Design of QbE-STD System: Audio Representation and Matching Perspective

by

Maulik C. Madhavi
201121003

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

DHIRUBHAI AMBANI INSTITUTE OF INFORMATION AND COMMUNICATION TECHNOLOGY

October, 2017

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Doctor of Philosophy in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Maulik C. Madhavi

Certificate

This is to certify that the thesis work entitled, "*Design of QbE-STD System : Audio Representation and Matching Perspective*," has been carried out by *Maulik C. Madhavi* for the degree of *Doctor of Philosophy* in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my supervision.

Prof. Hemant A. Patil
Thesis Supervisor

Acknowledgments

First and foremost, I would like thank the Almighty God for providing me the courage and capability to proceed and to complete this thesis work successfully. Many extraordinary people assisted me directly or indirectly to achieve this work. I would mention just a few of them.

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Hemant A. Patil for his insightful suggestions and expert guidance. His patience, motivation, enthusiasm, immense knowledge and his friendly nature helped me a lot to produce this fruitful outcome. I could not have imagined having a better supervisor (and mentor) than Prof. Hemant A. Patil. I am fortunate to have excellent support from him during my doctoral study.

I would also like to thank DA-IICT, my research progress seminar (RPS) committee (Prof. L. Pillutla and Prof. A. Tatu) and thesis examination committee (Prof. A. Banerjee, Prof. L. Pillutla, and Prof. Y. Vasavada) for suggesting valuable inputs to my work. I acknowledge my thesis examiners, namely, Prof. Hynek Hermansky (Johns Hopkins University) and Prof. K. Sri Rama Murty (IIT Hyderabad) for valuable technical inputs related to the thesis. Besides them, I would also thank all the staff and faculty members of DA-IICT, who have been kind enough to help me throughout my stay at DA-IICT. I have learned many things, and their help shaped my career as a researcher. I thank Resource Center (RC), DA-IICT for creating research excelling atmosphere and infrastructure. I would like to thank the organizers of MediaEval SWS 2013 and MediaEval QUESST 2014 for providing the database, which is extensively used in this thesis. I would also like to thank Prof. (Dr.) Cheung-Chi Leung and his team for sharing MATLAB code of Acoustic Segment Model (ASM) training, which was used in Chapter 4.

I also sincerely thank Department of Electronics and Information Technology (DeitY), Govt. of India, for sponsoring a consortium project (in which I was a project staff member) authorities of DA-IICT for providing excellent infrastructure. I would like to thank anonymous reviewers, who reviewed my work submitted to journals and conferences. They provided very useful insights into the work, which tremendously helped me to revise the particular paper as well as this thesis.

Besides them, I would also thank my lab-mates of Speech Research Lab, DA-IICT. The fruitful technical discussions with all the members were incredible learning opportunities. Thanks to all for creating a continuous learning environment in the lab. The team effort and sharing learning experiences across the members tremendously helped in shaping the thesis and developing me as a good team member. We all worked together to meet the deadlines and enjoyed a lot several celebrating moments during the last six years.

Last but not the least, I would like to thank my family, my parents, brother and wife. I have no words to describe their huge support and love. *Thanks* is very tiny word to describe their affection to me. They missed me during our family get-together, and supported me throughout the doctoral study. I must acknowledge them, without their encouragement, love, care and blessings, I would not have finished this thesis.

Contents

Abstract

The retrieval of the spoken document and detecting the query (keyword) within the audio document have attained huge research interest. The problem of retrieving audio documents and detecting the query (keyword) using a spoken form of a query is widely known as Query-by-Example Spoken Term Detection (QbE-STD). This thesis presents the design of QbE-STD system from the representation and matching perspective.

A speech spectrum is known to be affected by the variations in the length of the vocal tract of a speaker due to the *inverse* relation between formants and vocal tract length. The process of compensating spectral variation caused due to the length of the vocal tract is popularly known as Vocal Tract Length Normalization (VTLN) (especially, in speech recognition literature). VTLN is a very important speaker normalization technique for speech recognition task. In this context, this thesis proposes the use of Gaussian posteriorgram of VTL-warped spectral features for a QbE-STD task. This study presents the novel use of a Gaussian Mixture Model (GMM) framework for VTLN warping factor estimation. In particular, presented GMM framework does *not* require phoneme-level transcription and hence, it can be useful for the unsupervised task. In addition, we also propose the use of the mixture of GMMs for posteriorgram design. The speech data governs acoustically similar broad phonetic structures. To capture broad phonetic structure, we exploit supplementary knowledge of broad phoneme classes (such as, vowels, semi-vowels, nasals, fricatives, plosive) for the training of GMM. The mixture of GMMs is tied with GMMs of these broad phoneme classes. A GMM trained under no supervision assumes uniform priors to each Gaussian component, whereas a mixture of GMMs assigns the prior probability based on broad phoneme class. The novelty of our work lies in prior probability assignments (as weights of the mixture of GMMs) for better Gaussian posteriorgram design.

