Sparse Subspace Modeling for Query by Example Spoken Term Detection

Dhananjay Ram, Afsaneh Asaei, *Senior Member, IEEE,* and Hervé Bourlard, *Fellow, IEEE*

***Abstract*—This paper focuses on the problem of query by example spoken term detection (QbE-STD) in zero-resource scenario. Current state-of-the-art approaches to tackle this problem rely on dynamic programming based template matching techniques using phone posterior features extracted at the output of a deep neural network (DNN). Previously, it has been shown that the space of phone posteriors is highly structured, as a union of low-dimensional subspaces. To exploit the temporal and sparse structure of the speech data, we investigate here three different QbE-STD systems based on sparse model recovery. More specifically, we use query examples to model the query subspace using dictionary for sparse coding. Reconstruction errors calculated using sparse representation of feature vectors are then used to characterize the underlying subspaces. The first approach uses these reconstruction errors in a dynamic programming framework to detect the spoken query, resulting in a much faster search compared to standard template matching. The other two methods aim at merging template matching and sparsity based approaches to further improve the performance. The first one proposes to regularize the template matching local distances using sparse reconstruction errors. The second approach aims at using the sparse reconstruction errors to rescore (improve) the template matching likelihood. Experiments on two different databases (AMI and MediaEval) show that the proposed hybrid systems perform better than a highly competitive QbE-STD baseline system.**

***Index Terms*—Speech processing, spoken term detection, query by example, deep neural network, posterior probabilities, sparse recovery modeling, sparse representation, subspace detection, subspace regularization.**

I. Introduction

Query-by-example spoken term detection (QbE-STD) refers to the task of finding a specific spoken query within an audio archive. Typically, the user generates a spoken query, and the search algorithm attempts to retrieve all audio documents containing the query from the searched archive. In this scenario, no training data is provided, making it a zero-resource task. Thus, the data can be generated in any language with no constraints on vocabulary, pronunciation lexicon, accents etc. It is essentially a pattern matching problem in the context of speech data where the targeted pattern is the information encoded using speech signal and presented to the system as a spoken query. The difference between keyword spotting and QbE-STD is that the former deals with textual queries, whereas the latter deals with spoken queries.

The solution to QbE-STD can be very useful in searching through audio archives which consist of data from news channels, radio broadcasts, internet, social media etc. Most search algorithms used in practice still depend on a textual description of data which may not be always available or it is insufficient for representing the complete content of data. Therefore, text based retrieval algorithms gives very limited search results. Moreover, it is desirable to search through those contents using speech as a natural and generic medium of communication, and not requiring any explicit transcript.

Traditionally, the spoken query detection is performed by cascading an automatic speech recognition (ASR) system with text based retrieval techniques [1], [2], [3], [4]. In this approach, the spoken queries as well as the test utterances are first converted into a sequence of words or symbols. Information retrieval methods are then applied to detect the queries. Recent approaches for keyword spotting have focused on low-resource languages, exploiting unsupervised acoustic models [5], language modeling with automatically retrieved web documents [6], or multilingual bottleneck features [7]. However, this is still a language dependent system which is unsuitable for detecting spoken queries from speech data of unknown languages.

Current approaches for QbE-STD are largely dominated by template matching techniques for their superior performance to the statistical methods in zero-resource conditions [3], [8], [9]. Such approaches mainly consist of two steps: feature extraction and template matching. A Dynamic Time Warping (DTW) algorithm [10] is generally used to find the degree of similarity between a query and a test utterance. The goal is to develop an unsupervised method so that the data can be processed without any transcription. This alleviates the problems associated with the ASR system for spoken term detection. Currently, the best performing system uses a DTW based template matching technique to find the queries [9].

The DTW-based approaches are able to consider the sequential information present in a spoken query. However, they do not take into account the low-dimensional subspace structure of the speech signal. This low-dimensional structure is the result of the constrained articulatory mechanism of human speech production [11], [12], which leads to the generation of linguistic units (e.g., phones, senones) lying on non-linear manifolds. As already shown in [13], [14], [15], these manifolds can be modeled as a union of low-dimensional subspaces and sparse representation is found to be a promising technique to model these subspaces. It is the goal of the present work to investigate how this sparsity property can be exploited to further improve state-of-the-art QbE-STD system.

The present work is motivated by the success of exemplar-based sparse representation in detection and classification

Authors are with Idiap Research Institute, Centre du Parc, Rue Marconi 19, 1920 Martigny, Switzerland. D. Ram and H. Bourlard are also with École Polytechnic Fédérale de Lausanne (EPFL), Switzerland. Email: firstname.lastname@idiap.ch

tasks [16], [17], which relies on the low-dimensional subspace structure of the data. In the context of speech processing, sparse recovery has already been studied for robust speech recognition [18], [19], [20], enhanced acoustic modeling [14], [15] as well as spoken query detection [13], [21], [22]. In our earlier work [13], we cast the query detection problem as subspace detection between query and non-query speech where the corresponding subspaces are modeled through dictionary learning for sparse representation. Given these dictionaries, detection of each frame is performed based on the ratio of the two corresponding sparse representation reconstruction errors, and the frame-level decisions are accumulated by counting the continuous number of frames detected as the query. Although this approach shows a promising direction, it lacks a proper framework to capture the temporal information inherent to speech signal. Also, it relies on the background dictionaries to model non-query speech which is usually not available for QbE-STD.

Building upon the above discussion, the present work explores new systems designed to take advantage of both *temporal information* and *subspace structure* of speech. The primary contribution of this paper is to show the effectiveness of subspace structure of speech data for finding a spoken query in a test utterance. In this context, a query is modeled through dictionary which can be built from single as well as multiple query examples. We present three different ways to achieve our goal, as discussed in the following:

1) *Sparse Subspace Detection (Section V)*: This approach relies on modeling the low-dimensional structure of sub-phonetic components of the query. These subspaces are modeled using dictionaries for sparse coding. The dictionaries are used to obtain a frame level sparse representation which quantifies the errors to reconstruct those frames. We propose to use a dynamic programming technique to obtain possible regions of occurrence of the query in test utterances. This reduces the effect of errors made by the sparse coding algorithm on frame level and captures the sequential information present in the data. It is a much faster technique compared to DTW based methods.
2) *Subspace Regularized DTW (Section VI-A)*: In this approach, we use both sparsity and DTW to make a better system, instead of relying solely on either of them to perform the same task. The idea is to consider the frame-level reconstruction errors as subspace based distances. We propose to regularize the distance matrix for DTW using this subspace based distance and perform DTW to detect the query. This regularization helps to take into account the temporal information as well as the subspace structure of speech signal.
3) *Subspace-Based Rescoring of DTW (Section VI-B)*: In another approach, we propose to rescore the hypothesized regions obtained from the DTW system using sparsity based system. This method aims at improving the likelihood score for a hypothesized region using the subspace structure of speech signal.

In all three cases discussed above, we rely on the low-dimensional subspace structure of speech signal for the task of QbE-STD. The systems proposed in this paper indicate different ways of utilizing this information. These systems are evaluated on two different databases with challenging conditions as we will see in Sections VIII and IX. The performance improvements provided by the combination of DTW and sparsity based systems show the importance of subspace structure of speech to perform QbE-STD.

II. Prior Works

In this section, we briefly discuss the main approaches to spoken query detection. The first set of approaches are referred to as template matching, which consists of two steps. First, the spoken query and test utterances are represented in terms of feature vectors. Both spectral [23], [24] and class-conditional posterior probabilities [9], [25] are used as features. The posteriors probabilities can be estimated from Gaussian mixture model (GMM) [25] as well as deep neural network (DNN) [9], [26]. Once we have extracted the features from both query and test utterance, a dynamic programming algorithm [10] is used to detect the query in a test utterance. Standard DTW algorithm finds a non-linear mapping between two sequences of feature vectors. The similarity score is computed using the optimal warping path between them. But, this is not exactly applicable to spoken query search because, the query can occur anywhere in the test audio as a sub-sequence. Thus, several variants of DTW have been proposed for QbE-STD.

Segmental DTW [23], [25] is a constrained dynamic programming technique used to detect a specific sub-sequence in a test utterance matching the spoken query. The first constraint is to keep the warping path in a pre-defined window to match the query as a sub-sequence. The second constraint is the step size of this window, which indicates the start of a matching subsequence. This method cannot handle utterances with widely varying speaking rate due to the restricted warping path. slope-constrained DTW [27] is proposed to deal with this problem. In this case, the slope of the warping path is constrained to allow the mapping of a query frame to multiple test frames and vice-versa, but not both at the same time. Also, it penalizes the mapping of one frame to multiple frames by introducing a slope factor to the similarity score. Another approach to find queries in a test utterance is called sub-sequence DTW [28]. This algorithm enforces the cost of insertion at the beginning and end points of the query to be equal to 0. It encourages the warping path to start and end at any frame of the test utterance, which gives us a sub-sequence matching the spoken query. These DTW based approaches are computationally expensive. Several methods [29], [30], [31], [32] have been proposed to speed up the process. In [29], the authors proposed a DTW technique to be used on graphical processing units for faster computation. On the other hand, information retrieval-based DTW [30] is proposed to index frames of speech using hashing techniques to reduce the search space.

