Int J Speech Technol
DOI 10.1007/s10772-013-9217-1

Recent developments in spoken term detection: a survey

Anupam Mandal · K.R. Prasanna Kumar · Pabitra Mitra

Received: 18 July 2013 / Accepted: 21 November 2013

Abstract Spoken term detection (STD) provides an efficient means for content based indexing of speech. However, achieving high detection performance, faster speed, detecting ot-of-vocabulary (OOV) words and performing STD on low resource languages are some of the major research challenges. The paper provides a comprehensive survey of the important approaches in the area of STD and their addressing of the challenges mentioned above. The review provides a classification of these approaches, highlights their advantages and limitations and discusses their context of usage. It also performs an analysis of the various approaches in terms of detection accuracy, storage requirements and execution time. The paper summarizes various tools and speech corpora used in the different approaches. Finally it concludes with future research directions in this area.

Keywords Spoken term detection · LVCSR · Subword-lattice · Query-by-example · Template matching

1 Introduction

With increasing amount of spoken data being stored, shared and processed nowadays, the mechanisms for their indexing and retrieval are also getting increasing attention. These indexing mechanisms are used to organize spoken data records that may be as diverse as cultural and heritage speech, audio lectures in universities, meeting speech, broadcast news, call center conversations, voice intercepts by law enforcement agencies and so-on and so-forth. Typically, speech indexing is performed based on identity of the speaker or spoken content or both. While speaker recognition and speaker diarization are used for indexing speech based on the identity of the speaker, variants of speech recognition techniques such as keyword spotting and STD are employed for content based indexing. Keyword spotting involves finding occurrences of specific spoken words in a speech utterance. STD extends the same by finding a sequence of such words (single word or a phrase) in the speech utterance. However, as far as this survey is concerned, keyword spotting is considered as a part of STD and both of them have been addressed in the survey. The important challenges in the context of STD as are follows:

1. Improvement of the detection performance.
2. Faster search time.
3. Provision of handling unrestricted vocabulary including OOV words.
4. Handling of pronunciation variants.
5. Resource sparseness towards application of statistical techniques in underrepresented languages.

These challenges have been addressed using different approaches as described in the next section.

2 Broad approaches to STD

Spoken term detection is viewed as a variant of the problem of speech recognition. The essential difference between

A. Mandal (✉) · P. Mitra
Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India
e-mail: amandal@cse.iitkgp.ernet.in

P. Mitra
e-mail: pabitra@cse.iitkgp.ernet.in

K.R. Prasanna Kumar
Center for AI & Robotics, Bangalore, India

Published online: 14 December 2013

the two is that the former detects only a pre-defined set of spoken terms while the later gives a complete transcription of the contents of a speech utterance. Nevertheless, many of the approaches meant for speech recognition finds their applicability in STD with modifications. A broad classification of the different approaches used for spoken term detection are as follows:

1. Supervised approaches
   (a) Acoustic keyword spotting based
   (b) LVCSR (Large Vocabulary Continuous Speech Recognition) based
   (c) Subword recognizer based
   (d) Query-by-Example (text based STD)
   (e) Event based
2. Unsupervised approaches: QBE (Query-by-example using template matching)
   (a) Frame based template matching
   (b) Segment based template matching

Much of the earlier research in this field originated in the framework of acoustic keyword spotting (Rose 1996). Section 3 presents a review of the developments in the area of acoustic keyword spotting. Many of the present day STD systems use large vocabulary continuous speech recognition (LVCSR) technology that requires supervised training of HMMs with huge amount of annotated speech and language resources. However, many of the new languages on which STD tasks are to be performed are under-represented in terms of resources required for building statistical models of HMMs used in LVCSRs (Boves et al. 2009). Even for the well represented languages, the LVCSR technology by itself suffers from several limitations with respect to STD tasks. Often the spoken data contains multilingual words which are difficult to model. This is quite common in the context of Indian languages, where English words are often interspersed in predominantly vernacular speech. The English words become OOV words in the vernacular language. Another example of OOV words are the different named entities like names of places and persons, that are generally not covered in a standard pronunciation dictionary of a language. Recognition of such OOV words is an important research challenge in keyword spotting as they are not naturally handled by LVCSRs. Another issue with the LVCSR based approach is that the high word level accuracy of these recognizers is primarily due to the effect of language model (Novotney et al. 2009). This aspect makes these recognizers less effective for STD tasks in domains for which an appropriate language model was not available during training. This issue has been addressed to a great extent by keyword spotters that makes use of phonetic recognizers (James and Young 1994; Thambiratnam and Sridharan 2005; Vergyri et al. 2007; Mamou et al. 2007) instead of word recognizers. These phonetic recognizers are based on subword modeling of HMMs. However, the low accuracy of the phone level outputs makes it difficult to achieve high accuracy for the overall system. In addition to the limitations mentioned above, all these methods require huge amount of transcribed speech and lexicon resources for statistical modeling of HMMs. The details of these supervised approaches involving LVCSRs are discussed in Sect. 4. The same for subword recognizers is discussed in Sect. 5.

Alternative approaches of spoken term detection involve query-by-example (QBE) methods, wherein the keyword or the spoken term is introduced from the speech, either from a speech recording interface or excising it from speech cuts. They have either little or no requirements of annotated speech data or prior knowledge of the underlying language and hence hold considerable promise in the context of low resource languages.

The QBE approaches can be classified into two categories. The first category of QBE approaches are supervised methods that use phone lattice representation of keyword exemplars to be matched against a similar representation of the target utterance. Text-based STD techniques are applied on phone lattices (Shen et al. 2009; Parada et al. 2009) during the process of matching. These methods do not require a phonetic lexicon for the queried keywords but require labeled resources for building lattices. Section 6 presents a review of those approaches. The second category of these approaches are unsupervised methods based on template matching paradigm where the queried keyword template is matched with the target utterance for detecting a possible presence of the same. These approaches do not require the availability of any kind of labeled resources and hence are most suitable for under-represented languages. Section 8 presents a discussion on these methods. The template based methods have two major steps. The first step provides a template representation of the spoken term. This is followed by matching of the template against a similar representation of the target utterance to determine the possible positions of occurrence of the term in the target utterance.

The template based methods are in existence since the early days of speech recognition research (Bridle 1973). They took a back seat with the advent of HMMs. However, they received renewed attention in the context of QBE methods after the STD evaluations conducted by National Institute of Standards and Technology (NIST) in 2006. The focus in this category in recent times has been primarily on novel methods of template representation and improvement in speed of the template matching process. Earlier, the templates were represented with Mel frequency cepstral coefficients (MFCC) features. The more recent approaches have used posterior features (Fousek and Hermansky 2006; Zhang and Glass 2009; Silaghi and Bourlard 1999; Huijbregts et al. 2011) and acoustic segment models (Wang

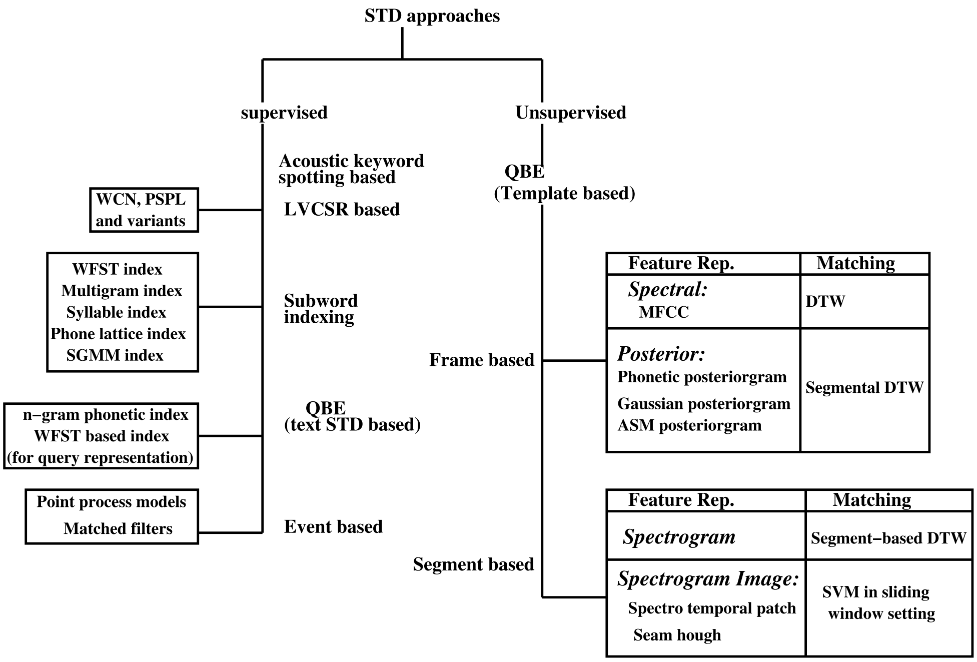

Fig. 1 Taxonomy of approaches to STD

et al. 2011) for template representation. Spectrogram segments (Chan and Lee 2010) and spectrographic image features (Barnwal et al. 2012; Ezzat and Poggio 2008) have also been used in template representation. For template matching, almost all methods use some or the other variant of Dynamic time warping (DTW) (Sakoe and Chiba 1978) (Meyers et al. 1980) that is essentially a dynamic programming based algorithm to find the degree of similarity between two time series differing in length. The DTW based approaches operate at frame level while comparing a keyword template with its counterpart in the target utterance. One such variant of frame based DTW, called segmental-DTW (Zhang and Glass 2009) has been used to improve the accuracy of detection. Alternatively, DTW techniques at segment level (segment based DTW) has been proposed in place of frame based DTW (Chan and Lee 2010) for reducing computation time with slightly lower detection accuracy. Computational efficiency of frame based DTW methods *e.g.,* segmental-DTW, have been improved using a lower bound estimate based on inner product distance in Zhang and Glass (2011). Apart from the methods employing DTW, a method using matched filters have been proposed in Barnwal et al. (2012), Ezzat and Poggio (2008) for template matching. The Sects. 8.2 and 8.3 discusses the various methods of template representation and template matching respectively.