In realistic scenarios, there is a need to retrieve the query, which does not appear exactly in the spoken document. However, the appeared instance of query might have the different suffix, prefix or word order. The DTW algorithm monotonically aligns the two sequences, and hence, it is not suitable to perform partial matching between the frame sequence of query and test utterance. We propose novel partial matching approach between spoken query and utterance using modified DTW algorithm, where multiple warping paths are constructed for each query and test utterance pair. This partial matching approach improves the detection of the non-exact query in the realistic scenarios, where both exact and non-exact queries are present. Next, we address the research issue associated with search complexity of DTW algorithm and suggest two approaches, namely, feature reduction approach and segment-level Bag-of-Acoustic-Words (BoAW) model. In feature reduction approach, the number of feature vectors is reduced by averaging across the consecutive frames within phonetic boundaries. Thus, a lesser number of feature vectors require a fewer number of comparison operations and hence, DTW speeds up the search computation. In BoAW model, we construct term frequency-inverse document frequency $(tf - idf)$ vectors at segment-level to retrieve audio documents. The proposed segment-level BoAW model is used to match test utterance with a query using $(tf - idf)$ vectors and the scores obtained are used to rank the test utterance. Both of these search space reduction approaches are used to speed up the execution with a slight degradation in the search performance.

We propose two-stage approaches for re-scoring the detection hypothesis with the help of acoustic features and detection sources. First, we explored several acoustic features to re-score the detection hypothesis. The second approach considers additional detection sources, such as, depth of detection valley and term-frequency, Self-Similarity Matrix (SSM), Pseudo Relevance Feedback (PRF) and Weighted mean feature with Gaussian and phonetic posteriorgram. These two-stage approaches improve the detection performance with the re-scoring from the hypothesis of a single QbE-STD system. Finally, the thesis concludes by presenting few miscellaneous studies, a summary of entire thesis, along with few potential future research directions.

List of Acronyms

AKWS	Acoustic Keyword Spotting
AM	Acoustic Model
AP	Average Precision
AR-BNF	Articulatory Bottleneck Features
ASM	Acoustic Segment Model
ASR	Automatic Speech Recognition
BNF	Bottleneck Features
BoAW	Bag of Acoustic Words
BoW	Bag of Words
BUT	Brno University of Technology
CD-DNN	Context-Dependent Deep Neural Network
CDTW	Cumulative Dynamic Time Warping
CZ	Czech Language
DBN	Deep Belief Networks
DCT	Discrete Cosine Transform
DET	Detection Error Trade-off
Dev	Devlopement
DFS	Depth-First Search
DFW	Dynamic Frequency Warping
DLG	Double-Layer neighborhood Graph
DNN	Deep Neural Network
DP	Dynamic Programming
DPGMM	Dirichlet Process Gaussian Mixture Model
DTW	Dynamic Time Warping
DTW-SS	Dynamic Time Warping String Search
EHMM	Ergodic Hidden Markov Model
EM-ML	Expectation Maximization-Maximum Likelihood

EN	English
Eval	Evaluation
FA	False Alarm
FBCCs	Fourier-Bessel Cepstral Coefficients
G	Gujarati Language
GBRBM	Gaussian-Bernoulli Restricted Boltzmann Machines
GCC	Gaussian Component Clustering
GMM	Gaussian Mixture Model
GMM-FST	Gaussian Mixture Model-Finite State Transducers
GP	Gaussian Posteriorgram
GPUs	Graphical Processing Units
GS	Greedy Search
GSS	Graph-based Similarity Search
HAC	Hierarchical Agglomerative Clustering
HGSS	Hierarchical Graph-based Similarity Search
HMM	Hidden Markov Model
HTK	Hidden Markov Model ToolKit
HU	Hungarian Language
IO	Input-Output
IPA	International Phonetic Alphabet
IPAs	Intelligent Personal Assistants
IR	Information Retrieval
IR-DTW	Information Retrieval Dynamic Time Warping
ISA	Intrinsic Spectral Analysis
k-DR	Degree Reduced k-nearest neighbors
k-NN	k-nearest neighbors
KL	Kullback-Leibler
KWS	KeyWord Spotting
LB	Lower Bound
LBG	Linde-Buzo-Gray algorithm
LBP	Local Binary Patterns
LLR	Log-likelihood Ratio
LP	Linear Prediction

LPCCs	Linear Prediction Cepstral Coefficients
LPCs	Linear Prediction Coefficients
LSH	Locality Sensitivity Hashing
LSTM	Long Short Term Memory
M	Marathi Language
MA	Mandarin
MAP	Mean Average Precision
MCA	Minimum Cost of Alignment
MDL	Minimum Description Length
MEX	MATLAB EXecutable
MFCC-TMP	Mel Frequency Cepstral Coefficients, where subband energy is computed via Teager Energy Operator considering the Magnitude and Phase spectra of subband signals
MFCCs	Mel Frequency Cepstral Coefficients
MIR	Music Information Retrieval
MLE	Maximum Likelihood Estimation
MLP	Multilayer Perceptron
MSC	Multi-view Segment Clustering
MTWV	Maximum Term Weighted Value
NMI	Normalized Mutual Information
NTSS	Note Taking Support System
OLN	On-the-fly Length Normalization (Online Length Normalization)
OOV	Out-of-Vocabulary
PAKWS	Parallel Acoustic Keyword Spotting
PCA	Principal Component Analysis
pdf	Probability density function
PE	Phonetic Engine
PER	Phone Error Rate
PF	Progressive Filter
PhnRec	Phoneme Recognizer
PLP	Perceptual Linear Prediction
PNCCs	Power Normalized Cepstral Coefficients
PP	Phonetic Posteriorgram
PRF	Pseudo-Relevance Feedback