The template matching approaches discussed above are sensitive to speaker and acoustic mismatch conditions. To overcome these limitations, model-based approaches have been investigated [24], [33], [34]. These approaches primarily rely

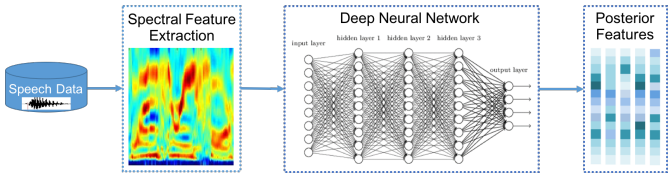

Fig. 1. Posterior feature extraction using a deep neural network. First, Mel Frequency Cepstral Coefficient (MFCC) based features are extracted over a sliding window. These features, together with their acoustic context, are then fed to a DNN to estimate phone conditional posterior probabilities.

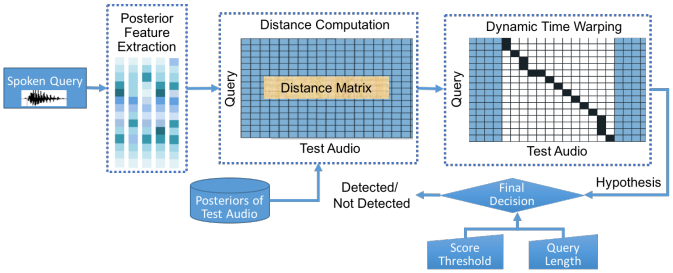

Fig. 2. Block diagram of the baseline system. After extracting phone posterior features to calculate the normalized distance matrix between query and test utterance, we apply DTW to obtain a sub-sequence matching the query. If the length of the hypothesis is smaller than half the query length, it is discarded to reduce false alarm rate. Otherwise, its score is compared to a threshold to yield a final decision.

on acoustic units discovered in an unsupervised manner. Those units are then used to train hidden Markov models (HMM) for the corresponding acoustic modeling. The resulting HMMs are used to find symbolic representation for both query and test utterance, and retrieval techniques are applied to perform QbE-STD.

III. Baseline System

The posterior-based QbE-STD system proposed in [9], and briefly discussed below, is used as our baseline system. It was the best system in MediaEval challenge 2013 [35] for the task of Spoken Web Search (SWS). The basic framework of the system is presented in this section. It consists of two blocks: posterior feature extraction and template matching as discussed in the following.

A. Posterior Features

Posterior features (e.g. mono-phone, tri-phone) are typically extracted at the output of a Deep Neural Network (DNN) [26] with spectral features as input. This type of speech features have been shown to be very effective for ASR systems which motivated the researchers to use them for template matching tasks.

Mono-phone based posterior probabilities are used as feature vectors in the baseline system. The setup for extracting the posterior features is illustrated in Figure 1. In the first step, mel frequency cepstral coefficient (MFCC) based features are computed from the speech signal over a temporal sliding window. Those acoustic features, together with some acoustic (left and right) context, are then fed to a DNN trained to estimate output class conditional posterior probabilities. Alternatively, Convolutional Neural Network (CNN) can also be used at the first layer of the DNN to better capture correlation over time (and frequency in the case of spectral features).

B. Query Template Construction

The posterior features of a query are used to construct a template for DTW based matching. If there is only one example provided for a query, the corresponding posteriors are used as the reference template for performing DTW. If multiple examples are provided for a query, we compute an average template from posteriors of those examples using DTW. In that case, we first select the example with highest number of posteriors as reference. We then use traditional DTW algorithm [36] to obtain posteriors-level alignment of the rest of the examples with the reference. The mapped posteriors are averaged together to generate the posteriors of the reference template [9], [37]. Finally, this template is used to find the query in test utterances as discussed in the following section.

C. Template Matching

The template matching algorithm presented in [9] is similar to the slope-constrained DTW [27] with some important differences. First, a distance matrix is calculated between each pair of frames of the query and test utterance using logarithm of the cosine distance. The distances are then normalized to be between 0 and 1. Dynamic programming is performed using this distance matrix where the optimal cost at each step is normalized by the corresponding partial path length. Also, it imposes constraints such that the warping path can start and end anywhere in the test utterance giving us a sub-sequence which optimally matches the query. The resulting hypothesis is then filtered depending on its length to reduce the false alarms. If the length of a hypothesis is less than half of the query length, it is discarded since small portions of the test utterance can match well with query segments and produce a high likelihood score. Finally, the score of a hypothesis is compared with a pre-defined threshold to decide the occurrence of the query. A block diagram of this system is presented in Figure 2 to find a spoken query in a test utterance.

IV. Subspace Modeling

In this section, we describe the modeling of subspaces of query exemplars for sparse representation. We start by describing the sparse representation of posterior feature vectors (as a sparse linear combination of an over-complete dictionary posteriors) before discussing different methods to construct dictionaries for query modeling.

A. Sparse Representation

When speech is represented in terms of posterior probabilities, the subspace corresponding to each sub-word class is a low-dimensional space [14], [38]. Accordingly, a speech utterance comprised of sub-word classes, can be modeled as a union of low-dimensional subspaces. Any data point in a union

of low-dimensional subspaces can be efficiently reconstructed by a sparse combination of other points in that space, a property referred to as the self-expressiveness [39] of data.

Let $\mathbf{y}_t$ be a posterior feature vector for a speech frame at time t. Each posterior vector $\mathbf{y}_t$ is a K dimensional feature vector where each dimension corresponds to a speech unit. These speech units (associated with DNN outputs) can be phones (context dependent or independent), senones, or any other sub-word unit. Following the self-expressiveness property of data, the feature vector $\mathbf{y}_t$ can be represented as a sparse linear combination of other feature vectors present in the training set, $\{\mathbf{d}_i\}_{i=1}^{N}$ corresponding to the query subspace, i.e.:

$$\begin{aligned}\mathbf{y}_t &\approx \alpha_1\mathbf{d}_1 + \alpha_2\mathbf{d}_2 + \ldots + \alpha_N\mathbf{d}_N \\ &= \underbrace{[\mathbf{d}_1 \quad \mathbf{d}_2 \quad \ldots \quad \mathbf{d}_N]}_{\mathbf{D}} \times \underbrace{[\alpha_1 \quad \alpha_2 \quad \ldots \quad \alpha_N]^\top}_{\boldsymbol{\alpha}_t} \\ &= \mathbf{D}\boldsymbol{\alpha}_t\end{aligned} \tag{1}$$

where N is the number of training samples (basis vectors) used to model the query subspace, $\mathbf{D}$ is a matrix of size $K \times N$ which consists of basis vectors $\mathbf{d}_i$ of the query subspace, and $\boldsymbol{\alpha}_t$ is the weight vector indicating the significance of each basis vector in construction of a test posterior. The matrix $\mathbf{D}$ is called a dictionary matrix and the columns of this matrix are called atoms. The weight vector $\boldsymbol{\alpha}_t$ is sparse, i.e., having very few non-zero entries. The non-zero entries correspond to the underlying low-dimensional subspaces which the test posterior belongs to.

The framework of sparse representation introduced above in (1) relies on construction of the dictionary matrix $\mathbf{D}$. Given this dictionary for characterizing the underlying subspaces, the independent subspaces are guaranteed to be identified correctly using sparse representation [39]. In the following section, we explain how these subspaces can be modeled using dictionary for sparse representation.

B. Dictionaries for Subspace Modeling

Dictionaries for sparse representation are constructed using an over-complete set of basis vectors obtained from the training examples of corresponding classes. These dictionaries can be modeled primarily in the following two ways:

1) *Concatenation of training examples*: In this method, we take the features of all available training examples for a desired class and concatenate them to build the dictionary [18], [19]. This method is more suitable in a scenario when very few training examples are available for a class. On the other hand, with huge number of training examples, the size of the dictionary can grow very large. This, in turn, can increase the computational complexity of the sparse coding algorithm. A method to extract all the information present in the training data without increasing the size of the dictionary to an undesirable magnitude is thus required.
2) *Learning from training examples*: Dictionary learning refers to the task of learning an over-complete set of basis vectors from the training exemplars such that each training exemplar can be reconstructed as a sparse linear combination of the dictionary vectors (atoms). These dictionaries can be learned by solving an optimization problem which gives the best approximation of training vectors while keeping the degree of sparsity on desirable level as discussed in the following.
Let us have a set of T training vectors with $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T\}$, and their sparse representations using dictionary $\mathbf{D} \in \mathbb{R}^{K\times M}$ with $\mathbf{A} = \{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_T\}$, where K is the dimension of exemplar vectors, and M is the number of dictionary atoms, the objective function for dictionary learning is defined as

$$\arg\min_{\mathbf{D},\mathbf{A}} \frac{1}{T}\sum_{t=1}^{T}\left(\frac{1}{2}\|\mathbf{y}_t - \mathbf{D}\,\boldsymbol{\alpha}_t\|_2^2 + \lambda\|\boldsymbol{\alpha}_t\|_1\right) \tag{2}$$

where λ is the regularization parameter. The first term in this expression, quantifies the *reconstruction error*. The second term denotes the ℓ_1-norm of $\boldsymbol{\alpha}$ defined as $\|\boldsymbol{\alpha}\|_1 = \sum_i |\alpha_i|$ which quantifies the sparsity of $\boldsymbol{\alpha}_t$. The joint optimization of this objective function with respect to both $\mathbf{D}$ and $\boldsymbol{\alpha}_t$ simultaneously is non-convex, it can be solved as a convex objective by optimizing for one while keeping the other fixed [40].