A different paradigm of supervised keyword spotting proposed in Jansen and Niyogi (2009), Kintzley et al. (2011) uses an event based model to represent a keyword. These approaches use a sparse representation of a keyword that makes keyword spotting significantly faster than conventional methods. Event based keywordspotting is discussed in Sect. 7.1. All the methods described so far apply to detection of spoken terms in unencrypted speech. An attempt towards performing the same on encrypted VoIP speech was done by Wright et al. (2010). Their work is described in Sect. 7.2.

The taxonomy of these approaches is shown in Fig. 1.

The paper concludes with a discussion and possible directions of future work towards spoken term detection in Sect. 10.

3 Acoustic keyword spotting

In acoustic keyword spotting, a parallel network of keyword and background filler models are used (Rose and Paul 1990). Here, the model of a keyword is represented by concatenating constituent phoneme models. The filler models are constructed using phoneme loops. Each phoneme is modeled as a HMM/GMM trained using statistical techniques. Neural networks are also used (Szoke et al. 2005) used for modeling phonemes (Szoke et al. 2005). Log-likelihood scores corresponding to the putative hits are obtained using keywords with *filler* and *filler* only passes. Variants of likelihood ratio of the scores are used to declare the putative keyword hits as true hits and false alarms.

The HMM/GMM models are generally trained using generative training techniques involving likelihood maximization. However, it has been argued that the training objective aims at maximizing the likelihood of the transcribed utterances and not that of keyword spotting performance (Grangier et al. 2009). This issue is addressed by discriminative training approaches. These approaches maximizes during training, different criteria that have a direct impact on the keyword spotting performance. The approach in Sukkar et al. (1996) aims at maximizing the likelihood ratio between the keyword and garbage models for keyword utterances and minimizing it over a set of false alarms generated by a first keyword spotter. Sandness and Hetherington (2000) proposed to apply Minimum Classification Error (MCE) to the keyword spotting problem. The approaches in Weintraub et al. (1997) and Benayed et al. (2003) combined different HMM-based keyword spotters. The former used a neural network to combine the likelihood ratios of different models while the later used a support vector machine to combine different averages of phone level likelihoods. Keshet et al. (2007) proposed a training algorithm to directly maximize the Figure-of-Merit criteria typically used to evaluate the performance of keyword spotters. The same is achieved by maximizing the area under the Receiver Operating Characteristics (ROC) curve.

A major limitation of the acoustic framework for keyword spotting is the difficulty encountered in handling new keywords. The system must decode the target utterance with the new keyword list all over again, every time a new keyword is entered. This results in excessively high search times. This limitation is addressed by STD system based on LVCSRs and subword recognizers as described next.

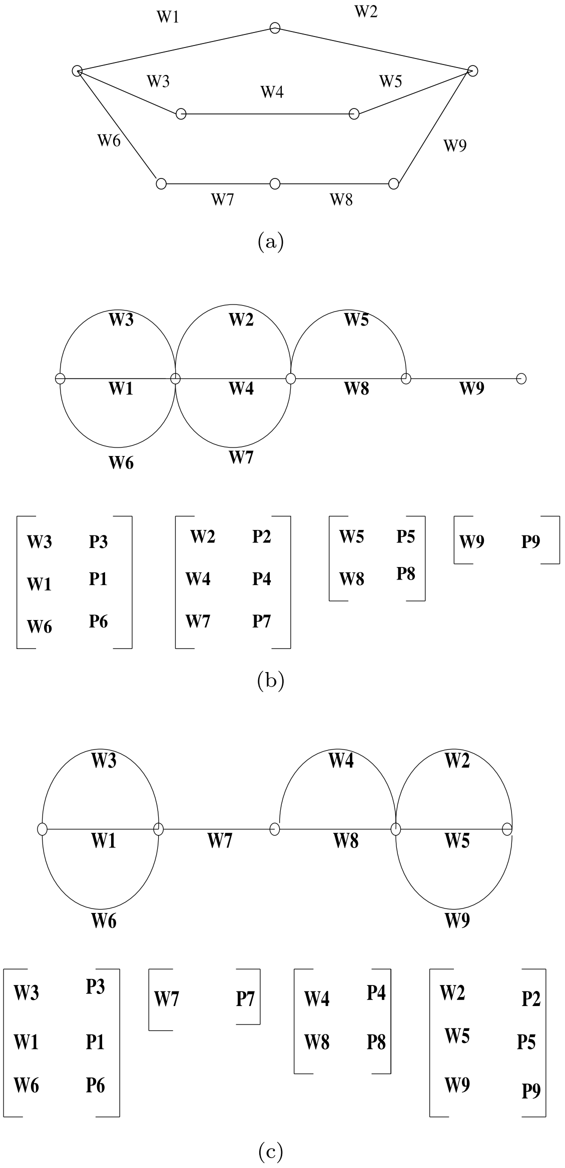

Fig. 2 (**a**) Word lattice with seven words and corresponding (**b**) PSPL (**c**) WCN respectively. W_i's represent keywords and P_i's represent the corresponding posterior probabilities associated with each word

4 STD using LVCSRs

A considerable amount of research efforts on spoken term detection have focussed on extending information retrieval techniques available for text to spoken documents. Some of these are described in Garofolo et al. (2000). A LVCSR system is used to generate word level transcription corresponding to the input speech. These are then indexed using information retrieval techniques available for text. These indices are searched for the presence of query terms. However, often the word level transcription generated by 1-best output of the LVCSR contains errors that affect the performance of the STD system. Hence, word lattices are used for indexing instead of 1-best output of the LVCSR. Word lattices are directed acyclic graphs. Each vertex in a lattice is associated with a timestamp. Each edge (u, v) is labeled with a word or phone hypothesis and its *prior probability*, that is the probability of the signal delimited by the timestamps of the vertices u and v, given the hypothesis. A similar but more compact representation of a word lattice is called a word confusion network (WCN) (Hakkani-Tur and Riccardi 2003; Mangu et al. 2000). Each edge (u, v) labeled with a word hypothesis and its *posterior probability*, that is the probability of the word given the signal. The construction of a WCN is based on word arcs. All word arcs that overlap in time are clustered together irrespective of the positions of these arcs in respective paths. Thus WCNs provide a strict alignment in time of all the words in the lattice. A still compact representation of the word lattice has been proposed in Chelba and Acero (2005) known as *position specific posterior lattice* (PSPL). PSPL gives the posterior probability of a word w at a specific position in a lattice. All paths in the lattice are enumerated, each with its own path weight. The posterior probability of a given word at a given position is computed by summing all path weights that include the given word at a given position as the numerator and then divided by sum of the weights in the lattice. Figure 2 shows a word lattice and its corresponding PSPL and WCN representation.

Word lattices and its variants have been successfully used for improving the detection rate of in-vocabulary (IV) terms. However, they cannot handle OOV or rare word queries when used in word-based fashion. A detailed performance comparison of PSPLs and WCNs have been discussed in Pan and shan Lee (2010). It is observed that the PSPLs always yield a better performance than WCNs but require more space for storing the indices. The use of subword units is also explored to extend the word based PSPL/WCN one step further to subword based PSPLs (S-PSPL) and subword based WCNs (S-WCN). It is found that S-PSPLs/S-WCNs always yield much better mean average precision (MAP) performance for both OOV and IV queries while consuming much less storage space than word-based PSPLs/WCNs. The approaches for STD using other subword units are discussed next.

5 STD using subword recognizers

The approaches in this category involve building indices with different subword units such as phone n-grams, multigrams, syllables, segments or lattice representations of the previously mentioned units (Ng and Zue 2000; Szoke et al. 2008). The indices are built from transcripts of an appropriate subword recognizer using techniques available in text IR community. In Ng and Zue (2000), phonetic transcripts of a spoken document are obtained using a phone recognizer. The phone level transcripts are then used to obtain subword units of varying complexity in terms of their level of detail and sequence length. It is observed that in terms of mean average precision (MAP), the best performance is exhibited by the 3-gram phone index (MAP value 0.86) followed by that of 5-multigrams (MAP value 0.81). It is also found that the overlapping subword units perform better than non-overlapping subword units. Again, among non-overlapping subword units, multigrams perform better than syllable units. For broad phonetic classes, decreasing the number of phonetic classes needs to be compensated by increasing the sequence length for maintaining the same MAP value. Their work also discusses techniques for building indices robust to erroneous transcriptions. It suggests modification of the query to include erroneous variants of the original terms to improve matching of corrupted terms. This can compensate towards substitution errors. Another approach is to provide higher weightage to terms appearing more number of times in the top N-hypotheses. The work proposes an approximate match retrieval metric between query and document defined as opposed to an exact match to accommodate errors in automatic transcription.