PSOLA	Pitch Synchronous Overlap and Add
QbE-STD	Query-by-Example Spoken Term Detection
QbH	Query-by-Humming
QbSH	Query-by-Singing/Humming
QUESST	Query-by-Example Search on Speech Task
RAILS	Randomized Acoustic Indexing and Logarithmic-time Search
RBM	Restricted Boltzmann Machines
RMS	Root Mean Square
RNN	Recurrent Neural Network
RU	Russian Language
SAD	Speech Activity Detection
SAM	Spectral Acoustic Model
SBNF	Stacked Bottleneck Features
SDR	Spoken Document Retrieval
SDTW	Segmental Dynamic Time Warping
SLNDTW	Segmentally Local Normalized Dynamic Time Warping
SMS	Short Message Service
SSM	Self-Similarity Matrix
STC-NN	Split Temporal Context Neural Network
STD	Spoken Term Detection
STM	Spectral Transition Measure
SubDTW	Subsequence Dynamic Time Warping
SVMs	Support Vector Machines
SWS	Spoken Web Search
TAM	Temporal Acoustic Model
TEO	Teager Energy Operator
TRAP	TempoRAl Pattern
TWV	Term Weighted Value
VAMFCC	Variance of Acceleration of Mel Frequency Cepstral Coefficients
VQ	Vector Quantization
WFST	Weighted Finite State Transducer

List of Symbols

α	VTLN warping factor
β	Feature reduction factor
λ_{init}	Initial GMM model with parameters, $(\mu_{init}, \Sigma_{init}, w_{init})$
$\mathbf{o}_t$	Observation vector at t^{th} frame or t^{th} speech frame index
B_k	k^{th} broad class
C_{nxe}	Normalized Cross Entropy Cost
C_{nxe}^{min}	minimum Normalized Cross Entropy Cost
D	Local distance matrix
F_n	n^{th} formant frequency for uniform tube model of vocal tract
K	Number of broad phoneme class
M	Number of frames in query
M_k	Number of Gaussian components in k^{th} broad phoneme class
M_s	Total number of segments in test utterance
N	Number of frames in test utterance
NC	Number of context in BoAW
NS	Number of QbE-STD systems used in discriminative calibration fusion
N_p	Number of classes in posteriorgram or dimension of posteriorgram
N_s	Total number of segments in query
Nd	Number of documents in BoAW model
$Nseg$	Number of consecutive acoustic segments considered while tf of BoAW model computation
O	Acoustic observation sequence
P	Path matrix that indicates start frame
$P(B_k\|\mathbf{o}_t)$	Posterior probability for k^{th} broad phoneme class and t^{th} speech frame index
$P(C_k\|\mathbf{o}_t)$	Posterior probability for k^{th} cluster and t^{th} speech frame index
S	Accumulated distance matrix
W	Word transcription
X^{α}	VTL-warped acoustic features with warping factor α

δ	Pruning threshold in BoAW
$\mathbf{o}_t$	t^{th} observation or acoustic feature vector
ξ_i	fusion or calibration coefficients, which are estimated by binary logistic regression
d	Document in BoAW model
$d_B(k)$	Discrimination capability of k^{th} broad phoneme class
idf	Inverse Document Frequency
t	Term in BoAW model
tf	Term Frequency
v	Velocity of the sound wave ($\approx 344m/s$ at sea-level)

List of Tables

List of Figures

CHAPTER 1
Introduction

1.1 Motivation

In the recent era, INTERNET has been very important communication media for sharing the information across the globe. The information available in the INTERNET is mostly in the form of text-based documents. Now-a-days, a lot of information is also available in the form of multimedia. In particular, spoken (audio) form of information is important, which is stored in the form of audio and video recordings. This information continuously grows in the future, for example, news broadcast, YouTube video, etc. This information has become inevitable need of the society, for knowledge and entertainment purpose. In addition, spoken information can be thought as an alternative representation of written (text) information so far as human communication is concerned. In such scenario, access and retrieval of audio information helps us tremendously for various purposes [2].

Human speech is easily available, i.e., natural to produce and provides an easy access, which primarily does not require a physical interface. In addition, the interaction between human and a machine does not necessarily require costly hardware, merely audio input-output (IO) devices (i.e., microphones and speaker), which are smaller in size and consume low power [3]. Recently, during the last decade, many technological companies are involved with the design of Intelligent Personal Assistants (IPAs). IPAs are the software agents that takes input from the user in the form of audio and responds to the query. Thus, IPAs form interactive channels between voice input from the user and the database system. The well known IPAs available are Apple's Siri, Google Now, Microsoft Cortana, Amazon Echo, etc. [4]. Figure 1.1 (a) shows the chronological development of voice-enabled IPAs during last few years. Recently, an open source project *Sirius* started to explore the voice as well as image query from the user [5]. The schematic of open end-to-end Sirius system is shown in Figure 1.1 (b).