In case of QbE-STD, we represent each query as a class to be modeled using dictionary. The training data for these dictionaries is obtained by extracting posterior features from the spoken instances of corresponding query as discussed in Section III-A. These are the same posterior features used to construct the query templates for DTW in the baseline system. We consider two cases to construct a dictionary depending on the number of examples available for a given query. If there is only one example available per query, the corresponding posterior feature vectors constitute the dictionary. Whereas with multiple examples per query, we either concatenate the posteriors of these examples to construct a dictionary or use these posteriors to learn a dictionary. In case of learning a dictionary, we initialize the dictionary with posteriors of the example having highest number of frames. Posteriors from rest of the examples are used to learn the dictionary according to (2).

V. Sparse Subspace Detection (SSD)

Once the query subspaces are modeled, the QbE-STD problem is cast as a subspace detection problem where the reconstruction errors of the respective sparse representations are used to detect the underlying subspaces. Given a test posterior feature vector $\mathbf{y}_t$ and the query dictionary $\mathbf{D}$, the test vector can be represented as a sparse linear combination of dictionary atoms characterizing the query. The sparse representation is obtained by solving the following optimization problem:

$$\boldsymbol{\alpha}_t = \arg\min_{\boldsymbol{\alpha}}\left\{\frac{1}{2}\|\mathbf{y}_t - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1\right\} \tag{3}$$

where λ is the regularization parameter. The first term in this expression quantifies the *reconstruction error*. The second term denotes the ℓ_1-norm of $\boldsymbol{\alpha}$ defined as: $\|\boldsymbol{\alpha}\|_1 = \sum_i |\alpha_i|$ which quantifies the level of sparsity in the co-efficient vector, $\boldsymbol{\alpha}_t$. In order to exploit the temporal information inherent to

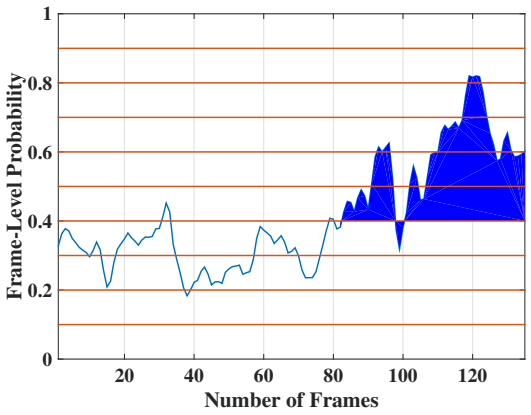

Fig. 3. Frame-level probability and different thresholds for an utterance containing "SO THAT CONCLUDES MY PRESENTATION". The query speech contains "PRESENTATION"

speech signal, a sequence of c posterior feature vectors are concatenated to form a contextually rich vector for dictionary construction as well as sparse representation as follows,

$$\tilde{\mathbf{y}}_t = [\mathbf{y}_{t-c}^\top \cdots \mathbf{y}_t^\top \cdots \mathbf{y}_{t+c}^\top]^\top \tag{4}$$

This mechanism is referred to as *context appending* which is a typical approach to incorporate the dynamics of exemplars [19], [41]. This context is a system parameter to be optimized using development queries.

The reconstructed vectors using the corresponding sparse representations will be: $\hat{\mathbf{y}}_t = \mathbf{D}\boldsymbol{\alpha}_t$. The subspace which can best represent a test vector $\mathbf{y}_t$ corresponds to the least reconstruction error [17], [21]. Hence, we use the Euclidean-norm based reconstruction error to perform QbE-STD at a later stage. The reconstruction errors are calculated as follows:

$$e(\mathbf{y}_t) = \frac{1}{2}\|\mathbf{y}_t - \hat{\mathbf{y}}_t\|_2 = \frac{1}{2}\|\mathbf{y}_t - \mathbf{D}\boldsymbol{\alpha}_t\|_2 \tag{5}$$

We use these reconstruction errors to calculate frame-level empirical probabilities of the query occurring in a test utterance as: $p(\mathbf{y}_t) = 1 - e(\mathbf{y}_t)$. These frame-level probabilities constitute a probability curve indicating the possibility of the query occurring in a test utterance. We perform a non-linear smoothing to compensate for the potential errors made by the sparse coding system. The probability curve for an example utterance is shown in Figure 3. In order to identify a hypothesized region of occurrence from this curve, we use Kadane's algorithm [42], a very simple dynamic programming technique with linear time complexity to obtain a contiguous sub-array within an one-dimensional array of numbers which has the largest sum. In our case, we subtract a threshold value from the probability curve to get an array of numbers. This threshold provides a trade-off between the missed detection and false alarm rate. Subsequently, we apply Kadane's algorithm to obtain a sub-array with the largest sum, which essentially indicates the hypothesized region. The area under this segment of the curve represents the likelihood score. We normalize this score with the length of the hypothesized region. Later, we compare the length of the hypothesized region with half the length of the query and reject the ones having a smaller length in order to reduce false alarms. A comprehensive block diagram for the proposed system is presented in Figure 4. The steps to implement the system is summarized in Algorithm 1.

Algorithm 1 Sparse Subspace Detection (SSD) (Fig. 4)

Input: Spoken query and posteriors of a test utterance
Output: Decision if the query occurs in the test utterance

1: Extract the posterior features from spoken query
2: Perform context appending according to (4) for both query and test posteriors
3: Construct a dictionary by concatenating the posteriors from different examples or learn a dictionary using (2)
4: Compute sparse representation of test posteriors using the dictionary according to (3)
5: Compute reconstruction error using (5)
6: Use Kadane's algorithm to find out a hypothesized region and corresponding score
7: Use query length and score threshold to make a final decision

The QbE-STD system developed in this section does not use DTW to find a query speech in a test utterance. However, the proposed system is not able to capture the temporal information so well as compared to a DTW based system as we will see in SectionVIII. Thus, we propose new approaches to build hybrid systems in the following section which will be able to combine the positive aspects of both systems.

VI. Sparse-DTW Hybrid Systems

In this section we propose two different ways to incorporate information coming from a DTW system and the sparsity based system discussed above. The first method implements a system-level fusion, whereas the second method performs a re-scoring of the hypotheses from the DTW system using sparsity. We describe these systems in the following.

A. Subspace Regularized DTW (SR-DTW)

The system presented here relies on the notion that the reconstruction error for a test frame can be considered as distance between the query subspace and the corresponding test frame [22]. In this method, we propose to use the subspace based distance to regularize the distance matrix for DTW as shown in Figure 5. Let us consider, $\mathbf{X} = [\,\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\,]$ represent the posterior feature vectors corresponding to the query speech and $\mathbf{Y} = [\,\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\,]$ corresponding to a test utterance. Here, m and n represent the number of frames in the query speech and test utterance respectively. The distance matrix (Δ) used for DTW can then be calculated as follows:

$$\Delta(i,j) = d(\mathbf{x}_i, \mathbf{y}_j) \quad \forall\, i = 1, 2, \ldots, m \\ \text{and } j = 1, 2, \ldots, n \tag{6}$$

where $d(.,.)$ is a standard distance measure such euclidean, cosine, etc.

On the other hand, the subspace based distance (reconstruction error $e(j)$) for a test frame $\mathbf{y}_j$ can be calculated using (3) and (5). We observe that each column of this distance

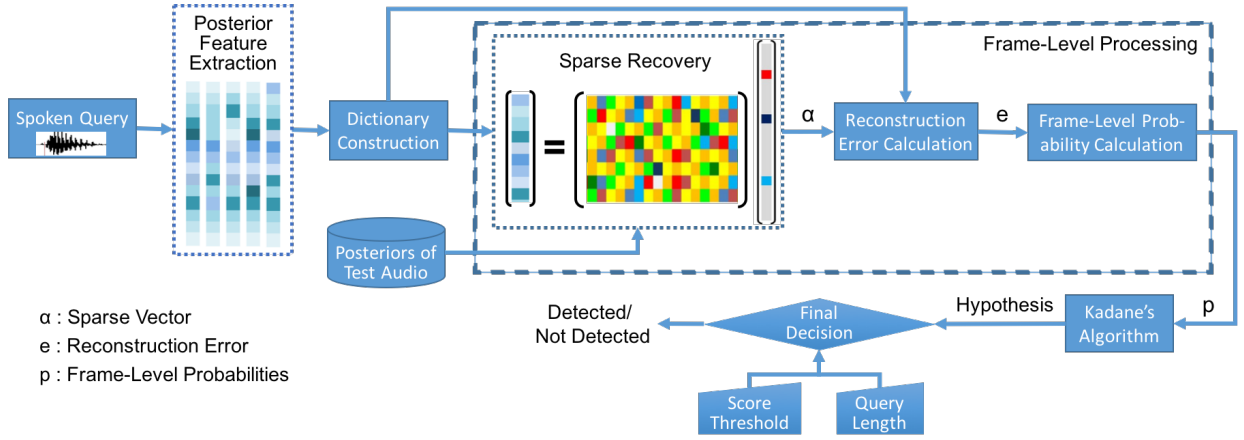

Fig. 4. Block diagram of the sparse subspace system. We first extract posterior features from the query and use it to construct a dictionary. We employ this dictionary to compute the sparse representation and corresponding reconstruction error for each frame of test audio. Exploiting these reconstruction errors, we use Kadane's algorithm [42] to hypothesize the region of occurrence of the query and calculate the likelihood score. If the length of the hypothesis is smaller than half the query length, it is discarded to reduce false alarm rate. Otherwise, the hypothesis score is compared to a threshold to yield a final decision.

