix, the values in the matrix are scaled using an exponential function and then each column is normalized with the maximum value of the column.

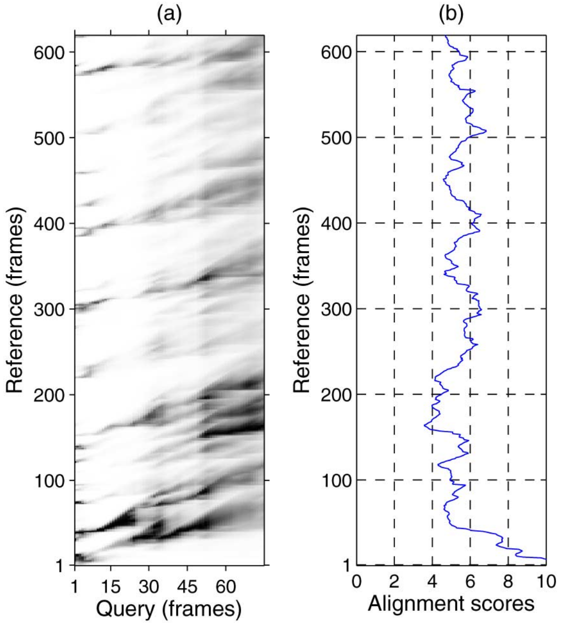


Fig. 4. (a) An example similarity matrix plot obtained using NS-DTW when a query is not present in the reference and (b) A plot of the alignment scores obtained from the last column of the similarity matrix. Please note that, to visualize the similarity matrix, the values in the matrix are scaled using an exponential function and then each column is normalized with the maximum value of the column.

The dark bands that have reached the end of the query are the required *search hits*. They can be obtained from the alignment scores from the last column of the similarity matrix S. Fig. 3(b) shows the alignment scores where the minimum values represent the end of the *search hits* and from these points the start time stamps are obtained using Eq. (5) and Eq. (6).

As shown in Fig. 3(a), the query could have more than one match in the reference and hence k-best alignment scoring indices are selected from the similarity matrix. In Section V-B, we show the effect of the choice of k-best alignment scores on the search performance of NS-DTW and thereby select the optimum k value.

Fig. 4(a) shows an example similarity matrix plot when a query is not present in the reference. The partial bands that are observed in Fig. 4(a) show a partial match between the query and the reference. From Fig. 4(b), it can be seen that the alignment scores of the search hits are higher than that of the scores of a search hit shown in Fig. 3(b).

B. Analytical Comparison with Segmental-DTW

Segmental DTW (S-DTW) [4] is a popular technique that overcomes the start and end point constraints by dividing the spoken audio into a series of segments, and then DTW is performed on each segment. S-DTW is computationally not efficient due to this segment based DTW approach that it performs to obtain the *search hits*.

Two constraints are imposed on the alignment. The first one is a parameter r which dictates the length of the segment to be taken from the reference. This is given by the inequality $|i - j| \leq r$ (Sakoe-Chiba band [23]), where i, j are the frame indices of the query and the reference. This constraint prevents the warping from going too far ahead or behind.

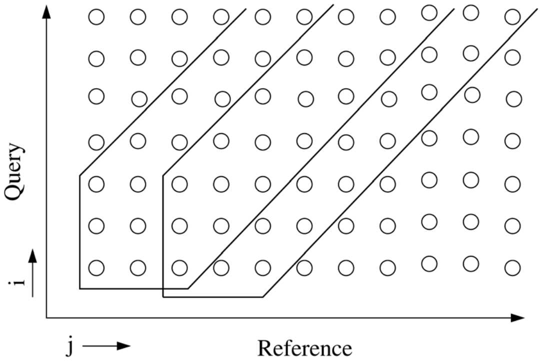


Fig. 5. An example of segmental DTW (S-DTW) with the first two segments for $r = 2$.

The second constraint is the number of such segments to be considered. Fig. 5 shows the first two segments of S-DTW for $r = 2$. Normally one would shift the segment by one frame as the query could start from any point in the reference, but due to the huge computational overload a shift of r is considered.

The total number of computations required is equal to $number\ of\ computations\ in\ each\ segment \times number\ of\ segments$. Given a query $\mathcal{Q}$ of size n, the length of the segment taken from reference $\mathcal{R}$ is $n + r$ ($\because j \leq i + r$). Thus the number of computations required in each segment is of the order $O(n^2)$. For $r = 1$, searching in a reference of size m, we need to initiate m DTW searches each of order $O(n^2)$. The overall computation would be of the order $O(mn^2)$.

In NS-DTW, we are computing a similarity matrix of size $m \times n$ and so the upper bound of NS-DTW would be $O(mn)$. This is computationally faster than the S-DTW whose upper bound is $O(mn^2)$. The upper bound on the distance computation between two vectors is $O(d)$, where d is the dimensions of the vector. This distance computation is common across S-DTW and NS-DTW and so it is omitted for calculating the computational upper bound.