The use of multigrams (Deligne and Bimbot 1995) as subword units for dealing with OOV words is explored in Szoke et al. (2008). The impact of multigram parameters namely its length and pruning factor on the size of the index and STD accuracy is studied. The highest detection accuracy is obtained with multigram units of length five. However, the pruning factor does not have much impact on the phone accuracy that saturates around a value of 50 %. Two constrained methods of multigram training are proposed that improved phone accuracy by 9 % and STD accuracy by 7 % relative. It is also found that incorporation of standard n-gram language model on top of multigram units is beneficial, with tri-gram language model performing the best.

The phonetic lattices have been most useful in accommodating high error rates in the transcripts and allowing OOV queries (Mamou et al. 2007; Allauzen et al. 2004; Can et al. 2009; Saraclar and Sproat 2004; Thambiratnam and Sridharan 2005). In Saraclar and Sproat (2004), an improvement in word spotting accuracy (in F-scores) by over five points compared to single-best retrieval is reported for IV and OOV queries by using both phonetic and word lattices. Three different retrieval strategies have been proposed. The first involves combining results after searching the word and the phonetic index. The second suggests searching the word index for IV queries and the phonetic index for the OOV queries. The final strategy is to search the phonetic index only if searching the word index does not return any result. However, the paper does not deal with hybrid queries involving both IV and OOV terms.

The issue of hybrid queries is addressed in Mamou et al. (2007). Their approach uses two indices, a WCN for storing word index and a phonetic index built from phone lattice. For each unit of indexing (both word and phone), the timestamps corresponding to the beginning and end of the unit are stored. During search of an IV query term, a posting list is extracted from the word index. For an OOV query term, the term is converted to a sequence of phones using a joint maximum entropy N-gram model. The posting list of each phone is then extracted from the phonetic index. For a hybrid keyword query involving both IV and OOV terms, word index for IV terms and phonetic index for OOV terms are used. In this case, the posting lists of the IV terms retrieved from the word index is merged with the posting lists of the OOV terms obtained from the phonetic index. The final result of the query is obtained by ANDing or ORing the results of the individual query terms depending on the relation between the terms in the query. This approach outperforms methods based only on word or phonetic index by achieving a precision value of 0.89 and a recall value of 0.83. This system achieved the highest overall ranking for US English speech in NIST 2006 STD evaluation.[1] The performance of the approach (referred as Method 1) on different kinds of speech data as well as its comparative evaluation with that of Saraclar and Sproat (2004) (referred as Method 2) is given in Table 1

The approach in Thambiratnam and Sridharan (2005) claims to decrease the miss rate and increase the search speed for unrestricted vocabulary keyword spotting. It

[1] http://www.itl.nist.gov/iad/mig/tests/std/2006/.

Table 1 Comparison of STD performance on different types of speech for Mamou et al. (2007) (Method 1) and Saraclar and Sproat (2004) (Method 2)

Type of speech	Method 1		Method 2	
	Precision	Recall	Precision	Recall
Broadcast speech	0.94	0.89	0.87	0.85
CTS	0.90	0.81	0.60	0.57
Meeting speech	0.65	0.37	0.5	0.5

makes use of Dynamic Match Phone lattice search (DM-PLS), an extension of Phone lattice search (PLS) that can handle insertion, deletion and substitution errors of a phone recognizer. The approach uses a phone lattice representation of speech using N-best Viterbi recognition pass. The lattice is then searched for the phone sequence constituting the keyword. During search, appropriate cost penalties are imposed for the errors of the phone recognizer following minimum edit distance (MED) principle. The system achieved a miss rate of 10.2 and a false alarm rate of 18.5 that are much lower than conventional HMM based keyword spotting (Rohlicek 1995). However, the search speed is around 300 times real-time. It is to be noted that the scope of searching is restricted to matching of text string representation of keywords against the phone lattice and does not include generation of lattice from speech.

An important aspect related to the use of ASR lattices is to construct the index in an efficient manner so as to minimize the storage and search time requirements. In this context, the ASR lattices are preprocessed into Weighted Finite State Transducers (WFST) and the timing information is pushed onto the output label of each arc in the lattice. The weights are converted into desired posterior probabilities through an additional normalization step (Mohri et al. 1996). Allauzen et al. (2004) describes an algorithm to create a full index represented as a WFST that maps each substring x to the set of indices in the automata in which x appears. The set of substrings are known as factors. The created index represents a factor transducer which is an inverted index of the factors. During search, the query is represented as a weighted acceptor and using a single composition operation with the index, the automata containing the query is retrieved. In Saraclar and Sproat (2004), the factor transducer (FT) maintains a single index entry for all the occurrences of a factor in an utterance and hence is suitable for a spoken utterance retrieval task. A variant of the same index structure named as Timed factor transducer (TFT) that stores timing information on arc weights is proposed in Can (2011) for STD task. The main idea behind TFT is that the timed index is represented by a WFST mapping each factor x to (1) start-end times of the interval where x appears in each automata and (2) the posterior probabilities of x actually occurring in each automaton in the corresponding time interval. The other considerations for FT are retained in TFT. The advantage of this approach is that the search complexity is linear to query length and hence is useful for longer query strings. Also, the structure is highly flexible and supports several other functions in addition to STD. It supports retrieval of any finite state relations from the index, searching complex relations between query words and searching of arbitrary permutations of query words without changing the index.

An approach that has lesser requirements of annotated resources compared to the techniques described above is presented in Garcia and Gish (2006). In this case, a small amount of annotated speech data (15 minutes of word-level transcribed speech) is used to train a self-organizing speech recognizer that defines its own sound units for a domain specific task. The transcriptions are used to train a grapheme-to-sound-unit converter. The input speech is segmented automatically and in an unsupervised manner based on spectral discontinuities. The segments thus obtained are then modeled using segmental Gaussian mixture models (SGMMs). The subset of the speech recordings for which word level transcriptions are available are decoded in terms of SGMM indices. Using the parallel transcriptions, a joint multigram model is used to used to obtain a probabilistic mapping between sequences of letters in the word-level text transcriptions and sequences of SGMM indices. This model is then used to predict the *pronunciation* of a given keyword in terms of the SGMM units, thereby eliminating the need for a pronunciation dictionary. Finally, a dynamic programming search, which minimizes the string edit distance between the predicted pronunciation of a keyword and the automatic transcription obtained, is used to find putative occurrences of the keyword. The average Figure-of-Merit (FOM) is 0.34 using 15 minutes of transcribed training set data.

On similar lines, an approach towards improving the performance of a LVCSR using unsupervised techniques is described in Novotney et al. (2009). The system uses unsupervised techniques for improving acoustic and language model training. The resulting acoustic model recovers 50 % of the gain compared to its supervised trained counterpart. The language model training involving multiplying the word confidences together could achieve a 2 % reduction in Word error rate (WER) over baseline and 0.5 % absolute over unweighted transcripts.

However, in all these approaches the query input is in the form of text. Hence, these approaches assume the availability of phonetic expansion of the keywords either using grapheme-to-phoneme rules or by some other means. This limitation is overcome in QBE paradigm, wherein the keyword or the spoken term is introduced from the speech, either from a speech recording interface or excising it from speech cuts. Non-usage of a phonetic dictionary also helps to address the issues of pronunciation variants and OOV words in the query. The two categories of QBE techniques for STD are described next.

6 QBE approaches using text based STD techniques

The QBE approaches falling in this category uses text based STD techniques on phonetic lattices. The lattices are generated using methods described in the previous section. In Shen et al. (2009), the approach is based on matching lattices (query lattice against utterance lattice indices) using different alignment models that include direct index matching, Viterbi string edit distance alignment and HMM alignment. The query term is converted into a lattice and compared against a phonetic index. The phonetic index is essentially a pruned confusion network that is represented using a compact phone n-gram index.

In direct index matching, the query and the index do not require alignment during the matching procedure. At query time, the query lattice is converted into a 2-gram index and is compared against the phonetic index in sliding windows over all utterances. The scores are obtained for each sliding window which are used to decide the presence of the spoken term in that particular window. The string-edit distance explicitly models the insertions and deletions of index columns. In case of HMM based alignment, the query is treated as a discrete HMM in which each column of the n-gram index is interpreted as a HMM state and unigram probabilities as observation probabilities respectively. Epsilon arcs in the index structure are interpreted as skip arc probabilities. A fixed stay probability parameter allows for insertions in indexed utterances, given the query as a model. The best achieved result under this approach is shown by Discrete HMM alignment that has a precision value of 0.77 on conversational telephone speech (Fisher database). However, the performance varies with the length of the query with very short queries having the lowest detection performance. In terms of computational cost, the system is very fast as n-grams can be accessed in constant time in case of direct a match. The implementation of discrete HMM also allows alignments to be done in 270 % faster than real time for alignment based matching.