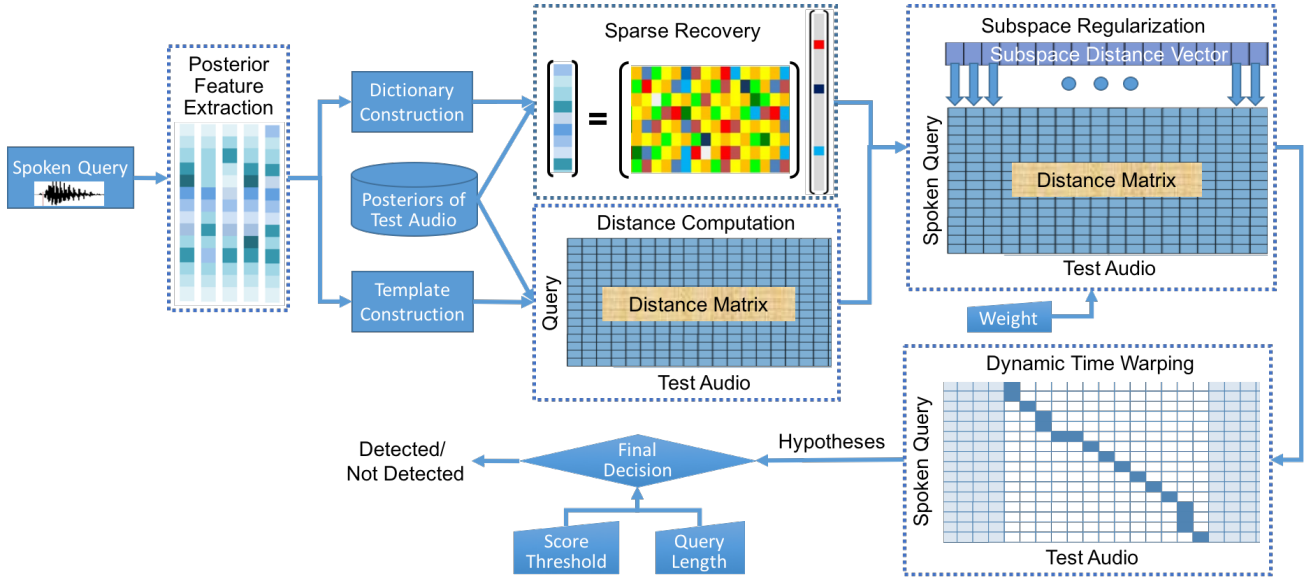

Fig. 5. Block diagram of the Subspace Regularized DTW system. We first extract posterior features from the query and use it to construct a dictionary and a template. The dictionary is used to calculate reconstruction errors for each frame of test audio to generate the subspace based distance vector. The distance matrix for DTW is computed using the query template and test posteriors. Each column of the distance matrix is then regularized using the errors from sparse recovery. DTW is finally applied to obtain a hypothesis. If the length of the hypothesis is smaller than half the query length, it is discarded to reduce false alarm. Otherwise, the hypothesis score is compared to a threshold to yield a final decision.

matrix corresponds to the frame-level distance between a test frame and all frames of the query whereas we only have one number representing the distance from a test frame to the query subspace as a whole. The DTW distance matrix is then regularized by replacing each of its columns by a weighted average of each element in this column and the subspace based distance obtained using the same test frame:

$$\begin{aligned} \Delta_{reg}(i,j) = w_d \times \Delta(i,j) + (1-w_d) \times e(j) \\ \forall\, i = 1,2,\ldots,m \\ and\ j = 1,2,\ldots,n \end{aligned} \tag{7}$$

where Δ_{reg} is the regularized distance matrix and w_d is a fixed regularization weight parameter, which will be optimized on an independent query development set. We then perform dynamic programming on this regularized distance matrix of $\Delta_{reg}(i,j)$ in a similar manner as the baseline system [9] to obtain a region of occurrence of the query and calculate the likelihood of its occurrence. The whole procedure to implement this system is presented in Algorithm 2.

The key idea behind the proposed method is that the frame-level DTW exploits local similarities and properly models the temporal information, while the subspace-based distance captures the similarity at the subspace-level, which considers all frames present in the query for each test frame. A combination of these two distances is then shown to provide better decision likelihoods, resulting in performance improvement.

In principle, this approach is applicable to any variant of DTW by regularizing the corresponding distance matrix.

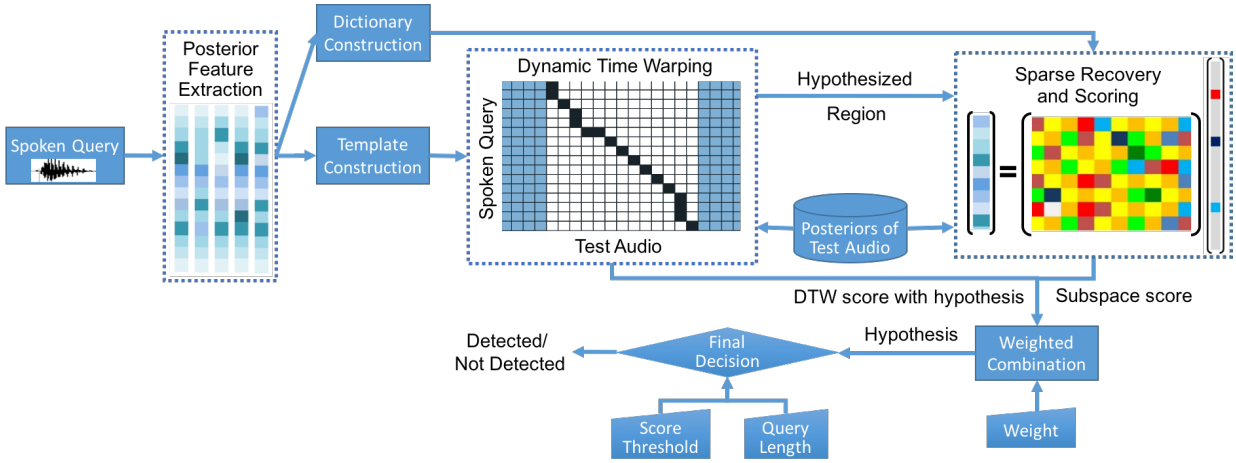

Fig. 6. Block diagram of the system for subspace based re-scoring of DTW. We first extract posterior features from the query and use it to construct a dictionary and a template. The template is used in the baseline DTW system to hypothesize the region of occurrence of the query and obtain a likelihood score. We obtain sparse representation of the hypothesized region and compute subspace score using corresponding reconstruction errors. The final score is then calculated by taking a convex combination these two scores. If the length of the hypothesis is smaller than half the query length, it is discarded to reduce false alarm. Otherwise, the hypothesis score is compared to a threshold to yield the final decision.

Algorithm 2 Subspace Regularized DTW (SR-DTW) (Fig. 5)

Input: Spoken query and posteriors of a test utterance

Output: Decision if the query occurs in the test utterance

1: Extract the posterior features from spoken query
2: Perform context appending according to (4) for both query and test posteriors
3: Construct a dictionary by concatenating the posteriors from different examples or learn a dictionary using (2)
4: Construct a template for DTW as discussed in Section III-B
5: Compute sparse representation of test posteriors using (3) and corresponding reconstruction errors using (5)
6: Construct a distance matrix by computing a normalized cosine distance between each pair of posteriors from query and test utterance as discussed in Section III-C.
7: Regularize the distance matrix using the reconstruction error according to (7)
8: Perform DTW to make a decision as described in Section III-C.

However, in this work, and to provide us with strong reference points, we implemented the system presented in [9] and perform the proposed regularization over the distance matrix followed by dynamic programming to obtain the detection regions along with their likelihood scores.

B. Subspace Based Rescoring of DTW (SRS-DTW)

In this section, we investigate another approach to integrate information from sparsity into DTW based systems. Instead of regularizing the distance matrix, we propose to re-score the hypothesized region obtained from DTW using sparse coding.

Considering the spoken query $\mathbf{X}$ and the test audio $\mathbf{Y}$, we apply the modified DTW algorithm (as explained in Section III-C) to obtain a hypothesized region denoted as $\mathbf{Y}_{hyp} = [\,\mathbf{y}_a, \mathbf{y}_{a+1}, \ldots, \mathbf{y}_{b-1}, \mathbf{y}_b\,]$ and the corresponding normalized similarity score is S_{DTW}. Then we construct a dictionary for the query using one of the methods discussed in Section IV-B. We use this dictionary in (3) to generate sparse representation for each frame of the query and employ those representations in (5) to calculate the reconstruction errors for each frame y_i of the hypothesized region. The resulting error vector is represented as $e_{a,b} = [\,e_a, e_{a+1}, \ldots, e_{b-1}, e_b\,]$, which is used to calculate another score for the hypothesized region, $\mathbf{Y}_{hyp}$ as follows,

$$S_{Subspace} = 1 - \frac{1}{b-a+1} \sum_{i=a}^{b} e_i \tag{8}$$

We call it subspace based score which represents average similarity between the hypothesized region and the spoken query using subspace structure of speech. Once we have scores from both systems, we take a weighted average as follows:

$$S = w_s \times S_{DTW} + (1 - w_s) \times S_{Subspace} \tag{9}$$

where S is the final similarity score between the hypothesized region and the spoken query, and w_s represents the associated weight. A block diagram of this proposed re-scoring mechanism is presented in Figure 6. Also, a step-by-step summary for implementing the system is described in Algorithm 3.