In NS-DTW, in order to avoid path traceback to obtain the start and end time stamps, we use a matrix P (as given by Eq. (6)). However, one can always use path traceback for obtaining the start time stamp. In such a case, the total time complexity of searching using NS-DTW is $O(mn)+O(n)$, where $O(n)$ is time complexity of path traceback. With $m \gg n, O(mn)+O(n)=O(mn)$. Thus, the time complexity of NS-DTW is $O(mn)$ irrespective of whether path traceback or a matrix P is used. It is to be noted that the use of a matrix P will result in a higher memory requirement for computation.

C. *Variants of NS-DTW*

In [6], [8], [9], variants of NS-DTW are used for QbE-STD. These variants differ in the type of local constraints, values of weights and frame-based normalization. In [6], frame-based normalization is used by dividing the values in the column by the maximum value of the column. In this work, we do not perform frame-based normalization. However, we normalize each value in the similarity matrix, $S(i,j)$, by a transition matrix value, $T(i,j)$ (as given by Eq. (3)). Further details of our implementations are described in Section V.

V. Evaluation and Results

All the evaluations are performed using 2006 NIST evaluation criteria [27] and the corresponding maximum term weighted values (MTWV) are reported. To compute the MTWV, the average miss probability (MP) and false alarm probabilities (FAP) are computed for all the queries. More details on the evaluation can be found in [28].

A. *Weights of Local Constraints*

As given by Eq. (3) and Eq. (5), we use weights for each of the local constraints to normalize the scores. During alignment, many deletions and insertions are an indication of the mismatch between the two sequence of feature vectors and hence more importance is given to the diagonal transition (w_d). Fig. 6 shows MTWV for various values of w_d (with $w_e = w_s = 1$). NS-DTW is evaluated using 128 dimensional Gaussian posteriorgrams computed from LPCC, PLP, MFCC and FDLP. From Fig. 6, it can be seen that (a) MFCC and FDLP based features have similar MTWV for $w_d = 3$ on the dev dataset, and (b) NS-DTW performs best for FDLP at $w_d = 2$. For all of the experiments reported in this work, $w_d = 2$ is considered based on the performance of Gaussian posteriorgrams of FDLP.

B. *Selection of Number of Search Hits*

In NS-DTW, after computing the similarity matrix, we select k-best alignment score indices. Using matrix P (as described in Section IV), we obtain the start time stamps of the search hits given k-best indices. After obtaining the k-best search hits, a post processing step is performed on the overlapping search hits. If there is an overlay of more than 50% between any two search hits, the search hit with the best alignment score is considered.

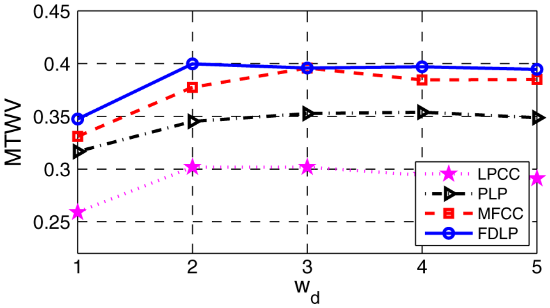


Fig. 6. Maximum term weighted value (MTWV) obtained using various values of w_d for dev dataset.

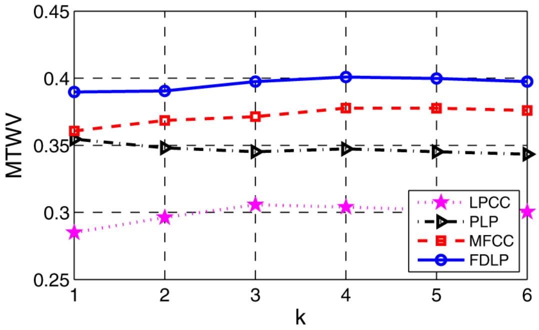


Fig. 7. MTWV obtained using various values of k for dev dataset.

In a reference, there might be a possibility of multiple occurrences of the query. In such a case, $k=1$ will result in an increase in miss probability. On the other hand a large value of k will increase in the number of false alarms. Thus, an appropriate value of k is needed. Fig. 7 shows the performance of NS-DTW for different values of k on dev dataset across different parameters. From Fig. 7, it can be seen that the MTWVs are similar for various values of k and thus the choice of $k=5$ is chosen.

C. *Number of Gaussians*

Table II shows the MTWV and the search speed (in minutes) obtained using LPCC, PLP, MFCC and FDLP parameters by varying the number of Gaussians for the dev dataset. In Table II, we show the rate of improvement in the MTWV (indicated within the brackets for each of the MTWV values) by increasing the number of Gaussians. For example, the rate of improvement in MTWV for FDLP by increasing the number of Gaussians from 64 to 128 is 0.050.