In Parada et al. (2009), the QBE framework is built upon a WFST based search and indexing system (Saraclar and Sproat 2004; Allauzen et al. 2004) that allows providing lattice representation of the audio sample directly as a query to the search system. The system employs two passes. The lattice indices matching the query are identified in the first pass. The relevant lattices are loaded and time marks corresponding to the query are extracted in the second pass. However, it is possible to implement the same in a single pass by modifying the index. Different representations and generation mechanisms for both queries and indices built with word and combined word and subword units are studied. It is found that the phone indices built from combination of word and subword units are better than those consisting of only words. Also, the queries represented using samples from the index (lattice cuts) yield better STD performance compared to the queries spoken in isolation. This approach exhibits better STD performance in terms of Actual term weighted value (0.479 vs 0.325) compared to those using reference pronunciations for OOV words.

7 Other supervised approaches of spoken term detection

All supervised approaches of STD mentioned so far makes use of a speech recognizer in some form. A different approach of supervised STD is proposed in event based keyword spotting and for performing STD on encrypted speech that does not use a conventional speech recognizer. However, they require availability of phonetic pronunciations of the keywords.

7.1 Event based keyword spotting

Event based keyword spotting is motivated by the fact that a keyword can be characterized by a set of phonetic events and a faster processing can be achieved by minimizing the set of phonetic events used to represent a keyword. One such event based KWS system using Poisson Process Models is proposed in Jansen and Niyogi (2009). The input speech to the system is represented by a sparse set of phonetic events. The sequence and the relative timing between the events that constitutes a particular keyword are used for modeling the keyword. Given a keyword w and a set of observed phonetic events $O(t)$ beginning at time t, the output is the detection function $d_w(t)$ given by

$$d_w(t) = \log\left[\frac{P(O(t)|\theta_w)}{P(O(t)|\theta_{bg})}\right] \tag{1}$$

where θ_w and θ_{bg} correspond to keyword-specific model parameters and background model parameters respectively. For each phone p in the set of all phones $\mathcal{P}$, $N_p = t_1, \ldots, t_{n_p}$ is defined as the set of points in time at which phone p occurs relative to time t. The observation $O(t) = \{N_p\}_{p\in\mathcal{P}}$ is collection of these set of points. These points are modeled with the assumption that they have arisen from underlying Poisson processes, the background model being homogeneous and the keyword model being in-homogeneous. The detection function is a log likelihood ratio evaluated at t that takes large values at possible occurrences of keywords. This is done by setting a threshold δ_w on $d_w(t)$ which can be determined from the development data. The system operates with a performance comparable to that of HMM based system while using a far more sparser representation of keywords. Another event based keyword spotting system is presented in Lehtonen et al. (2005) that performs data reduction through phonetic matched filters. This filter reduces a phone posteriorgram to a sequence of phonetic events for the task of detecting digits.

The approach in Jansen and Niyogi (2009) and Lehtonen et al. (2005) are combined in Kintzley et al. (2011) using phone-specific filters to derive a reduced set of phonetic events for an event-based keyword spotting. The system achieves a reduction of the event set by 40 % while improving the average word spotting performance by 23 %.

7.2 STD on encrypted speech

The methods discussed so far apply to detecting spoken keywords in unencrypted speech. The complexity of the problem gets increased by several orders of magnitude when the same needs to be performed on encrypted speech. An approach towards identifying specific spoken terms in encrypted VoIP speech (traffic) was studied by Wright et al. (2010). They concluded that it is possible to identify such spoken terms when the audio is encoded using variable bit rate codecs. A variable rate codec encodes phonemes constituting the spoken term with different bit rates, thereby generating packets of different size after encoding. The sequence of the packet sizes corresponding to a spoken keyword provides its signature. A hidden Markov Model trained with the knowledge of phonetic pronunciations of the spoken terms and corresponding size of the packet sequences is used to search instances of the specified term.

All the supervised methods discussed so far require the availability for huge amount of speech and language resources for statistical training. This may be a limitation for under-represented languages which neither has adequate speech nor language resources for building recognizers using statistical methods. This kind of resource sparseness is addressed by the class of unsupervised QBE approaches using template matching as described next.

8 QBE approaches using template matching

The principle of template matching has been used for keywordspotting since early days of speech recognition research. The first work was reported in Bridle (1973). Initially, this framework was used for matching isolated words. Later, the same was extended to detect keywords in a continuous utterance in a sliding window setting. In template matching, several examples of the spoken term to be detected are provided in the form of spoken queries. A template of the spoken term is created from the provided examples by deriving appropriate features. Though the most commonly used features are Mel Frequency Cepstral coefficients (MFCC's), other novel features for template representation are discussed in Sect. 8.2. The query template is slided across the length of the target utterance in overlapping windows. A similarity measure is computed between the query template and its counterpart in the target utterance for every window of observation. The similarity measure is computed using Dynamic Time Warping (DTW) (Sakoe and Chiba 1978), a technique based on dynamic programming to measure the degree of match between two different sized vector sequences.

For the purpose of DTW, the optimal alignment path $\hat{\phi}$, also known as *warp path* between the two sequences is computed and the distortion between the two along the warp path is used for comparison. The measure of distortion is defined by a local distance function between two vectors. The distortion corresponding to the warped path (also called DTW distance) gives the minimum overall costs between the keyword template and its counterpart in the target utterance. A threshold on the DTW distance is set to indicate the extent of match between the two thereby indicating the presence of the keyword. In recent works, variants of DTW, such as segmental DTW and segment-based DTW have been proposed that are presented in Sect. 8.3.

The important considerations in template matching framework are template selection, representation and matching that are discussed in the following sub-sections.

8.1 Selection of query templates

An important issue in QBE approaches using template matching setting pertains to selection of the optimal example to be used as the query template from a set of given exemplars. The following three criteria have been proposed in Tejedor et al. (2010) to select the optimal query.

1. *Dot product based example selection*: The phonetic posteriorgram is computed for each query example. The one that has higher average probabilities for each frame is considered to be the optimal example. This is obtained by computing the sum of the *self* dot product of each frame vector of the posteriorgram. An individual value $c_{DP}(E)$ for an example E that has N frames is given by

$$c_{DP}(E) = \frac{-\log(\sum_N \sum_{i=1}^{P} \mathbf{q_i} \cdot \mathbf{q_i})}{N} \tag{2}$$

where $\mathbf{q_i}$ represents a frame vector containing P phone state posteriors for the ith frame of the query example.
2. *Cross entropy*: The second criterion is based on the average cross-entropy of individual frames of the phonetic posteriorgram of an example. The example having the highest cross-entropy is judged as the optimal one. The

cross entropy based value $c_{CE}(E)$ for an example E having N frames is given by

$$c_{CE}(E) = \frac{-\sum_N \sum_{i=1}^{P} \mathbf{q_i} \log(\mathbf{q_i})}{N} \quad (3)$$

3. *DTW based example selection*: The previous two criteria are based on the evaluation of an individual example itself. If a mistake occurs during selection of any of the examples in exemplar pool, there is a possibility that the wrong example is judged as the optimal one. This problem is overcome with the DTW based example selection in which the DTW distance on phonetic posteriorgram of an example is computed with the rest using cosine distance function. The one having the least average distance (best average similarity) is considered to be the optimal (best) query example.

Apart from choosing the best example, several individual examples can be combined to produce an average representative that gives a better performance. Two such query examples are combined using the following steps: (1) The examples are ordered using the previously defined metrics. (2) DTW based match is performed between the best and the worst query examples. (3) The phone state posteriors of the best example is updated with that of the worst example according to the best path derived from the DTW search. In case of more than two examples, combination starts with two least optimal examples that are combined into a temporal one. The temporal example is further combined with the third least optimal example and this process continues in a *tree based* manner. The final length of the combined example is of the same length as that of the best example. Subsequently, STD is performed using DTW match between the optimal query example and the target utterance. Both dot product and cosine distance are used as the local distance metric in the DTW search. It is reported that DTW cosine distance-based example selection significantly outperforms random query selection when cosine distance is also used as a local distance metric in subsequent DTW search.

8.2 Representation of query templates

In a template based setting, apart from using the MFCC features directly for representing the templates, several novel feature representations have been proposed in recent years. One class of representation uses posteriorgrams derived from different units of speech. The other class of representation creates template using features derived from spectrogram image of the query speech template.

8.2.1 Posteriorgram based template representation

Posteriorgram is a time-vs-class matrix representing the posterior probability of each class for a specific time frame. To state formally, for a speech template having n frames (feature vectors), $\mathbf{O} = (\mathbf{o_1}, \mathbf{o_2}, \ldots, \mathbf{o_n})$, the corresponding posteriorgram is defined as

$$PG(O) = (q_1, q_2, \ldots, q_n) \quad (4)$$

Each vector q_i can be calculated by

$$q_i = \big(P(C_1|s_i), P(C_2|s_i), \ldots, P(C_m|s_i)\big) \quad (5)$$

where C_i represents the ith class and m denotes the number of classes. The class can be the set of phonetic units, Gaussian components or acoustic segments, depending on which the posteriorgrams are referred to a phonetic (Hazen et al. 2009; Fousek and Hermansky 2006), Gaussian (Zhang and Glass 2009) or acoustic segment (Wang et al. 2011) posteriorgrams respectively.