VII. Experimental Set-up

We use two different databases to evaluate and analyze the proposed approaches: the AMI meeting corpus [43] and the MediaEval 2013 spoken web search (SWS 2013) corpus [35]. A brief description of these two databases is presented in this section, before discussing the posterior feature extraction and the pre-processing steps involved. Finally, we describe different evaluation metrics used for our experiments.

A. AMI Meeting Corpus

The AMI meeting corpus [43] is used for the experiments where the training, development and evaluation sets are as

Algorithm 3 Subspace Based DTW Rescoring (SRS-DTW) (Fig. 6)

Input: Spoken query and posteriors of a test utterance
Output: Decision if the query occurs in the test utterance

1: Extract the posterior features from spoken query
2: Perform context appending according to (4) for both query and test posteriors
3: Construct a dictionary by concatenating the posteriors from different examples or learn a dictionary using (2)
4: Construct a template for DTW as discussed in Section III-B
5: Perform DTW using the template to obtain a hypothesized region and corresponding score, S_{DTW} as described in Section III-C
6: Compute sparse representation of test posteriors of the hypothesized region using (3)
7: Compute corresponding reconstruction errors according to (5) and use it to obtain the subspace based score, $S_{Subspace}$ according to (8)
8: Compute the final score using S_{DTW} and $S_{Subspace}$ according to (9)
9: Use query length and score threshold to make a final decision

described in [44]. This database contains meeting recordings in English where many participants were non-native speakers, and provides us with about 81 hours for DNN training and about 9 hours for each of the development and test data. Also, the headset recordings contain considerable amount of overlapping speech (competing speakers) which makes the QbE-STD task even more challenging. There are approximately 12k words in the training, out of which we extracted 200 more frequent words (excluding functional words) for our detection experiments including very short words such as 'ADD' to long words such as 'TECHNOLOGY'. Later, these queries are divided into two groups in a random manner to obtain sets of 100 queries each. One set is used as development queries to optimize the parameters of different systems whereas the other set is used to evaluate the performance of corresponding system. We use the test set of AMI as the search database for QbE-STD evaluation which contains 12612 utterances.

B. MediaEval 2013 Spoken Web Search (SWS 2013)

This database is part of the MediaEval benchmarking initiative [35] for evaluating spoken query detection systems. It consists of audio recordings from 9 different low-resourced languages: Albanian, Basque, Czech, non-native English, Isixhosa, Isizulu, Romanian, Sepedi and Setswana. These recordings were collected from many different sources with varying acoustic conditions and different amounts of data corresponding to different languages. The variety of data reduces the possibility of over-fitting on the development and evaluation query sets. There are 505 queries in the development set and 503 queries in the evaluation set. These sets are mutually exclusive. There are 3 types of queries depending on the number of examples available per query. The number of queries available in each category is shown in Table I. The search space consists of 20 hours of audio with 10762 utterances.

TABLE I
NUMBER OF DIFFERENT TYPES OF QUERIES AVAILABLE IN DEVELOPMENT AND EVALUATION SETS WHICH ARE PARTITIONED ACCORDING TO THE NUMBER OF EXAMPLES PER QUERY.

Query Set	Examples per query		
	1	3	10
Development	311	100	94
Evaluation	310	100	93

C. Feature Extraction and Pre-processing

We have used the setup presented in Section III-A to extract phone posterior features for our detection experiments. The setup corresponding to different databases is implemented separately as described below.

1) AMI Phone Recognizer: The posterior features are extracted from a DNN with Mel Frequency Cepstral Coefficients (MFCC) based spectral features as input. These spectral features are computed from small segments of speech signal obtained by applying triangular overlapping temporal windows of 25ms with an overlap of 15ms on a speech utterance. Additionally, 'delta' and 'delta-delta' coefficients are calculated to account for the dynamics of each segment and appended to the MFCCs to obtain 39 dimensional feature vectors. To add contextual information, 4 frames of left and right acoustic contexts are appended (total 9 frames) to have a 351 dimensional input vector to the DNN. The DNN consists of 3 hidden layers of 1024 neurons each, to estimate 43 dimensional phone posteriors at the output. There are 39 phones obtained from CMU pronunciation dictionary* for lexical modeling. The remaining 4 phones are used to model silence and non-speech sounds. The training labels for the DNN are generated using a GMM-HMM based speech recognizer [26]. The recognizer is used to force align the training data to obtain the corresponding phonetic transcription. This whole setup is implemented using the Kaldi toolkit [45].

2) BUT Phone Recognizer: There is no data available for training a phone posterior extractor for the SWS 2013 database. Thus, we use the same phone recognizers as used in [46] to estimate phone posteriors. The phone recognizers were developed at Brno University of Technology (BUT) for three different languages: Czech, Hungarian and Russian [47]. These recognizers were trained using SpeechDAT(E) [48] database which contains 12, 10 and 18 hours of speech for the respective languages. There are 43, 59 and 50 phones for the respective languages. In all cases, 3 additional units were considered to model silence and non-speech sounds.

Once we have calculated the phone posteriors for both databases, we perform speech activity detection (SAD) to remove the noisy frames from test utterances as well as queries. The SAD relies on the output of the phone recognizers to perform this task. It calculates the probability of no voice activity by summing up the probabilities corresponding to

*http://www.speech.cs.cmu.edu/cgi-bin/cmudict

silence and non-speech units in the posterior vector. If for any frame, this probability is highest, the frame is considered noisy and rejected from the corresponding audio. Also, if there are less than 10 frames in an audio file, it is not considered for the experiments to reduce the false alarm rate and computational complexity. Finally, the dimensions corresponding to silence and non-speech units are removed from the posterior vectors as these are unlikely to help in the query matching task [46].

We use these posterior features to perform query detection experiment and obtain a likelihood score for each pair of spoken query and test utterance. Following the score normalization technique used in [46], we normalize the scores to have zero-mean and unit-variance per query. This reduces the variability in scores across different queries and make them comparable for final evaluation. We have also tried sum-to-1 (STO) normalization and keyword-specific thresholds (KST) [49], [50]. However they did not perform better than the mean-variance normalization. Thus the results presented in this work utilize the mean-variance normalization.

D. Evaluation Metric

Several metrics were used to evaluate the performance of different systems. Maximum Term Weighted Value ($MTWV$) [46] is considered as the primary metric which is used to optimize the parameters of different systems. $MTWV$ is the maximum value of Actual Term Weighted Value ($ATWV$) which can be achieved with a well calibrated system. $ATWV$ is a measure based on system hard decision which takes into account the miss and false alarm rate as well as the corresponding costs. It also considers the prior probability of occurrence (P_{target}) of a query in the test utterances which is 4×10^{-3} and 8×10^{-4} for AMI and SWS 2013 respectively. For our experiments, we consider cost of false alarm (C_{fa}) to be 1, cost of missed detection (C_m) to be 100 resulting in the weight factor (β) of 2.49 and 12.49 for AMI and SWS 2013 respectively.

Minimum normalized cross entropy ($minCnxe$) [46] is reported for these systems as a secondary evaluation metric. The normalized cross entropy quantifies the knowledge that a QbE-STD system has on the ground truth. More specifically, it computes the information that is not provided by the scores of a given system. $minCnxe$ is the minimum normalized cross entropy that can be attained by calibrating the system. A perfect system will give $minCnxe \approx 0$, whereas a non-informative system will give $minCnxe = 1$. In addition to these two measures, we also use detection error trade-off (DET) curves to analyze the performance of different systems and compare them for a given range of false alarm probabilities.

To compare the computational efficiency of different approaches, we report the Searching Speed Factor (SSF) [46], which indicates the amount of CPU effort required to search a query in an audio document. Let the duration of a query and a test audio be t_q and t_a units of time, respectively. If an algorithm takes t units of time to search the query, the SSF is defined as: $\frac{t}{t_q \times t_a}$. The total CPU time is reported as if it was computed on a single CPU. If multiple examples are used to search a given query, we use average duration of those examples as the query duration. Lower value of SSF corresponds to a faster system.

E. Test of Statistical Significance

We have also performed Student's t-test to measure statistical significance of the improvements obtained in both MTWV and minCnxe scores by our proposed systems. To perform this test, we compute the scores (MTWV or minCnxe whichever applicable) per query and these scores are considered as samples for a paired-samples t-test. In order to indicate improvement by our systems, the test is one-tailed t-test and the corresponding p-values are indicated with the results.

VIII. Experimental Analysis

We conducted extensive experiments on AMI meeting corpus to analyze the performance of different systems proposed in this paper. The experiments are performed in two challenging scenarios when very few examples (10 examples) or just one query example is provided for QbE-STD, and the test utterances are conversational spontaneous speech with competitive speakers.