In Table II, we also show the search speed, i.e. the time required to search all the queries within the dataset. The distance computation, given by Eq. (2), between a query feature (q_i) and a reference feature (u_j) is $O(d)$, where d is the dimension of the feature. This distance computation is common across S-DTW and NS-DTW and so it is omitted for calculating the computational upper bound. However, the feature dimension has an impact on the search speed of NS-DTW and is shown in Table II. The search speed of NS-DTW using a d dimensional Gaussian posteriorgram will be similar irrespective of the parameters (such as MFCC, FDLP) used to build a GMM. Thus, we have reported the search speed of NS-DTW using Gaussian posteriorgrams of FDLP by varying the number of Gaussians (as shown in Table II).

TABLE II
MAXIMUM TERM WEIGHTED VALUE (MTWV) AND SEARCH SPEED[1] ON DEV DATASET BY VARYING THE NUMBER OF GAUSSIANS FOR EACH OF THE PARAMETERS. THE VALUES INDICATED IN THE BRACKETS SHOW THE RATE OF IMPROVEMENT IN THE MTWV ON INCREASING THE NUMBER OF GAUSSIANS.

No. of Gaussians	**MTWV**				**Search Speed (mins)**
	LPCC	**PLP**	**MFCC**	**FDLP**	
8	0.031 (-)	0.080 (-)	0.059 (-)	0.084 (-)	18.39
16	0.127 (0.096)	0.128 (0.048)	0.119 (0.06)	0.207 (0.123)	22.57
32	0.218 (0.091)	0.271 (0.143)	0.246 (0.127)	0.292 (0.085)	30.60
64	0.252 (0.034)	0.319 (0.048)	0.347 (0.101)	0.349 (0.057)	47.53
128	**0.301 (0.049)**	**0.345 (0.026)**	**0.377 (0.030)**	**0.399 (0.050)**	**80.24**
256	0.310 (0.009)	0.387 (0.042)	0.410 (0.033)	0.410 (0.011)	145.07
512	0.311 (0.001)	0.399 (0.012)	0.410 (0.000)	0.422 (0.012)	274.98
1024	0.319 (0.008)	0.404 (0.005)	0.413 (0.003)	0.432 (0.010)	534.15

[1] Please note that, for a given number of Gaussians, the search speed (in NS-DTW) will be similar for each of the parameters. Thus, the search speed is reported using Gaussian posteriorgrams of FDLP

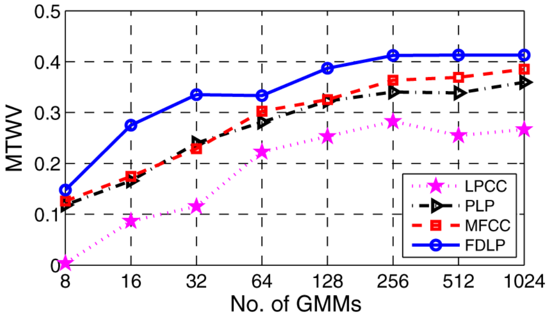


Fig. 8. Maximum term weighted values (MTWV) on eval dataset by varying the number of Gaussians for each of the parameters.

From the MTWV reported in Table II, it can be seen that (a) The performance of NS-DTW improves by increasing the number of Gaussians. However, the rate of improvement in performance for NS-DTW decreases when the number of Gaussians exceeds 128, (b) With the increase in number of Gaussians, MTWV of FDLP, MFCC and PLP seems to be converging, and (c) FDLP performs similar to that of MFCC for 256 Gaussians. From Table II, it can also be seen that there is a trade-off between the performance of NS-DTW and the search speed by increasing the number of Gaussians. Considering the MTWV and the search speed on dev dataset we have chosen 128 Gaussians as an optimum number for NS-DTW.

Although we have chosen 128 to be the optimum number of Gaussians, we would want to verify the effect of search performance on the eval dataset by varying the number of Gaussians. Fig. 8 shows the performance of NS-DTW using different number of GMMs trained with LPCC, MFCC, PLP and FDLP parameter streams using the eval dataset. In Fig. 8, we observe the following: (a) The curve flattens after 256 Gaussians for the features obtained from FDLP. Thus there is no further improvement in the search performance by increasing the number of Gaussians, (b) FDLP is performing better than the other acoustic parameters such as LPCC, PLP and MFCC. However, on increasing the number of Gaussians, the MTWVs of MFCC and PLP seems to be converging towards that of FDLP, and (c) Drop in the search performance for LPCC at 512 Gaussians which may be an indication of model over-fitting.

D. Effect of Different Local Constraints

In this section, we analyze the performance of DTW-based techniques with other local constraints as shown in Table III. In [6], local constraints T2 and in [8], [9], local constraints T3 are used for QbE-STD.

TABLE III
SOME OF THE TYPES OF LOCAL CONSTRAINTS USED IN DTW-BASED QBE-STD.