While phonetic posteriorgrams represent the acoustic likelihood scores for each phonetic class at each time frame, Gaussian posteriogram represents the posterior probabilities of a set of Gaussian components corresponding to a speech frame. The phone posterior probabilities in case of a phonetic posteriorgram is obtained from the lattice outputs of a phone recognizer (Shen et al. 2009) or by using a well trained Multi Layer perceptron (MLP) (Fousek and Hermansky 2006). For generating Gaussian posteriorgram, a GMM is trained after removal of silence from the training data. The GMM is used to obtain raw posterior values corresponding to a speech segment. A threshold on the raw probability values is set to zero out very low probability values. The posteriorgram vector is re-normalized to set the summation of each dimension to one. Next, a discounting based smoothing strategy is applied to move a small portion of probability mass from non-zero to zero dimension while obtaining the final posteriorgram vector.

Acoustic segment model (ASM) posteriograms (Wang et al. 2011) are derived from acoustic segment models, that are a set of HMMs obtained in an unsupervised manner without transcription information. The ASMs are obtained using an unsupervised iterative training procedure that consists of initial segmentation, segment labeling and subsequent HMM training. During segmentation, similar neighbouring frames are grouped into small segments based on minimization of local distortion. The segments are labeled using GMM tokenization. The assigned label is the index of the highest scoring component when a segment is matched against a GMM trained using the training data. With the initialized segment labels, a HMM model is constructed and trained iteratively for each segment. These trained models, also known as ASMs are used to compute posteriorgrams as per Eq. (5).

A variation of the segment based approach is described in Chan and Lee (2011) that uses a sequence of spectrogram segments to construct the template instead of building the segment models. The segmentation is performed in an unsupervised manner using hierarchical agglomerative clustering (HAC) on the vectors representing the spectrogram. Similar approaches have been used by Baghai-Ravary et al. (2009) for identification of phoneme clusters using data driven techniques.

8.2.2 Spectrogram image based template representation

The second approach for representing keyword templates make use of information derived from the spectrogram image of the corresponding spoken query. The word spotting architecture in Ezzat and Poggio (2008) uses a set of ordered spectro temporal patches extracted from exemplar spectrograms of keywords at random locations in frequency and time. The extracted patches are organized into a dictionary with two parameters namely their center locations in frequency $\{f\}_{k=1}^{K}$ and their center location in relative time $\{rt\}_{k=1}^{K}$, K being the number of patches in the dictionary. The kth extracted patch is represented as $P_k(f,t)$. Each patch may be viewed as a matched filter that is selective for other similar patterns. Given a target spectrogram $S(f,t)$ of duration T, the patch dictionary is applied to compute the patch dictionary response $\{R_k\}$ as follows: each patch P_k in the dictionary is placed at location $(f_k, rt_k * T)$ and the L_2 norm is computed between the patch and the underlying portion of the spectrogram. The final feature vector representation generated from the spectro-temporal response R_k is of fixed dimension k that is independent of the length T of the spectrogram.

Another recent approach that makes use of spectrogram image to represent keyword templates is given in Barnwal et al. (2012). The approach captures patterns of high energy tracks or *seams* across frequency in spectrograms that carry time invariant signatures of underlying sounds. The *seams* are computed using a seam carving algorithm (Avidan and Shamir 2007) based on a energy function. On a spectrogram, each pixel bin corresponds to a time frequency bin and represents an energy value. The energy function for seam computation maximizes the energy of each bin along the seam. In the next step, Hough transform is used to capture characteristics of the ensemble of *seams* that has been detected in the images of all exemplars of a particular keyword. This results in a *seam-hough* feature vector that is used to represent keyword templates.

Once the template is constructed using any one of the template representation methods, the next step is to match the query template with the target utterance. The methods for matching query templates with the target utterance is given below.

8.3 Matching of query templates

Template matching techniques are used for matching the query templates against audio segments in a test utterance in order to detect a possible existence of the spoken term. DTW techniques and its variants have been most widely used in this regard. Originally this technique was used for aligning examples of isolated words with reference keyword templates. Later it was extended to detect keywords in a continuous utterance wherein matching is done with segments of speech in a sliding window setting. The following gives a formal description of DTW.

Let the frame level representation of the query be

$$\mathcal{X} = x_1, x_2, \ldots, x_{N_x} \tag{6}$$

and that of the utterance segment be

$$\mathcal{Y} = y_1, y_2, \ldots, y_{N_y} \tag{7}$$

A warp path ϕ is an alignment that maps $\mathcal{X}$ to $\mathcal{Y}$ while obeying several constraints. The warping relation is written as a sequence of ordered pairs

$$\phi = (i_k, j_k), \quad k = 1, 2, \ldots, T \tag{8}$$

that represents the mapping

$$\begin{array}{lcl} x_{i_1} & \leftrightarrow & y_{j_1} \\ x_{i_2} & \leftrightarrow & y_{j_2} \\ & . & \\ & . & \\ & . & \\ x_{i_T} & & y_{j_T} \end{array}$$

In case of global alignment, ϕ ties the endpoints of both the utterances *e.g.*, $(i_1, j_1) = (1, 1)$ and $(i_T, j_T) = (N_x, N_Y)$. During alignment, the monotonicity and continuity constraints are enforced. The monotonicity condition ensures that the aligned sequences must retain their original ordering and move forward in time. The continuity condition ensures that no intermediate frames are skipped along the warp path. Thus the two utterances can be compared based on accumulated distortion between aligned frames for a given a warp path as:

$$D_\phi(\mathcal{X}, \mathcal{Y}) = \sum_{k=1}^{T} d(x_{i_k}, y_{j_k}) \tag{9}$$

The optimal warping path $\hat{\phi}$ is given by the warping path that minimizes the accumulated distortion *i.e.,*

$$\hat{\phi} = \underset{\phi}{\arg\min} \quad D_\phi(\mathcal{X}, \mathcal{Y}) \tag{10}$$

The optimal path can then be found using dynamic programming techniques. Figure 3 shows an example of matching two sequences $\mathcal{X}$ and $\mathcal{Y}$ using DTW.

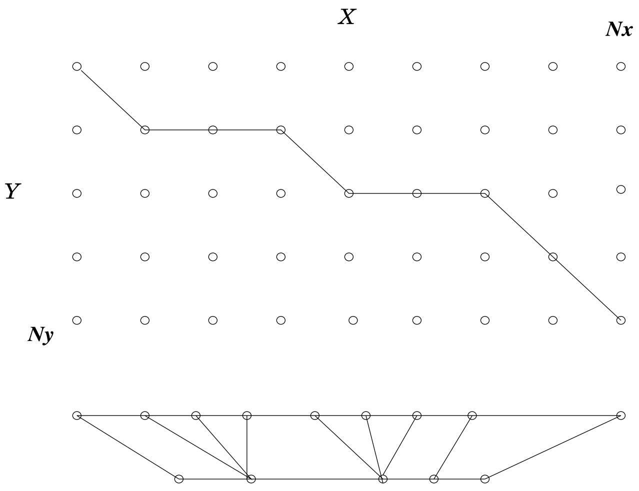

Fig. 3 An example of warped path aligning sequences $\mathcal{X}$ and $\mathcal{Y}$ of length N_x and N_y respectively. The warp path ϕ in this case is the sequence of ordered pairs $(1,1)(2,2)(3,2)(4,2)(5,3)(6,3)(7,3)(8,4)(9,5)$. The alignment corresponding to the warp path is displayed in the lower path of the figure. Adapted from Park and Glass (2008)

Over years various modifications to the original DTW algorithm were made. Variations were made in the constraints imposed and the distance metrics used for computing local distance. Such a modified DTW technique was applied on phonetic posteriorgram in Hazen et al. (2009). Another variation of DTW called Segmental DTW (Zhang and Glass 2009) overcomes the limitation of isolated word templates of the reference query. The other is Segment-based DTW (Wang et al. 2011) that is used for matching utterances represented as sequences of segments instead of frames. These techniques are discussed below:

8.3.1 Modified DTW on phonetic posteriorgram

The comparison of the query and the test segments in the context of phonetic posteriorgram representation in Shen et al. (2009) is translated to finding the similarity measure between two posterior distributions. The feature vectors of a posteriorgram correspond to the posterior distributions. The similarity measure between two posterior distributions q and x is computed based on the principle that the distributions resulting from the same underlying phonetic event should exhibit strong similarity. Such a similarity measure $D(q,x)$ is represented by the dot product of corresponding feature vectors $\mathbf{q}$ and $\mathbf{x}$ in log probability space as:

$$D(q,x) = -\log(\mathbf{q}\cdot\mathbf{x}) \tag{11}$$

To compare the posteriorgrams, the similarity measure between N frames of the query posteriorgram and M frames of the test posteriorgram is computed. This results in a $N \times M$ similarity matrix corresponding to the two posteriorgrams compared.