Mobile devices, such as, computers, mobile phones, smart watches are used

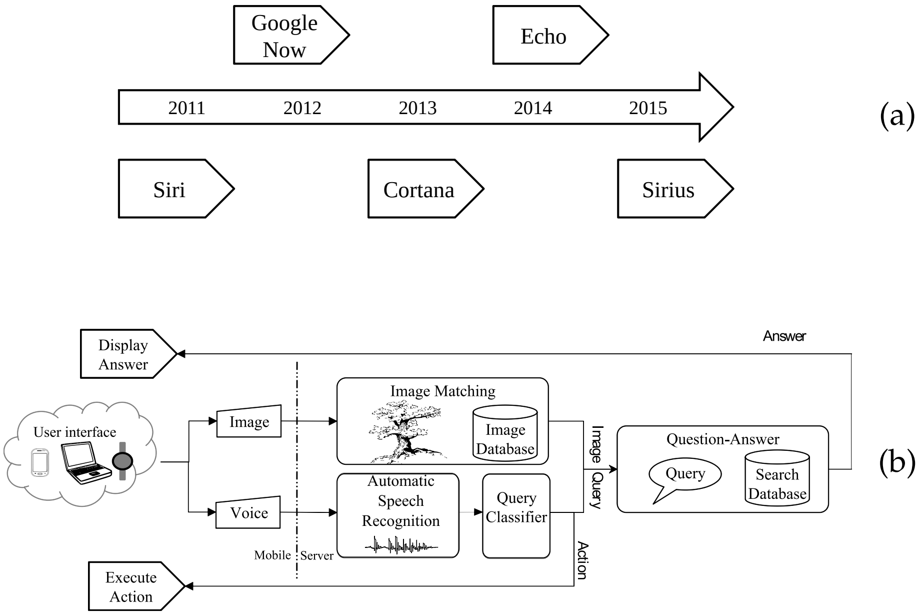


Figure 1.1: Voice-enabled IPAs: (a) The development of voice-driven IPAs. After [4], (b) an architecture of Sirius (Lucida) end-to-end architecture. After [5].

to take input from the user. The response from the system is either displayed on respective devices or executed in the form of action. Many mobile users prefer to stay connected wherever they are and whatever they are doing. For instance, to communicate via Short Message Service (SMS) through a mobile phone, a cab driver chooses to use voice rather than typing the SMS, i.e., *hands free and even eyes free* mode of communication [6]. The spoken dialogue systems use the state-of-the-art speech technologies, such as, speech recognition, speech synthesis, and machine translation to provide human and machine interface through the voice. However, there are challenges created by the poor performance of speech recognition due to noisy automobile ambiance as well as head movements of the driver [6].

The recent technological advancements allow recording and storing of vast collections of speech or audio data with various contents. In addition, classroom lectures that are stored in audio form can be effectively browsed and accessed if one can have reliable audio retrieval mechanism [1]. Furthermore, the current speech technology allows access to speech data effectively via telephone. Thus, it is important to seek for automatic, reliable and fast solutions to search large collections of speech data. The information retrieval associated with spoken content

[1] Few online portals for online video lectures are: (1) http://nptel.ac.in/courses.php, (2) http://ocw.mit.edu/courses/audio-video-courses/, (3) https://www.ted.com/talks, etc.

has been important area of research for spoken language processing area. Spoken Content Retrieval (SCR) deals with information retrieval and processing on spoken media to provide access and control to the user. SCR system focuses on the retrieval of linguistic message as well as semantics associated with the speech data. However, in this thesis, we focus only the retrieving linguistic message information present in the speech data. The SCR system aims to retrieve audio documents from the collection of large number of documents and detect the location of query. The detailed review on SCR is presented in [7,8]. The query can be exploited in different form, i.e., either text or spoken form.

1. Text-based query representation: The problem is known as *Spoken Term Detection (STD)*.

2. Spoken-example query representation: The problem is known as *Query-by-Example Spoken Term Detection (QbE-STD)*.

In that context, several applications were explored, basically enhancing speech technology for SCR. STD technology was used for electronic note-taking support system (NTSS) [9]. The NTSS is equipped with speech interface and Automatic Speech Recognition (ASR). The user can simply touch and trace the notes by their fingers and avoid the difficulty, while preparing the notes during classroom lecture. The ease of access to speech also motivated to annotate (tag) and search digital photographs or images [10,11].

The organization of this chapter is as follows. Section 1.2 discusses the SCR systems, namely, STD, keyword spotting (KWS) and QbE-STD. Section 1.3 discusses the focus of this thesis and the contributions. Section 1.4 presents the overall organizations from the thesis.