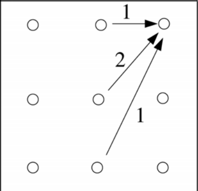
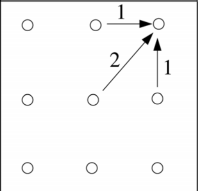
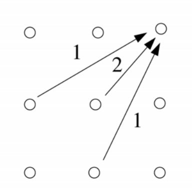
	Local Constraints
T1	
T2	
T3	

Fig. 9(a) and 9(b) show the MTWV obtained using 128 dimensional Gaussian posteriorgrams of LPCC, PLP, MFCC and FDLP parameters for dev and eval datasets using T1, T2 and T3 local constraints. T1 is the local constraints used in NS-DTW (also shown in Fig. 2).

From Fig. 9(a), T2 is performing better than the other local constraints on the dev dataset. In Fig. 9(b), it can be seen that T1 is performing similar to that of T2 on eval dataset. T2 allows insertions in a query which can be interpreted as a deletion operation on the reference and this might be the reason for T1 and T2 to perform similarly on eval dataset. However, the results are not consistent, i.e., T2 performs better than T1 on dev dataset (as shown in Fig. 9). One could argue that T2 allows insertions within a query and thus more suitable for QbE-STD. As described in Section IV, we are motivated to use T1 for NS-DTW by their use in large vocabulary speech recognition and feasibility in usage of embedded training for unsupervised acoustic models with left-to-right Bakis topology [29], [30].

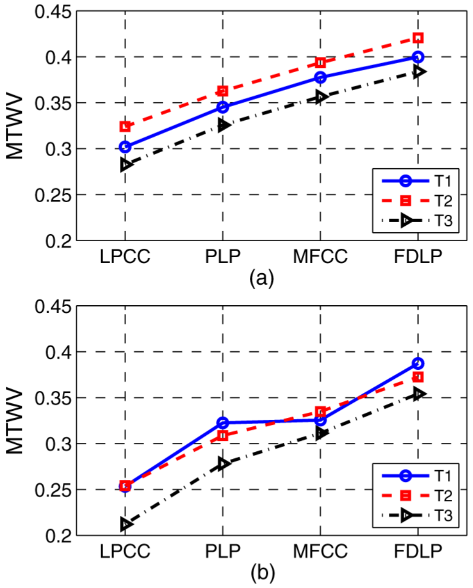


Fig. 9. MTWV obtained using 128 dimensional Gaussian posteriorgrams of various parameters using T1, T2 and T3 local constraints for (a) dev, and (b) eval datasets.

TABLE IV
MISS PROBABILITY (MP), FALSE ALARM PROBABILITY (FAP) AND MAXIMUM TERM WEIGHTED VALUE (MTWV) OBTAINED USING NS-DTW AND GAUSSIAN POSTERIORGRAMS OF LPCC, MFCC, PLP AND FDLP.

Feats.	**dev**			**eval**		
	MP	FAP (10^{-2})	MTWV	MP	FAP (10^{-2})	MTWV
LPCC	0.575	0.802	0.301	0.564	1.529	0.253
MFCC	0.492	0.848	0.377	0.572	0.860	0.325
PLP	0.504	0.982	0.345	0.505	1.441	0.322
FDLP	0.426	1.136	**0.399**	0.402	1.766	**0.387**

E. Use of FDLP for QbE-STD

Speech parameters such as LPCC, PLP and MFCC are obtained by windowing the speech signal and followed by estimating the spectrum from each window. However, speech signal has information spread across longer temporal context and this information can be captured by using FDLP parameters. In Table II, it can be seen that FDLP performs similar to that of MFCC using 256 Gaussians. Thus, we show that FDLP parameters, which capture the temporal characteristics of a speech signal, can be used as an alternative to other spectral parameters such as MFCC. In Fig. 8, it can be seen that FDLP performs better than MFCC for 128 and 256 GMMs and thus a motivation to use FDLP parameters for QbE-STD. To summarize the search performance of the various parameters, in Table IV we show detail results in terms of MP, FAP and MTWV using 128 dimensional Gaussian posteriorgrams.

VI. FAST NS-DTW

The computational analysis shown in Section IV indicates that NS-DTW is faster, than S-DTW, with an upper bound of $O(mn)$. Even with this computational improvement, DTW based techniques are still slow as compared to other model based techniques [1]–[3].