A. Baseline System

The DTW based QbE-STD system discussed in Section III is used as a highly competitive baseline system [8]. The performance of this system using evaluation queries is shown in Table II. We observe that the performance with multiple example per query is significantly better than its one example counterpart. This indicates that template averaging is able to incorporate variations from multiple examples of the same query which is similar to the observations presented in [9].

B. Sparse Subspace Detection (SSD)

In this section, we evaluate the sparse subspace detection system presented in Section V. This is a sparsity based system which completely relies on the subspace structure of speech data. The purpose of developing this system is to quantify the contribution of subspace structure of speech for the task of QbE-STD. We follow the steps presented in Algorithm 1 to implement this system. As discussed in Section V, in case of one example per query context appended posterior feature vectors of the example constitute the dictionary. On the other hand, with multiple examples for each query, we construct the dictionary in two ways: (i) concatenation of context appended posteriors and, (ii) learning from the context appended posteriors of different examples of the query. The size of these dictionaries vary depending on the length of query examples, context size and the dictionary construction method being used (concatenation or learning). The number of rows equals the length of context appended posterior feature vector whereas the the number of columns (atoms) depend on the number of frames in the query examples and the dictionary construction method. Dictionary construction is followed by the rest of the steps in Algorithm 1 to find the region of occurrence and likelihood score of the query in the test utterance. Context size

TABLE II
PERFORMANCE OF THE BASELINE SYSTEM AND THREE DIFFERENT SYSTEMS PROPOSED IN THIS WORK. EACH SYSTEM IS EVALUATED USING DIFFERENT NUMBER OF EXAMPLES FOR EACH QUERY. MTWV (HIGHER IS BETTER) AND MINCNXE (LOWER IS BETTER) IS USED AS EVALUATION METRIC. COMPUTATIONAL EFFICENCY IS SHOWN USING SSF (LOWER IS BETTER)

System	1 Example			10 Examples (Concatenation)			10 Examples (Learning)		
	MTWV	minCnxe	SSF	MTWV	minCnxe	SSF	MTWV	minCnxe	SSF
Baseline DTW†	0.4758	0.6526	0.1778	0.6028	0.5014	0.2070	0.6028	0.5014	0.2070
Sparse Subspace Detection	0.3030	0.7874	**0.0367**	0.4117	0.6613	**0.1109**	0.3992	0.6897	**0.0406**
Subspace Regularized DTW	**0.4914****	**0.6376****	0.1889	0.6332***	0.4797**	0.2415	0.6231***	0.4847**	0.2220
Subspace based Re-scoring of DTW	0.4875**	0.6399*	0.1831	**0.6374*****	**0.4610*****	0.2242	**0.6323*****	**0.4674*****	0.2123

†the baseline system uses template averaging in case of 10-examples
* significant at $p < 0.05$; ** significant at $p < 0.001$; *** significant at $p < 0.00001$;

(c), the level of sparsity (λ) and a single threshold parameter used in Kadane's algorithm of this system are optimized using development queries to have the best detection performance. The parameters are optimized for all development queries, and these are not dependent on individual queries. The final results using evaluation queries are presented in Table II. The performance of this system is not as good as the baseline system. However, this is a much faster technique compared to the baseline system as indicated by the lower value of SSF. Also, it shows the subspace structure of speech can be used to perform QbE-STD with reasonable accuracy. The performance degradation can be attributed to the absence of a framework to incorporate temporal information.

C. Subspace Regularized DTW (SR-DTW)

The subspace regularized DTW system is proposed to include the temporal information from the spoken utterance, as discussed in Section VI-A. To evaluate this system, we construct a dictionary for each query as discussed in previous section and follow the steps shown in Algorithm 2 to obtain the likelihood score for a query occurring in a test utterance. The parameters ($Context$, λ and w_d) are optimized using the development queries and the results on evaluation queries are shown in Table II. The optimization is done by varying all the parameters in their respective ranges and maximizing the MTWV. Clearly, this system gives improvement over the baseline system which shows the importance of exploiting the subspace structure of speech while developing a QbE-STD system.

D. Subspace Based Rescoring of DTW (SRS-DTW)

Another approach to take advantage of both DTW and sparsity based system, is to combine their respective scores as discussed in Section VI-B. In this case also, we construct different dictionaries depending on the number of examples provided for each query and perform corresponding detection experiment. We follow the procedure described in Algorithm 3 to obtain the likelihood score for a query occurring in a test utterance. For this system, the $Context$ and λ parameter are optimized by keeping w_s equal to 0. This essentially means that we are trying to obtain the best set of scores using only the sparsity based errors, irrespective of the scores generated by the baseline system. Once the context and λ have been so optimized , we vary w_s in a given range to obtain the best value.

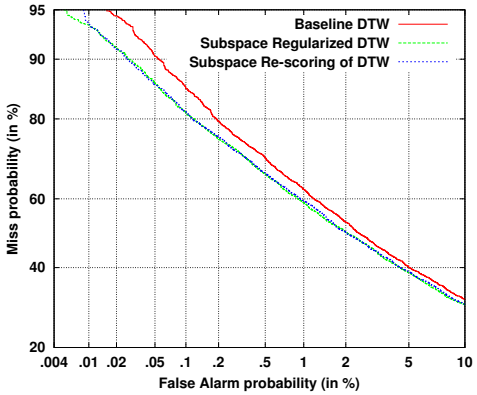

Fig. 7. *DET curves showing the performance of the Sparse-DTW hybrid systems compared to the baseline DTW system using 1 example per query.*

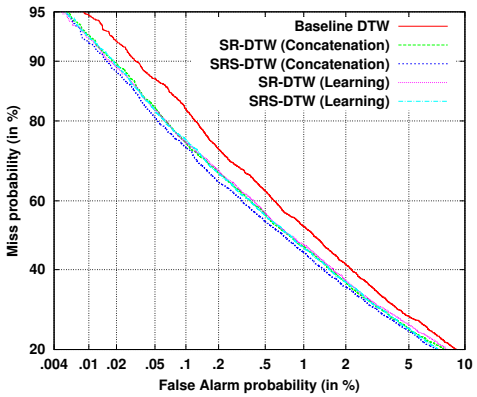

Fig. 8. *DET curves showing the performance of the Sparse-DTW hybrid systems compared to the baseline DTW system using 10 examples per query.*

The resulting performance is summarized in Table II. Similar to the SR-DTW system, this system also gives improvement over the baseline system, once again indicating the importance of subspace structure of speech for the problem at hand. The performance of these systems are also shown using DET curves in Figures 7 and 8 corresponding to 1-example and 10-examples case respectively. The curves show that the performance improvement is consistent over all operating points in the DET curve.

E. Concatenated vs Learned Dictionary

We have performed two sets of experiments for all systems

proposed in this work when multiple examples per query are provided. They differ in the way corresponding query dictionaries are constructed. When we compare the performance in these cases for all three systems, we can see that the performance is worse when the dictionary is learned from the given examples compared to concatenating them to form the dictionary. This indicates that the dictionary learning algorithm has not been able to capture all the information from the query examples provided. However, the performance difference is small, indicating the validity of dictionary learning when concatenating the examples of a query increases the computational cost significantly.

F. Effect of Context and λ

In this section, we discuss the effect of acoustic context (as discussed in Section IV-B) and λ on different systems. The optimal value of these parameters to obtain best MTWV using development queries is presented in Table III. The context size depends on the average query length to capture longer temporal dependency. As we add more examples to generate query templates, the optimal context size increases. On the other hand, the value of λ indicates the desired level of sparsity. It is dependent on the number of atoms present in the dictionary for sparse representation. Bigger dictionaries require higher λ to achieve good reconstruction of the test frames. Thus we require higher λ for the 10-examples case compared to the 1-example case. However, the number of atoms in case of 10-examples (Learning) is higher than 1-example case, but lower than 10-examples (Concatenation) case. This leads to the optimal value of λ for 10-examples (Learning) case being higher than 1-example case but lower than 10-examples (Concatenation) case. We also observe that optimal context size in case of SR-DTW is smaller compared to other two systems. The reason is, in SR-DTW system, we are not only trying to obtain better reconstruction of test frames, but also want to hypothesize the regions representing queries. Higher context produces better reconstruction for smaller regions, effectively reducing the length of the hypothesized regions. Thus the system makes a trade-off between quality of reconstruction and length of the detected region and the optimal context size is smaller than other systems. As an example of this optimization, we present in Figure 9 the variation of MTWV score with respect to context size and different values of λ for 10-examples (Concatenation) case. The scores are generated using SRS-DTW system on development queries while keeping the weight parameter, $w_s = 0$. Clearly, $Context = 7$ and $\lambda = 0.5$ gives the best performance, which is later used to optimize the value of w_s.