A modified DTW search is employed to find the region of time in the above matrix having high similarity values between the query segment and the target utterance. The DTW search incorporates the following modifications:

1. It accumulates similarity scores along path extensions in search space.
2. The search disallows simultaneous multi-frame path extensions in both query and test segments.
3. Path extensions with similar duration are favoured by scaling the similarity along individual extensions of a hypothesized path by an alignment slope factor. The alignment slope factor is exponentially weighted by a factor ϕ designed to control the strength of alignment slope constraints.

This approach also allows the use of multiple query examples during search. The scores are generated individually by matching the target utterance with one query example at a time. These scores are then combined to arrive at the total score for an input utterance as follows:

$$S(X|Q_1,\dots,Q_{N_Q}) = \frac{1}{\alpha}\log\frac{1}{N_Q}\sum_{i}^{N_Q}\exp\bigl(-\alpha S(X|Q_i)\bigr) \tag{12}$$

where $S(X|Q_i)$ is the score of the ith query, α is the weighting factor and N_Q is the number of queries. The results show that performance improves when multiple queries are used. The system also has a provision to receive feedback from user, thereby confirming potential hits and re-evaluating the scores for newly observed positive examples. This user-relevance feedback improves precision and significantly reduces the EER of the system.

8.3.2 Segmental DTW on Gaussian posteriorgram

Segmental DTW as described in Zhang and Glass (2009) is a variation of DTW for comparing Gaussian posteriorgrams of query and test utterances. In Segmental-DTW two important modifications are made. (i) Global constraints are imposed to restrict the shape of the warp path. (ii) Multiple alignment path for the same two input sequences are generated from different starting and ending points. The global constraints are meant to prevent an overly large temporal skew between two sequences. To state formally, for a warp path originating at (i,j), the kth coordinate of the path $\mathcal{P} = (i_k, j_k)$ must satisfy

$$\bigl|(i_k - i_1) - (j_k - j_1)\bigr\| \le R \tag{13}$$

This restricts the warp path to a diagonal of width $2R+1$. Additionally, it also generates multiple alignment paths based on different starting points. For utterances of

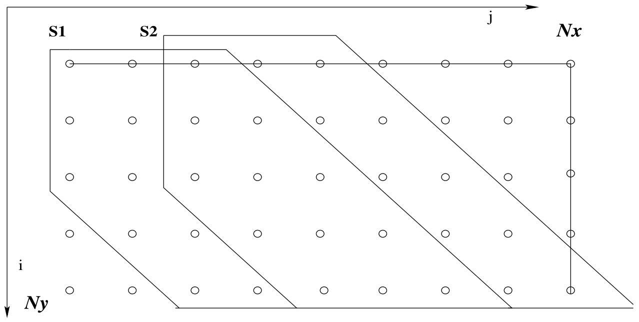

Fig. 4 Alignment in Segmental-DTW with band constraint $R = 2$. The first two start coordinates of the warping path is given by $s1$ and $s2$ respectively. Adapted from Zhang and Glass (2009)

length N_x and N_y with constraint parameter R, the start coordinates will be

$$\big((2R+1)k+1, 1\big), \quad 0 \le k \le \left\lfloor \frac{N_x - 1}{2R+1} \right\rfloor$$

$$\big(1, (2R+1)k+1\big), \quad 0 \le k \le \left\lfloor \frac{N_y - 1}{2R+1} \right\rfloor$$

Based on these coordinates, the total number of diagonal regions will be

$$N_R = \left\lfloor \frac{N_x - 1}{2R+1} \right\rfloor + \frac{N_y - 1}{2R+1} \rfloor + 1 \tag{14}$$

During the selection of start coordinates, a constraint is imposed on the step length of the start coordinates. By applying different start coordinates, the matrix can be divided into several contiguous diagonal regions with width $2R + 1$. Figure 4 shows a snapshot of the diagonal regions formed following band constraint $R = 2$.

The distortion score for each warping path is also computed. The warping region with the minimum distortion score is chosen as the candidate region of the keyword occurrence for that utterance. This approach also supports multiple keyword examples for each query as in Hazen et al. (2009). However, unlike the direct merging method used in Shen et al. (2009), the reliability of each warping region on the test utterance is considered for during merging of scores. The reliability of a warping region is described as follows: Given multiple keyword samples for the query and a test utterance, a reliable warping region on the test utterance is the region where most of the keyword samples aligned. Regions having two or more samples of keyword aligned are only considered for merging of scores. The final distortion score for region is

$$S(r_j) = -\frac{1}{\alpha} \log \frac{1}{k} \sum_{i=1}^{k} \exp\big(-\alpha S(s_i)\big) \tag{15}$$

where r_j denotes jth region, s_i denotes ith sample and k is the no of keyword samples aligned to region r_j. The variable α changes the averaging function from a geometric mean to an arithmetic mean by varying between 0 and 1. The region with the smallest average distortion score may contain the keyword.

Table 2 STD performance comparison using Segmental-DTW on different types of posteriorgram representations

System	P@10	P@N	EER/MAP
PP + SDTW	63.3	52.8	16.8 (EER)
GP + SDTW	61.3	33.0	10.4 (EER)
ASM + SDTW	56.6	40.6	40.4 (MAP)
ASM + GMM + SDTW	59.2	42.1	43.1 (MAP)

In Wang et al. (2011), the segmental DTW is applied on acoustic segment posteriorgrams. A comparison of the STD systems employing segmental-DTW on GMM and ASM posteriorgrams with different local distance measures is presented. The distance measures considered are inner-product, cosine and Bhattacharya distance. A system employing score level linear fusion of the above two systems with equal weights is also presented. Among the different distance measures used for computing the local distortion during DTW search, Bhattacharya distance consistently exhibits the best performance for ASM based system. It gives performance comparable to that of the inner-product distance in case of GMM based system.

It is seen that the ASM system outperforms the GMM system in all the above evaluation metrics. The performance also improves as the number of query examples increases.

The performance comparison between phonetic posteriorgram with modified DTW (PP + SDTW), Gaussian posteriorgram with Segmental-DTW (GP + SDTW), ASM posteriorgram with Segmental DTW (ASM + SDTW) and fusion system (ASM + GMM + SDTW) is given in Table 2.

Though these frame based DTW techniques on different types of posteriorgrams are effective in an unsupervised setting, one major limitation is their high computational overhead. Segment-based DTW, discussed next addresses the issues of computational overhead but at the cost of reduced detection performance.

8.3.3 Segment-based DTW

In segment-based DTW (Chan and Lee 2010), a supersegment in the query is matched with a supersegment in the test utterance. A supersegment is represented as a sequence of several contiguous basic segments. Let, the query Q be represented as $(q_1, q_2, \ldots, q_{N_Q})$ where q_i's are basic segments constituting the query. Similarly, for a test utterance $T = (t_1, t_2, \ldots, t_{N_t})$, t_i denotes the basic segment. Therefore, the supersegment $Q(i_{k-1} + 1, i_k)$, consists of basic segments $(q_{i_{k-1}+1}, q_{i_{k-1}+2}, \ldots, q_{i_k})$ that starts from the first frame of $q_{i_{k-1}+1}$ and ends at the last frame of q_{i_k}. This is matched with $T(j_{k-1}+1, j_k)$ or $(t_{j_{k-1}+1}, t_{j_{k-1}+2}, \dots, t_{j_k})$. During matching, the duration ratio between the two supersegments to be matched is also constrained. Under these conditions, the distance between Q and T is given by

$$D(Q,Y) = \sum_{k=1}^{K} d\big(Q(i_{k-1}+1, i_k), T(j_{k-1}+1, j_k)\big) \tag{16}$$

where K is the number of matched index pair sequence on the warped path. Here, the indices refer to the indices of the basic segments constituting the query and the test utterance. In the next step, a fixed number of vectors M is used to represent each supersegment. This is done by splitting the basic segment if the number of basic segments constituting the basic segments is less than M. Alternatively, the basic segments are merged using a HAC scheme if the number of basic segments is greater than M. In both cases, the number of final segments is equal to M. This results in two sequences of M subsegments. By averaging the feature vectors in each of these M subsegments, we get M vector representations of the supersegment denoted by $(q'_1, \dots, q'_M)$ and $(t'_1, \dots, t'_M)$ respectively. Let, $\mathbf{v}(q'_i)$ be the mean vector of q'_i, $f(q'_i)$ the number of frames in q'_i and $p(q'_i) = \frac{f(q'_i)}{f(q'_1)+\cdots+f(q'_M)}$. Then

$$d\big(Q(i_{k-1}+1, i_k), T(j_{k-1}+1, j_k)\big)$$
$$= f\big(Q(i_k+1, i_{k+1})\big) \sum_{m=1}^{M} e^{\alpha|p(q'_m)-p(t'_m)|} \big\|\mathbf{v}\big(q'_m - \mathbf{v}\big(t'_m\big)\big)\big\| \tag{17}$$

where the first term $f(Q(i_k+1, i_{k+1}))$ makes the distance proportional to the length of the query segment, and the exponential term penalizes composition differences of these two super segments with a control variable α.

It is observed that the segment-based DTW has a minimum 4.1 % lower detection performance in terms of MAP compared to frame based DTW. However, it achieves a 65.4 percent reduction of CPU time compared to the later.