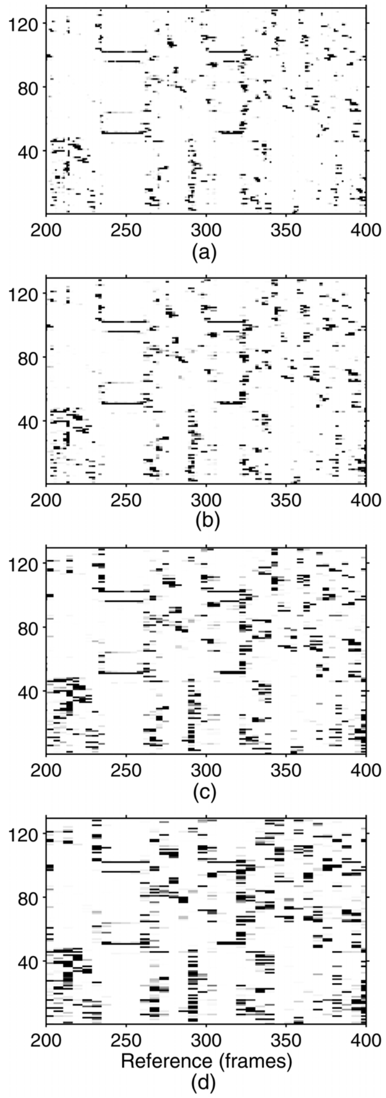


Fig. 10. Gaussian posteriorgrams of a reference segment for (a) $\alpha = 1$, (b) $\alpha = 2$, (c) $\alpha = 4$, (d) $\alpha = 6$. The y-axis represents the indices of the Gaussian components in GMM. Please note that the frames on the x-axis are repeated for α times to visualize the smoothed Gaussian posteriorgrams on the same scale. For visualization, we normalize each of the columns with the maximum value of the column.

Some of the standard techniques to improve the computational performance of DTW are [31]:

- *Constraints*: Use of constraints such as Sakoe-Chiba band [23] or Itakura parallelogram [24] to limit the number of computations in the similarity matrix.
- *Data Abstraction*: Use a reduced feature representation to perform DTW. To improve the computational performance of NS-DTW, we use reduced Gaussian posteriorgrams to perform the search.
- *Indexing*: Indexing based techniques retrieve the reference feature vectors used to construct a sparse similarity matrix, which makes the search efficient [7], [17]. Use of indexing techniques is not in the scope of this paper and we compute a full similarity matrix to perform the search.

In this section, we introduce a modification to NS-DTW by reducing the query and reference Gaussian posteriorgram vectors before performing search. We refer to this algorithm as

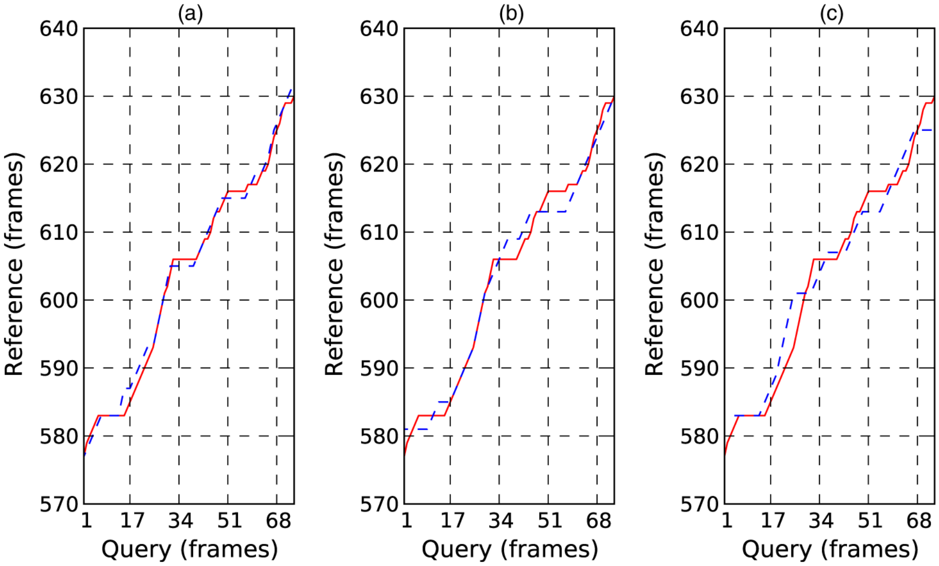


Fig. 11. Alignment paths for an example query and reference using NS-DTW and FNS-DTW using (a) $\alpha = 2$, (b) $\alpha = 4$, (c) $\alpha = 6$.

TABLE V

CORRECTIONS: MISS PROBABILITY (MP), FALSE ALARM PROBABILITY (FAP) AND MAXIMUM TERM WEIGHTED VALUE (MTWV) OBTAINED USING FNS-DTW FOR VARIOUS VALUES OF α.