1.2 Spoken Content Retrieval (SCR) Systems

1.2.1 Spoken Term Detection (STD)

STD evaluation campaign was initiated by the NIST, the USA in 2006 [12]. The term corresponds to a word or a sequence of words, which is to be detected in the audio documents. The objective was to use speech technology, in particular, Automatic Speech Recognition (ASR), for the audio retrieval using text query.

The state-of-the-art STD system is the cascade connection of ASR and text-retrieval system [7]. An ASR system can be seen as a nonlinear transformation from the speech signal to a sequence of words [13]. The core ASR architecture

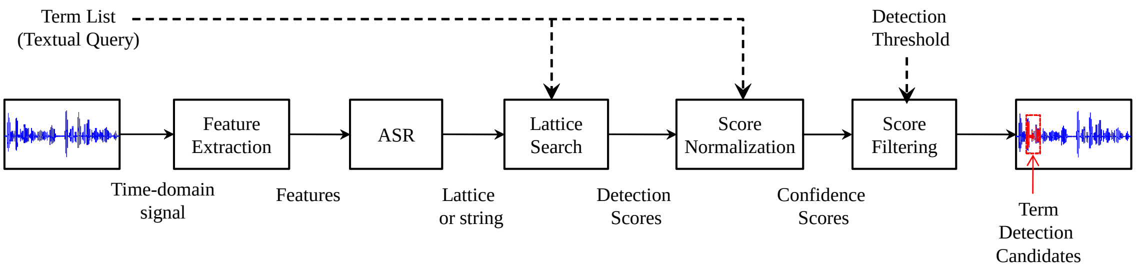


Figure 1.2: Block diagram representation of STD system. The red colored speech waveform inside the dashed box indicates to the detection candidate. After [14].

has been constructed on the Bayesian probabilistic framework. In particular, the acoustic observation sequence, $O = \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_n$, the objective function of an ASR is to determine the best word transcript, $\widehat{W} = w_1, w_2, \cdots, w_m$, that maximizes the posterior probability, $P(W|O)$, as [13]:

$$\widehat{W} = \underset{W}{\text{argmax}}\{P(W|O)\} = \underset{W}{\text{argmax}}\frac{P(O|W)P(W)}{P(O)}. \tag{1.1}$$

The estimate is highly dependent on two models, namely, acoustic model $P(O|W)$ and language model $P(W)$.

ASR system decodes speech into word transcript, and the term is searched in the decoded transcripts. The output from one-best hypothesis seems erroneous so far as the performance of ASR is concerned. To cope up with the issues of ASR errors, ASR system produces multiple word transcripts or an output in terms of the lattice [7,15,16]. The traditional STD framework for audio content retrieval is shown in Figure 1.2. The detection scores are transformed into confidence scores by employing score normalization (which takes into account scores of different terms that have different dynamic range). The detection threshold is applied to filter out confidence scores and obtain few detection candidates, whose confidence scores are higher than the detection threshold [14].

The lattice information needs to be effectively stored for the fast retrieval. Weighted Finite State Transducer (WFST) is one of the way to represent lattice for STD task [17]. If a query contains Out-of-Vocabulary (OOV) words that does not appear in ASR vocabulary, an ASR cannot generate output for STD. To resolve this, ASR system is modelled on subword-level, i.e., smaller unit than the word, such as, phonemes or syllables. Such subword-based ASR generates subword sequences or lattice and then searching is performed by minimum edit distance criteria [15,18,19]. In order to search quickly using subword units, output of ASR has to be converted into index representation. Few such indexing schemes are suffix indexing scheme [20] and metric subspace indexing [21]. STD system exploits the well-trained speech recognizer to convert speech utterances into word-sequence or word lattice. In the case of OOV word, STD system generates subword, i.e., phoneme or syllable-level sequence. In order to perform matching, soft-matching (approximation) is needed to overcome the insertions, deletion and substitution errors. This is also known as *phonetic search*. In phonetic search approach, each text query is converted into string of phonemes via pronunciation dictionary.

1.2.1.1 Research Issues in STD

The major research issue in STD is to have abundant resources for ASR building, efficient indexing and searching methods to speed up the search; and OOV words that are very often considered as a part of query. Speech in vernacular language may contain multiple code-switching, where ASR systems cannot be useful [22].

1.2.2 Query-by-Example Spoken Term Detection (QbE-STD)

As discussed earlier, STD technology requires the ASR system (to generate transcription) and text-retrieval (for content/message retrieval). [7]. However, the lack of resources in few languages (which are called as under-resourced languages) makes the ASR building infeasible and SCR in such language cannot be feasible without ASR system. Thus, ASR system suffers from OOV word recognition and hence, STD system required subword (such as, phoneme, syllable, etc.) recognizer. In this context, a study was conducted to exploit spoken example and phoneme transcription of OOV query [23]. QbE-STD is important for low-resourced languages and under non-mainstream conditions, where ASR is not available or feasible to develop and hence, it was also called as unsupervised STD [7,24,25].