A more desirable approach will reduce both the computation load as well as preserve the detection performance at the same time. It has been observed that when inner-product distance between posteriorgram vectors is used as a local measure, it yields better results in a DTW setting. In this regard, a lower bound estimate is derived for the inner product distance that significantly reduces the number of DTW computations (Zhang and Glass 2011). The proposed lower bound estimate eliminates 89 % of the previously required calculations in a segmental DTW setting without affecting the keyword spotting performance. Alternatively, the segment based DTW is replaced by a two-pass framework that integrates frame-based and segment-based DTW (Chan and Lee 2011). The segment based DTW is performed in the first pass to locate hypothesized spoken term regions followed by a frame-based DTW in the second pass on those regions for a precise rescoring. This method achieves a 54.6 % reduction of CPU time compared to frame-based DTW and a 8.4 % increase of MAP score compared to only segment-based DTW.

8.3.4 Techniques other than DTW for template matching

Apart from DTW, support vector machines (SVMs) have been used for matching of keyword templates in a sliding window setting. Features are computed for the segment of speech under an analysis window, the length of which is roughly same as that of the average keyword duration. These features are used to train a SVM from positive and negative example segments. In Ezzat and Poggio (2008), the fixed length spectro-temporal response vectors derived from the spectrogram segments of keywords (Sect. 8.2.2) are considered as positive training examples. The ones from the non-keyword segments are considered as negative training examples. The negative examples are chosen randomly from the speech segments of the training set that do not contain the keyword. The positive and the negative examples are used to train a two-class SVM to be used for classification of the test segments. The test segments are generated by sliding a window on the test utterance with overlaps. This approach consistently outperforms HMM-MFCC based keyword spotting. However, the performance decreases with increase in the size of the patch dictionary and increases with number of positive training examples.

The approach described in Barnwal et al. (2012) also uses SVM for classification in a sliding window setting. The *seam-hough* features obtained from segmented out instances of a keyword makes the positive training set. Similarly, the *seam-hough* features obtained from randomly drawn segments of speech that do not contain the target keyword constitute the negative training set. These set of positive and negative examples are used to train a SVM for the purpose of classifying a speech segment under the analysis window as an instance of the keyword. This approach surpasses the method in Ezzat and Poggio (2008) terms of performance accuracy.

9 Corpora and tools used in experiments

It is seen that most of the work reported in the survey used one or the other speech corpora available from Linguistic data consortium, USA.[2] The tools used for experimentation, specifically speech recognizers (word and phonetic)

[2] https://www.ldc.upenn.edu/.

Table 3 Speech corpora and the tools used in the methods reported in the survey

Method	Corpora	Other tools
Saraclar and Sproat (2004)	1998 HUB4, SWB, Teleconferences	–
Thambiratnam and Sridharan (2005)	TIMIT and WSJ (clean), SWB (telephony)	–
Mamou et al. (2007)	2006 NIST evaluation data	IBM 2004 CTS system, Juru
Szoke et al. (2008)	2006 NIST STD	–
Ezzat and Poggio (2008)	TIMIT	–
Parada et al. (2009)	HUB4	OpenFst toolkit, IBM 2004 CTS
Shen et al. (2009)	SWB, 2006 NIST evaluation data	AMI meeting transcription system (AMIDA project LVCSR)
Zhang and Glass (2009)	TIMIT, MIT lecture corpus	–
Hazen et al. (2009)	SWB (cellular)	BUT phone recognizer
Jansen et al. (2010)	SWB, Fisher	–
Tejedor et al. (2010)	Fisher, SWB	BUT STK toolkit
Chan and Lee (2010, 2011)	Mandarin broadcast news	–
Wang et al. (2011)	TIMIT, Fisher	–
Zhang and Glass (2011)	TIMIT	–
Huijbregts et al. (2011)	Dutch broadcast news	LVCSR-BN system
Kintzley et al. (2011)	TIMIT	–
Can (2011)	2006 STD evaluation data	–
Barnwal et al. (2012)	TIMIT	–

[a]SWB—Switchboard-1 Release 2 (LDC97S62)

[b]TIMIT—TIMIT acoustic phonetic speech corpus (LDC93S1)

[c]Fisher—Fisher English speech corpus (LDC2004S13 and LDC2005S13)

[d]WSJ—Wall street journal complete (LDC94S13A)

[e]HUB4—1998 HUB4 speech corpus (LDC2000S86)

are from both proprietary and open sources. The important open source tools used in this context are HTK toolkit[3] from Cambridge University, tools from Brno University of Technology (BUT phone recognizer[4] and STK toolkit[5]) and OpenFST[6] tookit. These were used in the supervised approaches for generation of lattices (word and subword) and generation of posteriorgrams (phonetic and Gaussian) in supervised QBE approaches. A summary of the corpora and tools used in different methods reported in the survey is given in Table 3.

10 Discussion and conclusion

The paper presents a comprehensive survey of recent developments in the field of spoken term detection. It is seen from the survey that both supervised and unsupervised approaches for STD have own their advantages and disadvantages and the choice is made depending on the context of usage. The best performing supervised approaches clearly surpasses their unsupervised counterparts both in terms of speed and accuracy. However, a major limitation of the supervised approaches is their requirement of huge amount of annotated speech resources for statistical training. The unsupervised approaches though have a low detection rate, but are suitable for use in zero/low resource environments. The supervised approaches rely on a high accuracy word (subword) recognizer to generate corresponding word (subword) lattices from which the indices are constructed. The use of lattices instead of 1-best output helps to accommodate errors in the output of the recognizer while the use of subword indices help to address the issue of OOV queries. The query terms are usually in text form and are searched in these indices. The size of the indices can be huge depending on the word (subword) vocabulary and size of the target utterances. Hence, recent research efforts focus on effective representation of the indices in terms of lesser storage space and faster searching. The use of WFST, PSPL, WCN and their variants to represent indices are steps in this direction.

[3] htk.eng.cam.ac.uk/.

[4] http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context.

[5] http://speech.fit.vutbr.cz/software/hmm-toolkit-stk.

[6] www.openfst.org.

The unsupervised approaches use template matching in a QBE framework. The spoken term queries are presented in speech form unlike text as in supervised approaches. Features are derived from examples of the spoken terms for representation of templates. These templates are matched with a similar representation of the target speech utterance in a sliding window setting. The matching process uses some variant of the DTW algorithm. In this context, posterior features in the form of phonetic posteriorgram, Gaussian posteriorgram and ASM posterior gram with segmental DTW have shown promising results, though they are yet to catch up with the performance of the supervised techniques. It is also seen that the above techniques incur considerable computational cost due to frame-wise DTW matching. Hence, recent efforts in this area are directed towards lessening the computational cost during matching, keeping the accuracy figures at an acceptable level. An instance to this point is the implementation of segmental-DTW based on inner-product distance between the posteriorgram vectors. In addition to speed and accuracy, issues related to indexing of spoken terms (queries) and the results in the context of unsupervised techniques in the QBE framework need to be studied from the perspective of effective storage.

Apart from the posterior features, templates comprising of spectrogram image features have been proposed in unsupervised template matching framework. It is observed that these methods exhibit performance better than HMM-MFCC methods. However, the efficacy of these methods have been studied only on clean speech. The next step can be in extending them to multi channel speech that is subjected to various distortions with appropriate pre-processing of the spectrogram images.

An important observation in the context of STD techniques in general, and the unsupervised techniques in particular is that most of the reported results have been performed on different databases and hence is difficult to compare the performance of different methods. The NIST 2006 STD evaluation workshop has been a major milestone that established the parameters for evaluation and benchmarked the performance of the participating systems on various types speech data (broadcast news, conversational telephone speech and meeting speech). However, the evaluation was carried out only for the well represented languages having sufficient annotated resources. From 2011 onwards, a STD track has been made a part of MediaEval workshops,[7] whose primay aim is to progress the current state-of-the-art in STD for low resource languages. This will help to benchmark the techniques for STD tasks with minimal amount of annotated speech resources, a requirement that will become a necessity in coming years.

[7] http://www.multimediaeval.org/.

References

Allauzen, C., Mohri, M., & Saraclar, M. (2004). General indexation of weighted automata: application to spoken utterance retrieval. In *HLT-NAACL*, Boston, USA.

Avidan, S., & Shamir, A. (2007). Seam carving for content-aware image resizing. *ACM Transactions on Graph*, *26*(3).

Baghai-Ravary, L., Kochanski, G., & Coleman, J. (2009). Data-driven approaches to objective evaluation of phoneme alignment systems. In *Proceedings of the 4th conference on human language technology*, Poznan, Poland.

Barnwal, S., Sahni, K., Singh, R., & Raj, B. (2012). Spectrographic seam patterns for discriminative word spotting. In *Proc. int. conf. acoustics, speech and signal processing*, Kyoto, Japan.

Benayed, Y. D., Fohr, J. H., & Chollet, G. (2003). Confidence measures for keyword spotting using support vector machines. In *Proc. int. conf. acoustics, speech and signal processing*, Hong Kong.