α	dev			eval		
	MP	FAP (10^{-2})	MTWV	MP	FAP (10^{-2})	MTWV
1	0.426	1.136	0.399	0.402	1.766	0.387
2	0.423	1.159	0.399	0.468	1.302	0.376
3	0.482	0.940	0.374	0.536	1.079	0.334
4	0.494	0.994	0.353	0.528	1.251	0.322
5	0.555	0.791	0.323	0.543	1.307	0.301
6	0.503	1.236	0.307	0.576	1.279	0.271

fast NS-DTW (FNS-DTW). Given a reduction factor $\alpha \in \mathbb{N}$, a window of size α is considered over the posteriorgram features and a mean is computed. The window is then shifted by α and another mean vector is computed. The posteriorgram vectors are replaced with the reduced number of posteriorgram features during this process. With a reduction factor of α, the new size of the query and the reference would be $\frac{n}{\alpha}$ and $\frac{m}{\alpha}$ respectively. This would result in a computational upper bound of $O(\frac{mn}{\alpha^2})$ for FNS-DTW. This technique is independent of the local constraints used and we use T1 local constraints for FNS-DTW.

Fig. 10 shows the 128 dimensional Gaussian posteriorgrams of a reference segment for $\alpha = 1, 2, 4, 6$, where $\alpha = 1$ represents no reduction in the Gaussian posteriorgrams. In Fig. 10, the frames on the x-axis are repeated for α times to visualize the smoothed Gaussian posteriorgrams on the same scale. From Fig. 10 and Table V, it is evident that for smaller values of α, such as $\alpha = 2$, the Gaussian posteriorgrams are similar to that of $\alpha = 1$ resulting in a fast search and yet obtaining a similar MTWV.

Fig. 11 show the alignment paths of FNS-DTW for $\alpha = 2, 4, 6$ (represented with dotted lines) in comparison with the alignment path of NS-DTW. The query and reference frames are reduced in FNS-DTW. For a graphical comparison with NS-DTW, the alignment path of FNS-DTW is stretched by a factor of α. From Fig. 11, it can be seen that the alignment path of FNS-DTW fluctuates around the alignment path of NS-DTW and the deviation is minimum for smaller values of α. This indicates that the *search hits* can be obtained by using FNS-DTW.

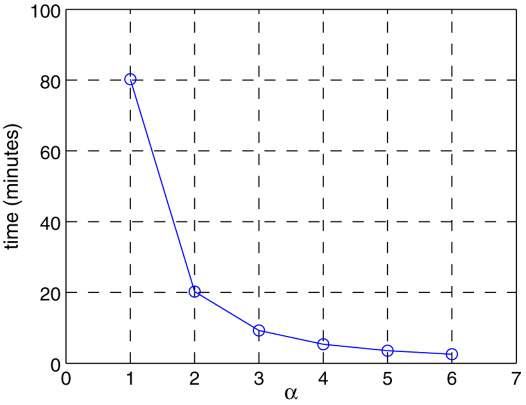


Fig. 12. Runtime of FNS-DTW for various $\alpha = 1, 2, 3, 4, 5, 6$ using dev dataset. This curve follows the trend of $\frac{1}{\alpha^2}$ due to computational upper bound of FNS-DTW being $O(\frac{mn}{\alpha^2})$.

Table V shows the MTWV using FNS-DTW for dev and eval datasets for various values of α. The alignment path of FNS-DTW is similar to that of NS-DTW for smaller values of α. Thus the performance of FNS-DTW is much better for $\alpha = 2$ as compared to other values of α.

Fig. 12 shows QbE-STD runtime for FNS-DTW and NS-DTW (FNS-DTW for $\alpha = 1$). In Fast NS-DTW, there is a trade-off between search performance and accuracy. However, for low values of α ($\alpha = 2$) the MTWV is comparable to the original system on the dev dataset and slightly worse on the eval dataset (as shown in Table V). From Fig. 12 it is evident that FNS-DTW is 4 times faster than NS-DTW for $\alpha = 2$.

[19] describes a fast indexing based search approach called Randomized Acoustic Indexing and Logarithmic-time Search (RAILS) whose results were reported for MediaEval 2012 database. RAILS technique is as follows: (a) Locality sensitive hashing for indexing the data, (b) Approximate nearest neighbor search for each query frame in logarithmic time and

TABLE VI
MTWV AND SPEEDUP FOR FNS-DTW AND RAILS EVALUATED ON DEV DATA.

	MTWV	**Speedup**
RAILS-I	0.381	1000X
FNS-DTW-I	0.399	1000X
RAILS-II	0.331	1600X
FNSDTW-II	0.353	4100X

constructing a similarity matrix, (c) Image processing techniques applied on the similarity matrix to obtain the *search hits*. The computation performance of the system was measured by the total size of the database in seconds divided by the average search time in seconds per query. The measure was referred to as *speedup*.

In [19], two search systems, RAILS-I and RAILS-II, were evaluated on MediaEval 2012 dev data and MTWV and *speedup* reported are shown in Table VI. From Table VI, it is shown the FNS-DTW-I (FNS-DTW for $\alpha = 2$) and FNS-DTW-II ($\alpha = 4$) are performing better than the RAILS system [19].

In [17], hierarchical K-Means clustering is used as an indexing technique and subsequently for computing the DTW scores. The *estimated speedup time* as reported on MediaEval 2012 dev data is 2400X with an MTWV of 0.364. In FNS-DTW with $\alpha = 4$, a *speedup* of 4100X is obtained with a slightly lower MTWV of 0.353 on the same dataset.