G. Effect of Fusion Weight

We have proposed two ways of fusing the baseline DTW and sparsity based system as discussed in Section VI. Parameters w_d and w_s indicate the fusion weights for SR-DTW and SRS-DTW system, respectively. In both cases, $(1 - w)$ represent the contribution of information obtained by relying on the subspace structure of speech. Thus, higher w corresponds to lower contribution from sparsity. We have optimized

TABLE III
OPTIMIZED VALUES OF CONTEXT AND λ GIVING THE HIGHEST MTWV SCORE ON DEVELOPMENT QUERIES FOR DIFFERENT SYSTEMS PROPOSED IN THIS WORK

Systems	1 Example		10 Examples			
			Concatenation		Learning	
	Context	λ	Context	λ	Context	λ
SSD	4	0.01	9	0.6	8	0.1
SR-DTW	1	0.1	2	0.3	2	0.2
SRS-DTW	4	0.01	7	0.5	8	0.1

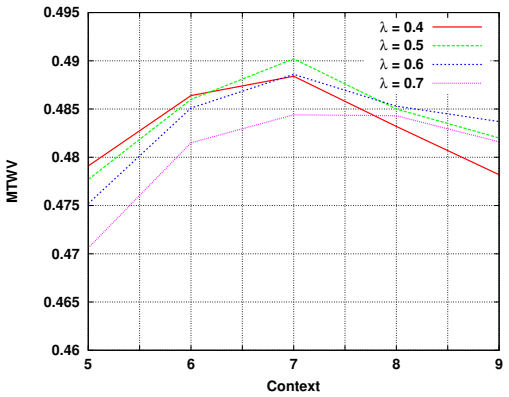

Fig. 9. *Variation of MTWV with changing context for different values of λ. The experiments are performed using SRS-DTW system on development queries with 10 examples (Concatenation) per query, while keeping $w_s = 0$. It corresponds to the scenario when we obtain the scores from sparse reconstruction errors by using boundaries from the baseline system*

these fusion weights on development queries for both systems. As an example, we show the performance variation of SRS-DTW system with corresponding weight, w_s in Table IV, while keeping the other parameters ($Context$ and λ) fixed. We also present the corresponding baseline performance for comparison. We observe that, $w_s = 0.7, 0.8, 0.9$ gives very similar results for 1-example case and $w_s = 0.6, 0.7, 0.8$ for 10-examples (Concatenation) case. This indicates a range of values of w_s to obtain similar results which are better than the baseline. The optimal values of w_s (to obtain best MTWV) are 0.8 and 0.7 corresponding to the cases of 1-example and 10-examples (Concatenation). So, the effective weights for sparsity based scores are 0.2 and 0.3 respectively. It shows that the sparsity based scores provide better discrimination with more examples for each query. This is in conformity with the idea of subspace modeling where many examples are needed for better modeling of a class [40].

H. Computational Efficiency

The computational efficiency of different systems is shown in Table II using SSF metric. It can be observed that sparse subspace detection (SSD) is the most efficient system among all in both cases of using different number of examples. The price for this efficiency is paid by degradation in detection performance. On the other hand, the hybrid approaches need more computation than the baseline system, because in both systems, we perform DTW as in the baseline system while

TABLE IV
VARIATION OF MTWV AND MINCNXE FOR DIFFERENT VALUES OF FUSION WEIGHT (w_s). THE EXPERIMENTS ARE PERFORMED USING SRS-DTW SYSTEM ON DEVELOPMENT QUERIES.

weight (w_s)	1 Example		10 Examples (Concat.)	
	MTWV	minCnxe	MTWV	minCnxe
0.6	0.4638	0.6394	0.6043	**0.4757**
0.7	0.4761	**0.6345**	**0.6051**	0.4765
0.8	**0.4795**	0.6355	0.6015	0.4828
0.9	0.4767	0.6414	0.5890	0.4965
Baseline	0.4646	0.6558	0.5684	0.5192

TABLE V
PERFORMANCE OF THE BASELINE SYSTEM AND SUBSPACE BASED RE-SCORING OF DTW SYSTEM PROPOSED IN THIS WORK. EACH SYSTEM IS EVALUATED FOR THREE DIFFERENT CASES WHERE DIFFERENT NUMBER OF EXAMPLES PER QUERY IS AVAILABLE. MTWV (HIGHER IS BETTER) AND MINCNXE (LOWER IS BETTER) IS USED AS EVALUATION METRIC.

Examples per query	Baseline System		Proposed System	
	MTWV	minCnxe	MTWV	minCnxe
1	0.4287	0.6183	0.4362*	0.6071**
3	0.3007	0.6682	0.3204***	0.6571**
10	0.2740	0.6893	0.3020***	0.6703***

* significant at $p < 0.05$; ** significant at $p < 0.001$; *** significant at $p < 0.00001$;

performing additional computation for obtaining the sparse representation to complete the hybridization. Also, SRS-DTW system is computationally more efficient than the SR-DTW system because in the case of SRS-DTW system, we perform sparse coding only for a sub-sequence (hypothesized region from baseline) of the test utterance, whereas for SR-DTW system, we need the sparse representation for the whole utterance to obtain the regularized distance matrix for DTW. We further observe that, dictionary learning approach is faster than the concatenation of examples of a query. This difference in speed is due to the smaller size of dictionary used for sparse coding when we have learned a dictionary from different examples of a query. Thus in all cases, there is a trade-off between performance enhancement and computational efficiency of the systems and we can choose a system to perform QbE-STD depending on our requirements.

IX. EXPERIMENTS ON SWS 2013

We conducted another set of experiments on SWS 2013 database to show the validity of proposed approach in real life scenarios. As discussed in Section VII-C, we use 3 different BUT phone recognizers to extract the posterior features. In [9], the authors concatenate the feature vectors obtained from different phone recognizers to perform query detection, which was their best individual system. Thus, we implemented it as our baseline system.

Out of the three proposed systems, we use the SRS-DTW system due to its ease of parameter optimization and superior performance compared to other systems. We perform separate experiments for queries with different number examples available per query. In case of multiple examples per query, we concatenate them to construct the corresponding dictionary as it gives better performance compared to dictionary learning experiments on AMI corpus. The parameters of our system are optimized using development queries and the results using evaluation queries are presented in Table V. The difference in performance in three sets of queries can be attributed to the corresponding quality of recordings. Clearly, our system performs better than the baseline system in all three cases. We observe that the performance gain increases with increasing number of examples per query. This is similar to the results obtained on AMI database. To analyze the effect of additional examples per query (for queries with 3 or 10 examples), we conduct another set of experiments where we add one example at a time to each query and obtain the corresponding detection performance. The resulting $MTWV$ values are presented as a function of the number of examples per query in Figure 10. Clearly, the performance improvement is higher with additional examples for SRS-DTW system compared to the baseline. The overall performance gain indicates that the proposed methods are generalizable to real-world scenario and shows the importance of low-dimensional subspace structure of speech for the task of QbE-STD.

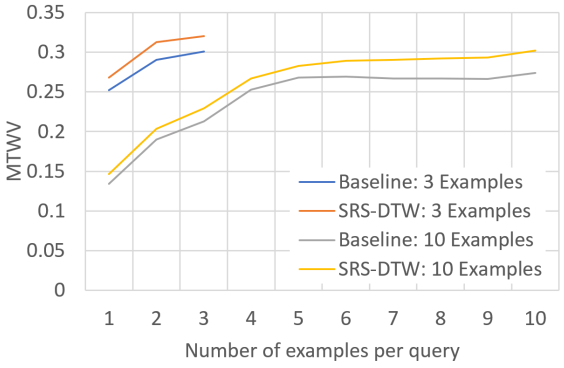

Fig. 10. *Comparison of improvements in $MTWV$ score with additional examples per query for baseline DTW and proposed SRS-DTW system. The performance gain is higher with the proposed system.*

X. CONCLUSION

In this paper, we have proposed three different systems exploiting on the low-dimensional subspace structure of speech. The performance of these systems indicate the usefulness of this structure for QbE-STD. The sparse subspace detection system is shown to be faster than the baseline template matching system with reasonable accuracy. On the other hand, the hybrid systems relying on sparse representation as well as template matching approach yield better performance. The improvement is higher in case of multiple examples per query, which indicates the capability of the proposed approaches to exploit the information from multiple examples better than the baseline system. The performance gain in MediaEval challenge database validates our approach in challenging real-world scenarios.

It has also been shown that the proposed systems benefit from multiple examples of a query. In the future, we plan to obtain these examples from the best scoring hypotheses generated by a QbE-STD system or by user driven feedback. These examples will then be used to learn the query dictionary

to keep the computational cost to an acceptable level. In another approach, the low-dimensional subspace structure of speech can also be used to find repetitive patterns in speech signal, which will help in identifying phone-like units in a data driven manner. A DNN trained using these units predicts new posterior feature vectors which can benefit all the systems proposed in this paper as well as the baseline system.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Swiss NSF project on "Parsimonious Hierarchical Automatic Speech Recognition and Query Detection (PHASER-QUAD)", grant agreement number 200020-169398. We also acknowledge the reviewers for their insightful remarks to improve this manuscript.

REFERENCES

[1] L.-s. Lee and Y.-c. Pan, "Voice-based information retrieval-how far are we from the text-based information retrieval?" in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 26–43.

[2] T. K. Chia, K. C. Sim, H. Li, and H. T. Ng, "A lattice-based approach to query-by-example spoken document retrieval," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2008, pp. 363–370.

[3] A. Mandal, K. P. Kumar, and P. Mitra, "Recent developments in spoken term detection: a survey," *International Journal of Speech Technology*, pp. 1–16, 2013.

[4] W. Shen, C. M. White, and T. J. Hazen, "A comparison of query-by-example methods for spoken term detection," DTIC Document, Tech. Rep., 2009.

[5] K. M. Knill, M. J. Gales, A. Ragni, and S. P. Rath, "Language independent and unsupervised acoustic models for speech recognition and keyword spotting," in *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.