Boves, L., Carlson, R., Hinrichs, E., House, D., Krauwer, S., Lemnitzer, L., Vainio, M., & Wittenburg, P. (2009). Resources for speech research: present and future infrastructure needs. In *Proc. int. conf. speech processing*, Brighton, UK.

Bridle, J. (1973). An efficient elastic template method for detecting given key words in running speech. In *Proc. of British acoustic society meeting*, UK.

Can, D. (2011). Lattice indexing for spoken term detection. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(8), 2338–2347.

Can, P., Cooper, E., Sethy, A., White, C., Ramabhadran, B., & Saraclar, M. (2009). Effect of pronunciations on oov queries in spoken term detection. In *Proc. int. conf. acoustics, speech and signal processing*, Taipei, Taiwan.

Chan, C., & Lee, L. (2010). Unsupervised spoken-term detection with spoken queries using segment-based dynamic time warping. In *Proc. int. conf. speech processing*, Chiba, Japan.

Chan, C., & Lee, L. (2011). Integrating frame-based and segment-based dynamic time warping for unsupervised spoken term detection with spoken queries. In *Proc. int. conf. acoustics, speech and signal processing*, Prague.

Chelba, C., & Acero, A. (2005). Position specific posterior lattices for indexing speech. In *Annual conference of the association of computational linguistics*, Ann Arbor, USA.

Deligne, S., & Bimbot, F. (1995). Language modeling by variable length sequences. In *Proc. int. conf. acoustics, speech and signal processing*, Michigan, USA.

Ezzat, T., & Poggio, T. (2008). Discriminative word spotting using ordered spectro-temporal patch features. In *ISCA workshop statistical and perceptual audition*, Brisbane, Australia.

Fousek, P., & Hermansky, H. (2006). Towards ASR based on hierarchical posterior-based keyword recognition. In *Proc. int. conf. acoustics, speech and signal processing*, Toulouse, France.

Garcia, A., & Gish, H. (2006). Keyword spotting of arbitrary words using minimal speech resources. In *Proc. int. conf. acoustics, speech and signal processing*, Toulouse, France.

Garofolo, J., Auzzane, G., & Voorhees, E. (2000). The trec spoken document retrieval track: a success story. In *Ninth text retrieval conference (TREC-9) NIST*.

Grangier, D., Keshet, J., & Bengio, S. (2009). Chapter on discriminative keyword spotting. In *Automatic speech and speaker recognition: large margin and kernel methods*. New York: Wiley.

Hakkani-Tur, D., & Riccardi, G. (2003). A general algorithm for word graph matrix decomposition. In *Proc. int. conf. acoustics, speech and signal processing*, Hong-Kong.

Hazen, T., Shen, W., & White, C. (2009). Query-by-example spoken term detection using phonetic posteriorgram templates. In *Proc. IEEE workshop on automatic speech recognition and understanding*, Merano, Italy.

Huijbregts, M., McLaren, M., & Leeuwen, D. V. (2011). Unsupervised acoustic sub-word unit detection for query-by-example spoken term detection. In *Proc. int. conf. acoustics, speech and signal processing*, Prague.

James, D., & Young, S. (1994). A fast lattice-based approach to vocabulary independent wordspotting. In *Proc. int. conf. acoustics, speech and signal processing*, Adelaide, Australia.

Jansen, A., & Niyogi, P. (2009). Point process models for spotting keywords in continuous speech. *IEEE Transactions on Audio, Speech, and Language Processing*, *17*(8), 1457–1470.

Jansen, A., Church, K., & Hermansky, H. (2010). Towards spoken term discovery at scale with zero resources. In *Proc. int. conf. speech processing*, Chiba, Japan.

Keshet, J., Grangier, D., & Bengio, S. (2007). Discriminative keyword spotting. In *Proc. of workshop on non-linear speech processing*, Paris, France.

Kintzley, K., Jansen, A., & Hermansky, H. (2011). Event selection from phone posteriorgrams using matched filters. In *Proc. int. conf. speech processing*, Florence, Italy.

Lehtonen, M., Fousek, P., & Hermansky, H. (2005). IDIAP research report: hierarchical approach for spotting keywords.

Mamou, J., Ramabhadran, B., & Siohan, O. (2007). Vocabulary independent spoken term detection. In *Proc. ACM special interest group on information retrieval*, New York, USA.

Mangu, L., Brill, E., & Stolcke, A. (2000). Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, *14*(4), 373–400.

Meyers, C., Rabiner, L., & Rosenberg, A. (1980). Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *28*(6), 623–635.

Mohri, M., Pereira, F., Pereira, O., & Reiley, M. (1996). Weighted automata in text and speech processing. In *ECAI workshop*.

Ng, K., & Zue, V. (2000). Subwordbased approaches for spoken document retrieval. *Speech Communication*, *32*(3), 157–186.

Novotney, S., Schwartz, R., & Ma, J. (2009). Unsupervised acoustic and language model training with small amounts of labelled data. In *Proc. int. conf. acoustics, speech and signal processing*, Taipei, Taiwan.

Pan, Y. C., & shan Lee, L. (2010). Performance analysis for lattice-based speech indexing approaches using words and subword units. *IEEE Transactions on Audio, Speech, and Language Processing*, *18*(6), 1562–1574.

Parada, C., Sethi, A., & Ramabhadran, B. (2009). Query-by-example spoken term detection for oov terms. In *Proc. IEEE workshop on automatic speech recognition and understanding*, Merano, Italy.

Park, A. S., & Glass, J. (2008). Unsupervised pattern discovery in speech. *IEEE Transactions on Audio, Speech, and Language Processing*, *16*(1), 186–197.

Rohlicek, J. R. (1995). Chapter on word spotting. In *Modern methods of speech processing*, Norwell: Kluwer Academic.

Rose, R. C. (1996). Word spotting from continuous speech utterances. In *Automatic speech and speaker recognition: advanced topics*, Norwell: Kluwer Academic.

Rose, R. C., & Paul, D. B. (1990). A hidden Markov model based keyword recognition system. In *Proc. int. conf. acoustics, speech and signal processing*, Albuquerque, USA.

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *26*(1), 43–49.

Sandness, E., & Hetherington, I. (2000). Keyword-based discriminative training of acoustic models. In *Proc. int. conf. speech and language processing*, Beijing, China.

Saraclar, M., & Sproat, R. W. (2004). Lattice based search for spoken utterance retrieval. In *HLT-NAACL*, Boston, USA.

Shen, W., White, C., & Hazen, T. (2009). A comparison of query-by-example methods for spoken term detection. In *Proc. int. conf. speech processing*, Brighton, UK.

Silaghi, M., & Bourlard, H. (1999). Iterative posterior-based keyword spotting without filler models. In *Proc. IEEE workshop on automatic speech recognition and understanding*, Colorado, USA.

Sukkar, R., Seltur, A., Rahim, M. G., & Lee, C. H. (1996). Utterance verification of keyword strings using word-based minimum verification error training. In *Proc. int. conf. acoustics, speech and signal processing*, Atlanta, USA.

Szoke, I., Schwarz, P., Patejka, P., Burget, L., Karafiat, M., Fapso, M., & Cernocky, J. (2005). Comparison of keyword spotting approaches for informal continuous speech. In *Eurospeech*, Lisbon, Portugal.

Szoke, I., Burget, L., Cernocky, J., & Fapso, M. (2008). Sub-word modeling of out-of-vocabulary words in spoken term detection. In *Spoken language technology workshop*, Goa, India.

Tejedor, J., Szoke, I., & Fapso, M. (2010). Novel methods for query selection and combination in query-by-example spoken term detection. In *ACM workshop on searching spontaneous conversational speech*, Firenze, Italy.

Thambiratnam, K., & Sridharan, S. (2005). Dynamic match phone-lattice searches for very fast and accurate unrestricted vocabulary keyword spotting. In *Proc. int. conf. acoustics, speech and signal processing*, Philadelphia, USA.

Vergyri, D., Shafran, I., Stocke, A., Gadde, R., Akbacak, M., Roark, B., & Wang, W. (2007). The sri/ogi 2006 spoken term detection system. In *Proc. int. conf. speech processing*, Antwerp, Belgium.

Wang, H., Lee, T., & Leung, C. (2011). Unsupervised spoken term detection with acoustic segment model. In *Int. conf. speech database and assessments*, China.

Weintraub, M., Beaufays, F., Rivlin, Z., Konig, Y., & Stolcke, A. (1997). Neuralnetwork based measures of confidence for word recognition. In *Proc. int. conf. acoustics, speech and signal processing*, Munich, Germany.

Wright, C., Ballar, L., Coull, S., Monrose, F., & Masson, G. (2010). Uncovering spoken phrases in encrypted voice over IP conversations. *ACM Transactions on Information and System Security*, *13*(4), 35.1–35.30.

Zhang, Y., & Glass, J. (2009). Unsupervised spoken keyword spotting via segmental dtw on Gaussian posteriorgrams. In *Proc. IEEE workshop on automatic speech recognition and understanding*, Merano, Italy.

Zhang, Y., & Glass, J. (2011). An inner-product lower-bound estimate for dynamic time warping. In *Proc. int. conf. acoustics, speech and signal processing*, Prague.