In other relevant works of [32], [33], a constraint based search was used to prune out the audio references. The pruning process was implemented by computing a lower bound estimate for DTW. It was shown that the computation of lower bound estimate is of the order $O(mn)$ [33]. Thus the total computational upper bound for such approaches would be $O(mn)$ plus the time taken to perform DTW alignment score. In our proposed fast NS-DTW, we use the reduced feature representation by averaging the successive Gaussian posteriorgrams. Thus the total computation time of fast NS-DTW would be $O(\frac{mn}{\alpha^2})$ plus the time taken to smooth the average the posteriorgrams. It should be noted that the fast NS-DTW is a one-stage process, whereas the lower bound estimate methods are implemented in two stages (pruning and score estimation).

VII. Conclusion and Future Work

In this paper we used a DTW based algorithm called non-segmental DTW (NS-DTW), with a computational upper bound of $O(mn)$. We have analyzed the performance of NS-DTW for query-by-example spoken term detection (QbE-STD) with Gaussian posteriorgrams obtained from different features of the speech signal. The results indicate that frequency domain linear prediction cepstral coefficients (FDLP), which capture the temporal dynamics of the speech signal, can be used as an alternative to traditional spectral parameters such as linear prediction cepstral coefficients (LPCC), perceptual linear prediction cepstral coefficients (PLP) and Mel-frequency cepstral coefficients (MFCC).

We have introduced a fast NS-DTW (FNS-DTW) which uses reduced Gaussian posteriorgrams for QbE-STD. We have shown that, for a given reduction factor $\alpha \in \mathbb{N}$, the computational upper bound of FNS-DTW is $O(\frac{mn}{\alpha^2})$. The reduction of the feature vectors was done via arithmetic mean and it was shown that for $\alpha = 2$, maximum term weighted values (MTWV) of FNS-DTW were similar or slightly lower to that of NS-DTW but three times faster.

We have also compared FNS-DTW with a fast indexing based search approach called Randomized Acoustic Indexing and Logarithmic-time Search (RAILS) whose results were reported for MediaEval 2012 database. It was shown that FNS-DTW was performing better than RAILS system with 0.353 MTWV search performance and a *speedup* of 4100X. One of the primary advantages of RAILS system over FNS-DTW is its indexing based technique to search over large databases and hence RAILS performance is better in terms of memory consumption. As a future work we plan to incorporate indexing based techniques in building sparse similarity matrix for FNS-DTW type of approach.

References

[1] I. Szöke, M. Fapso, L. Burget, and J. Cernocky, "Hybrid word-subword decoding for spoken term detection," in *Proc. Workshop Searching Spontaneous Conversational Speech*, 2008, pp. 4–11.

[2] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proc. HLT-NAACL*, 2004, pp. 129–136.

[3] D. R. H. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. INTERSPEECH*, 2007, pp. 314–317.

[4] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. ASRU*, 2009, pp. 398–403.

[5] C.-A. Chan and L.-S. Lee, "Unsupervised spoken-term detection with spoken queries using segment-based dynamic time warping," in *Proc. INTERSPEECH*, 2010, pp. 693–696.

[6] V. Gupta, J. Ajmera, A. , and A. Verma, "A language independent approach to audio search," in *Proc. INTERSPEECH*, 2011, pp. 1125–1128.

[7] A. Jansen and B. V. Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. ASRU*, 2011, pp. 401–406.

[8] X. Anguera, "Speaker independent discriminant feature extraction for acoustic pattern-matching," in *Proc. ICASSP*, 2012, pp. 485–488.

[9] X. Anguera and M. Ferrarons, "Memory efficient subsequence DTW for query-by-example spoken term detection," in *Proc. ICME*, 2013.

[10] K. Prahallad, A. R. Toth, and A. W. Black, "Automatic building of synthetic voices from large multi-paragraph speech databases," in *Proc. INTERSPEECH*, 2007, pp. 2901–2904.

[11] K. Prahallad and A. W. Black, "Segmentation of monologues in audio books for building synthetic voices," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 5, pp. 1444–1449, Jul. 2011.

[12] M. Müller, *Information Retrieval for Music and Motion*. New York, NY, USA: Springer., 2007.

[13] F. Metze, N. Rajput, X. Anguera, M. H. Davel, G. Gravier, C. J. V. Heerden, G. V. Mantena, A. Muscariello, K. Prahallad, I. Szöke, and J. Tejedor, "The spoken web search task at MediaEval 2011," in *Proc. ICASSP*, 2012, pp. 5165–5168.

[14] S. Thomas, S. Ganapathy, and H. Hermansky, "Recognition of reverberant speech using frequency domain linear prediction," *IEEE Signal Process. Lett.*, vol. 15, pp. 681–684, 2008.