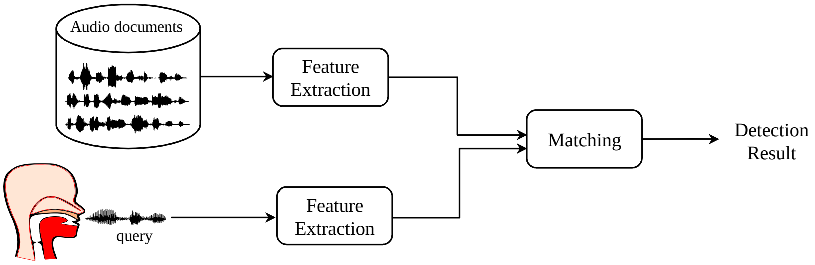


Figure 1.3: Block diagram of QbE-STD system.

The architecture of QbE-STD system is shown in Figure 1.3. The use of spoken example offers audio retrieval in multi-lingual scenario (i.e., audio documents belong to more than one language). Inspired by this motivation, MediaEval campaign commenced, Spoken Web Search (SWS) task in 2011 [26]. This task involved language-independent audio search for low-resourced languages. This task was held almost every year in MediaEval [26–30]. Since 2014, SWS task was renamed as *Query-by-Example Search on Speech Task* (QUESST) to well exhibit the fundamental characteristics of QbE-STD system, i.e., query in the form of spoken example [29, 31]. More detailed reviews related to QbE-STD systems are presented in Section 2.8 of Chapter 2.

1.2.2.1 Research Issues in QbE-STD

The major research issues in QbE-STD framework are audio representation and the complexity of search algorithm.

- The audio representation should be speaker-invariant (i.e., w.r.t. speaker variabilities). In addition, it should emphasizes phonetic information more rather than the speaker and other paralingual characteristics.
- QbE-STD is very slow due to computationally intensive Dynamic Programming (DP)-based algorithm, with frame-based representation. Matching task has to be executed fast that demands another indexing or search space reduction approaches without much affecting matching performance.
- For realistic scenario, QbE-STD system should consider the non-exact variants of query, such as, variation of query at suffix, prefix or word ordering.

More detailed discussion on these research issues is presented in Section 2.7.

1.2.3 Keyword Spotting System

The Keyword Spotting (KWS) problem is slightly different than the ASR. The keyword spotter involves in locating few set of words rather than the optimal word sequence as in ASR [32]. In a traditional keyword spotting framework, two acoustic models are used, namely, keyword model for modeling the keyword and background model to model entire speech signal [19,33]. The schematic block diagram for acoustic keyword spotting (AKWS) is shown in Figure 1.4. The likelihood ratio between keyword model and the background model is used to compute the detection threshold. These detection scores are normalized (transformed into confidence scores) and filtered to get detection candidates, whose confidence score is higher than the detection threshold.

In this query detection, the score is computed by taking likelihood ratio between the phoneme sequence passing through keyword sequence and an arbitrary phoneme loop (ergodic model). This model is applicable, when pre-defined

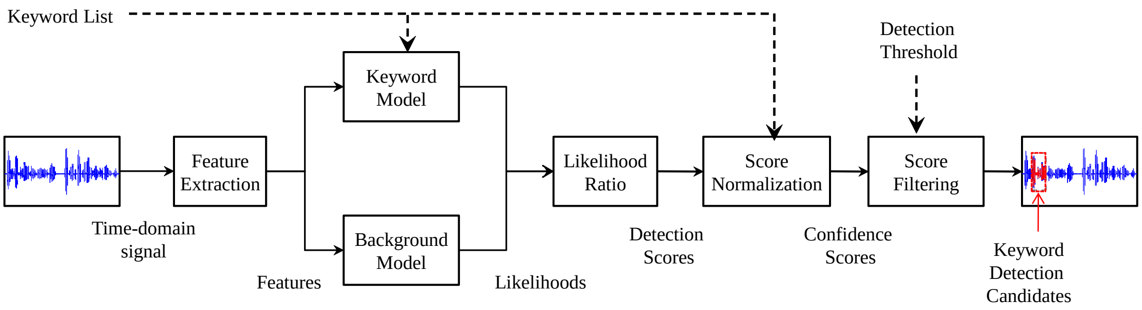


Figure 1.4: Block diagram representation of AKWS system The red colored speech waveform inside the dashed box indicates to the detection candidate. After [14].

set of queries are available. This framework can be useful for text and spoken query realization. In traditional STD scenario, keyword is described by phonemic sequence using pronunciation dictionary and Viterbi decoders can be used [23,34].

1.2.3.1 Research Issues in KWS

The major limitation of AKWS is its applicability for fixed set of keywords. The entire system needs to be retrained for the new set of keywords. In addition, KWS problem was not attempted for non-exact keyword matching task, where the objective is to detect the keyword having lexical variation either at suffix or prefix.