[6] L. Zhang, D. Karakos, W. Hartmann, R. Hsiao, R. Schwartz, and S. Tsakalidis, "Enhancing low resource keyword spotting with automatically retrieved web documents," in *Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

[7] Z. Tuske, D. Nolden, R. Schluter, and H. Ney, "Multilingual mrasta features for low-resource keyword search and speech recognition systems," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7854–7858.

[8] X. Anguera, L. J. Rodriguez-Fuentes, I. Szoke, A. Buzo, F. Metze, and M. Penagarikano, "Query-by-example spoken term detection evaluation on low-resource languages," in *The 4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU'14)*, 2014.

[9] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, "High-performance query-by-example spoken term detection on the SWS 2013 evaluation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7819–7823.

[10] L. R. Rabiner, A. E. Rosenberg, and S. E. Levinson, "Considerations in dynamic time warping algorithms for discrete word recognition," *The Journal of the Acoustical Society of America*, vol. 63, no. S1, pp. S79–S79, 1978.

[11] L. Deng, "Switching dynamic system models for speech articulation and acoustics," in *Mathematical Foundations of Speech and Language Processing*. Springer New York, 2004, pp. 115–133.

[12] S. King, J. Frankel, K. Livescu, E. McDermott, K. Richmond, and M. Wester, "Speech production knowledge in automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 121, no. 2, pp. 723–742, 2007.

[13] D. Ram, A. Asaei, and H. Bourlard, "Subspace detection of dnn posterior probabilities via sparse representation for query by example spoken term detection," in *Seventeenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016.

[14] P. Dighe, G. Luyet, A. Asaei, and H. Bourlard, "Exploiting low-dimensional structures to enhance DNN based acoustic modeling in speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2016.

[15] P. Dighe, A. Asaei, and H. Bourlard, "Low-rank and sparse soft targets to learn better DNN acoustic models," in *Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 2017.

[16] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 2, pp. 210–227, 2009.

[17] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Sparse representation for target detection in hyperspectral imagery," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 3, pp. 629–640, 2011.

[18] T. N. Sainath, B. Ramabhadran, M. Picheny, D. Nahamoo, and D. Kanevsky, "Exemplar-based sparse representation features: From TIMIT to LVCSR," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2598–2613, 2011.

[19] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2067–2080, 2011.

[20] A. Asaei, "Model-based sparse component analysis for multiparty distant speech recognition," Ph.D. dissertation, École Polytechnique Fédéral de Lausanne (EPFL), 2013.

[21] D. Ram, A. Asaei, P. Dighe, and H. Bourlard, "Sparse modeling of posterior exemplars for keyword detection," in *Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

[22] D. Ram, A. Asaei, and H. Bourlard, "Subspace regularized dynamic time warping for spoken query detection," in *Workshop on Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, 2017.

[23] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[24] C.-a. Chan and L.-s. Lee, "Model-based unsupervised spoken term detection with spoken queries," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 7, pp. 1330–1342, 2013.

[25] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2009, pp. 398–403.

[26] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[27] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *IEEE Workshop on Automatic Speech Recognition & Understanding (ASRU)*, 2009, pp. 421–426.

[28] M. Müller, *Information retrieval for music and motion*. Springer, 2007, vol. 2.

[29] Y. Zhang, K. Adl, and J. Glass, "Fast spoken query detection using lower-bound dynamic time warping on graphical processing units," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5173–5176.

[30] X. Anguera, "Information retrieval-based dynamic time warping." in *Fourteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2013.

[31] X. Anguera and M. Ferrarons, "Memory efficient subsequence dtw for query-by-example spoken term detection," in *Multimedia and Expo (ICME), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1–6.

[32] A. Asaei, D. Ram, and H. Bourlard, "Redundant hash addressing for large-scale query by example spoken query detection," Idiap, Tech. Rep., 2016.

[33] C.-y. Lee and J. Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 40–49.

[34] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Acoustic segment modeling with spectral clustering methods," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 2, pp. 264–277, 2015.

[35] X. Anguera, F. Metze, A. Buzo, I. Szoke, and L. J. Rodriguez-Fuentes, "The spoken web search task," in *the MediaEval 2013 Workshop*, 2013.

[36] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.
[37] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.
[38] G. Luyet, P. Dighe, A. Asaei, and H. Bourlard, "Low-rank representation of nearest neighbor phone posterior probabilities to enhance DNN acoustic modeling," in *Seventeenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016.
[39] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.
[40] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research (JMLR)*, vol. 11, pp. 19–60, 2010.
[41] P. Dighe, A. Asaei, and H. Bourlard, "Sparse modeling of neural network posterior probabilities for exemplar-based speech recognition," *Speech Communication*, 2015.
[42] J. Bentley, "Programming pearls: algorithm design techniques," *Communications of the ACM*, vol. 27, no. 9, pp. 865–873, 1984.
[43] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos *et al.*, "The AMI meeting corpus," in *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, vol. 88, 2005.
[44] "AMI corpus partition," http://groups.inf.ed.ac.uk/ami/corpus/datasets.shtml.
[45] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE workshop on automatic speech recognition and understanding (ASRU)*, 2011.
[46] L. J. Rodriguez-Fuentes and M. Penagarikano, "Mediaeval 2013 spoken web search task: system performance measures," *n. TR-2013-1, Department of Electricity and Electronics, University of the Basque Country*, 2013.
[47] P. Schwarz, "Phoneme recognition based on long temporal context," Ph.D. dissertation, Faculty of Information Technology BUT, 2008.
[48] P. Pollák, J. Boudy, K. Choukri, H. Van Den Heuvel, K. Vicsi, A. Virag, R. Siemund, W. Majewski, P. Staroniewicz, H. Tropf *et al.*, "Speechdat (e)-eastern european telephone speech databases," in *the Proc. of XLDB 2000, Workshop on Very Large Telephone Speech Databases*. Citeseer, 2000.
[49] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen *et al.*, "Score normalization and system combination for improved keyword spotting," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 210–215.
[50] Y. Wang and F. Metze, "An in-depth comparison of keyword specific thresholding and sum-to-one score normalization," in *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.

Dhananjay Ram received his Master of Technology (M.Tech) degree in Electrical Engineering from Indian Institute of Technology Kanpur in India. Currently, he is a PhD student at École Polytechnic Fédérale de Lausanne (EPFL) in Electrical Engineering and research assistant at Idiap Research Institute, Switzerland. His PhD thesis focuses on building Query by Example Spoken Term Detection system in zero-resource scenario. His research interests lie in the areas of speech processing, machine learning, deep learning and Bayesian inference.

Afsaneh Asaei received her B.Sc. and M.Sc. (Hons) from Amirkabir and Sharif Universities of Technologies in Electrical and Computer Engineering. She held a research engineer position at Iran Telecommunication Research Center (ITRC) during 2002-2008. She then joined Idiap Research Institute in Martigny and was a Marie Curie fellow of Speech Communication with Adaptive LEarning (SCALE) training network. She received her PhD in 2013 from École Polytechnic Fédérale de Lausanne (EPFL). Her thesis focused on model-based sparsity for reverberant speech processing and its key idea was awarded the IEEE Spoken Language Processing Grant. She has more than 15 years experience on applications of machine learning and signals processing for pattern recognition, voice detection, localization, separation, enhancement, recognition, authentication, coding, assessment and evaluation for neurodegenerative pathology with a focus on robustness and generalization for adversarial applications. Currently, she is the head of artificial intelligence (AI) at the Center of Innovation and Business Creation at Technical University of Munich (UnternehmerTUM). Her research interests lie in the area of machine learning, signal processing, statistics, scene analysis and cognition, and health informatics.

Hervé Bourlard received the Electrical and Computer Science Engineering degree and the PhD degree in Applied Sciences both from "Facult Polytechnique de Mons", Mons, Belgium. After having been a member of the Scientific Staff at the Philips Research Laboratory of Brussels and an R&D Manager at L&H SpeechProducts, he is now Director of the Idiap Research Institute, Full Professor at the Swiss Federal Institute of Technology Lausanne (EPFL), and Founding Director of the Swiss NSF National Centre of Competence in Research on "Interactive Multimodal Information Management (IM2)" (2001-2013). Having spent (since 1988) several long-term and short-term visits (initially as a Guest Scientist) at the International Computer Science Institute (ICSI), Berkeley, CA, he is now an ICSI External Fellow and a member of its Board of Trustees. His main research interests mainly include statistical pattern classification, signal processing, multi-channel processing, artificial neural networks, and applied mathematics, with applications to a wide range of Information and Communication Technologies, including spoken language processing, speech and speaker recognition, language modeling, multimodal interaction, augmented multi-party interaction, and distant group collaborative environments. H. Bourlard is the author/coauthor/editor of 8 books and over 330 reviewed papers (including one IEEE paper award) and book chapters. He is a Fellow of IEEE and ISCA, and a Senior Member and Member of the European Council of ACM. He is (or has been) a member of the program/scientific committees of numerous international conferences (e.g., General Chairman of IEEE Workshop on Neural Networks for Signal Processing 2002, Co-Technical Chairman of IEEE ICASSP 2002, General Chairman of Interspeech 2003) and on the Editorial Board of several journals (e.g., past co-Editor-in-Chief of "Speech Communication"). He is the recipient of several scientific and entrepreneurship awards.