[15] S. Ganapathy, S. Thomas, and H. Hermansky, "Temporal envelope compensation for robust phoneme recognition using modulation spectrum," *J. Acoust. Soc. Amer.*, vol. 128, pp. 3769–3780, 2010.

[16] M. Athineos and D. P. W. Ellis, "Autoregressive modeling of temporal envelopes," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5237–5245, Nov. 2007.

[17] G. Mantena and X. Anguera, "Speed improvements to information retrieval-based dynamic time warping using hierarchical k-means clustering," in *Proc. ICASSP*, 2013, pp. 8515–8519.

[18] E. Barnard, M. H. Davel, and C. J. V. Heerden, "ASR corpus design for resource-scarce languages," in *Proc. INTERSPEECH*, 2009, pp. 2847–2850.

[19] A. Jansen, B. V. Durme, and P. Clark, "The JHU-HLTCOE spoken web search system for MediaEval 2012," in *Proc. MediaEval*, 2012.

[20] H. Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” *J. Acoust. Soc. Amer.*, vol. 57, no. 4, pp. 1738–52, Apr. 1990.

[21] J. Makhoul, “Linear prediction: A tutorial review,” *Proc. IEEE*, vol. 63, no. 4, pp. 561–580, Apr. 1975.

[22] S. Ganapathy, “Signal analysis using autoregressive models of amplitude modulation,” Ph.D. dissertation, Johns Hopkins Univ., Baltimore, MD, USA, Jan. 2012.

[23] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.

[24] F. Itakura, “Minimum prediction residual principle applied to speech recognition,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-23, no. 1, pp. 67–72, Feb. 1975.

[25] F. Wessel and H. Ney, “Unsupervised training of acoustic models for large vocabulary continuous speech recognition,” *IEEE Trans. Speech Audio Process.*, vol. 13, no. 1, pp. 23–31, Jan. 2005.

[26] H. Ney and A. Noll, “Phoneme modelling using continuous mixture densities,” in *Proc. ICASSP*, 1988, pp. 437–440.

[27] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddington, “Results of the 2006spoken term detection evaluation,” in *Proc. Workshop Searching Spontaneous Conversational Speech*, 2007, pp. 45–50.

[28] F. Metze, E. Barnard, M. H. Davel, C. J. V. Heerden, X. Anguera, G. Gravier, and N. Rajput, “The spoken web search task,” in *Proc. MediaEval*, 2012.

[29] R. Singh, B. Lambert, and B. Raj, “The use of sense in unsupervised training of acoustic models for ASR systems,” in *Proc. INTERSPEECH*, 2010, pp. 2938–2941.

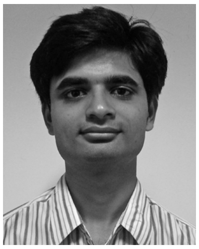
[30] A. Jansen and K. Church, “Towards unsupervised training of speaker independent acoustic models,” in *Proc. INTERSPEECH*, 2011, pp. 1693–1692.

[31] S. Salvador and P. Chan, “Toward accurate dynamic time warping in linear time and space,” *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, Oct. 2007.

[32] Y. Zhang and J. Glass, “An inner-product lower-bound estimate for dynamic time warping,” in *Proc. ICASSP*, 2011, pp. 5660–5663.

[33] P. Yang, L. Xie, Q. Luan, and W. Feng, “A tighter lower bound estimate for dynamic time warping,” in *Proc. ICASSP*, 2013, pp. 8525–8529.

Gautam Mantena (S’13) received the B.Tech. degree from Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, Gujarat, India, in 2006 and the M.S. in IT degree from International Institute of Information Technology, Hyderabad (IIIT-H), India, in 2008. He is currently pursuing the Ph.D. degree from the Speech and Vision Lab, IIIT-H. His research interests include spoken audio search, spoken dialogue systems and speech recognition.



Sivanand Achanta received the B.Tech. degree in Electronics and Communication from Kamala Institute of Technology and Science, Karimnagar, India in 2010. He is currently pursuing the Ph.D. degree from the Speech and Vision Lab, International Institute of Information Technology, Hyderabad. His research interests include speech signal processing, machine learning and speech synthesis.

Kishore Prahallad (M’07) received the B.E. degree from the Deccan College of Engineering and Technology, Osmania University, Hyderabad, India, in 1998, the M.S. (by Research) degree from the Indian Institute of Technology (IIT) Madras, in 2001 and the Ph.D. degree from the Language Technologies Institute, School of Computer Science, Carnegie Mellon University (CMU), Pittsburgh, USA, in 2010. He is an Associate Professor at the International Institute of Information Technology, Hyderabad (IIIT-H). He has been associated with IIIT-H since March 2001, and started the speech activities in Language Technologies Research Center at IIIT-H. His research interests are in speech and language processing, multimodal mobile computing and interfaces, and artificial neural networks.