1.3 Focus and Contributions in the Thesis

The major focus of present thesis is to exploit spoken query and build language-independent QbE-STD task. In this thesis, we present the speech representation and matching perspective for design of QbE-STD systems. The brief summary of contributions in the thesis are as follows:

1.3.1 GMM Framework for VTLN

In this thesis, in order to make query and document invariant w.r.t. speaker, we exploit Vocal Tract Length Normalization (VTLN) approach. In particular, we used Gaussian posteriorgram of VTL-warped spectral features for a QbE-STD task. The novel use of a Gaussian Mixture Model (GMM) framework for VTLN warping factor estimation is presented. In particular, presented GMM framework does *not* require phoneme-level transcription and hence, it can be useful for the unsupervised task. In GMM-based VTLN warping factor estimation approach, initially GMM parameters are estimated using unwarped features, i.e., having

VTLN warping factor $\alpha = 1$. Then, the optimal VTLN warping factors $\hat{\alpha}$ are estimated based on Maximum Likelihood Estimation (MLE). In the next cycle, GMM parameters are re-estimated with the help of the features having optimal warping factor, i.e., $\hat{\alpha}$. The GMM framework along with proposed iterative variant estimates the VTLN warping factor and captures the spectral scaling suitable to adjust with the speaker-independent scenario. We will discuss VTL-warped Gaussian posteriorgram in Chapter 4.

1.3.2 Mixture of GMMs for Posteriorgram Design

The speech data governs acoustically similar broad phonetic structures. To capture broad phonetic structure, we exploit supplementary knowledge of broad phoneme classes (such as, vowels, semi-vowels, nasals, fricatives, plosive) for the training of GMM. The mixture of GMMs is tied with GMMs of these broad phoneme classes. A GMM trained under no supervision assumes uniform priors to each Gaussian component, whereas a mixture of GMMs assigns the prior probability based on broad phoneme class. The novelty of this work lies in prior probability assignments (as weights of the mixture of GMMs) for better Gaussian posteriorgram design. The proposed posterior features from the mixture of GMMs outperform Gaussian posteriorgram because of its implicit constraints supplied by broad phonetic posteriorgram. We will discuss posteriorgram representation based on mixture of GMMs in Chapter 4.

1.3.3 Partial Matching for Non-Exact Query Matching

In realistic scenarios, there is a need to retrieve the query that does not present exactly in the spoken documents. However, the appeared instance of query might have the different suffix, prefix or word order. The DTW algorithm monotonically aligns the two sequences, and hence, the conventional approach is not suitable to perform partial matching between the frame sequence of query and test utterance. The proposed modified approach does not require to run Dynamic Time Warping (DTW) for multiple times for each query and test utterance pair [2]. The non-exact query matching can be handled by considering four different cases, namely, forward partial match, reverse partial match, the query containing filler and the query having reordered word sequence. In Chapter 5, we will discuss these partial matching strategies.

1.3.4 Feature Reduction Approach

To execute DTW in a faster mode for matching, the average of consecutive features is considered without overlapping [35]. However, the blind merging of features might average the information in the vicinity of phonetic boundaries. The reason can be the posterior features on either side of phone boundaries exhibit different characteristics. Hence, one such loss might be introduced due to the merging of feature vectors in the vicinity of phoneme boundaries. In feature reduction approach, we merged consecutive feature vectors within phonetic segment boundaries and executed DTW by reduced number of feature vectors. Thus, a lesser number of feature vectors reduces the computational cost by reducing the number of comparison operations with slight degradation in search performance. We will discuss the feature reduction-based search space reduction in Chapter 5.

1.3.5 Segment-Level Bag-of-Acoustic Words

The conventional QbE-STD systems are based on DTW, which is computationally intensive algorithm leading to scalability issue. The study discusses a segment-level approach of a novel Bag-of-Acoustic Words (BoAW) [2]. The objective is to use speech segmentation at phone-level to build an inverted index. This inverted index representation of spoken query and test utterance are used to reduce the search space. To restore the time information and perform query detection, DTW search is executed on selected test utterances. In Chapter 5, we will discuss segment-level BoAW for search space reduction.

1.3.6 Exploring Detection Sources and Multiple Acoustic Features

In addition, we discussed the two-stage zero-resource approach for QbE-STD [36] that exploits the detection candidates at first level and several acoustic features. At the first-stage, subDTW search algorithm with GP representation gives detection candidates. Then, several acoustic features and detection sources are used to re-score the detection hypothesis. The individual performance of various detectors was found to be complementary. In the similar framework, we exploited several detection sources, such as, the self-similarity matrix, the depth of valley along the warping path in DTW, Term Frequency (TF) and the weighted mean representation to improve the performance of Gaussian posteriorgram and phonetic posteriorgram [37]. These additional cues are complementary to the posteriorgram representation and hence, give better performance than posteriorgram alone.

1.4 Organization of the Thesis

The organization of thesis is as follows.

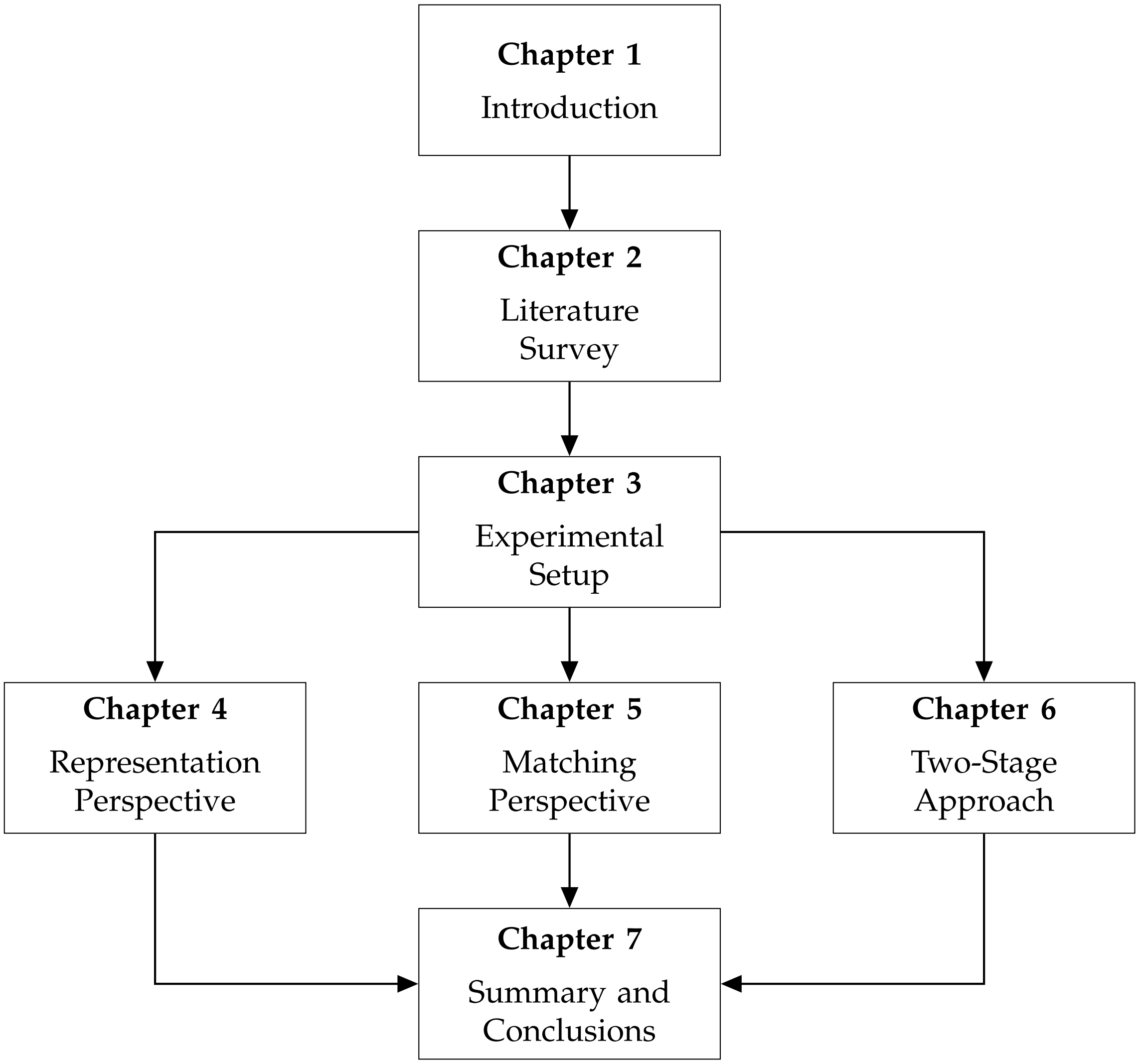


Figure 1.5: Flowchart of the organization of the thesis.

- Literature survey for QbE-STD problem is discussed in Chapter 2. In particular, we will discuss the representation, matching, and detection subsystems in detail. The primary performance evaluation metrics used in QbE-STD and various research issues are also presented. We will discuss several MediaEval SWS and QUESST QbE-STD systems in brief. However, detailed literature survey can be found in [7,22,24].
- The experimental setup used in this thesis is discussed in Chapter 3. In particular, this chapter gives details of databases used in the thesis, posteriorgram representation, search and detection sub-systems. In this chapter, we investigated

the effect of local constraints and dissimilarity functions used in subsequence DTW.

- Chapter 4 discusses representation perspective for QbE-STD system. In particular, VTL-warped Gaussian posteriorgram and posteriorgram of mixture of GMMs for QbE-STD are discussed. The objective is to design speaker-invariant speech representation that emphasizes the phonetic information rather the speaker-specific information.
- Chapter 5 discusses search (matching) perspective of QbE-STD system. In particular, partial matching approach for non-exact query matching and search space reduction approaches are presented. Feature reduction-based and segment-level BoAW approach are discussed as search space reduction approaches.
- Chapter 6 discusses the fusion of evidences from various acoustic features as well as detection sources. This chapter contains two parts, namely, two-stage zero-resource approach, which exploits several acoustic features and detection sources. We observed that the incorporating these detection sources and acoustic features improves the performance as compared to the performance of the posteriorgram alone.
- Finally, Chapter 7 discusses the summary of the thesis along with limitations and research directions that can be explored in the future.

1.5 Chapter Summary

This chapter presented an introduction to spoken content retrieval problem to retrieve audio documents. We briefly discussed spoken content retrieval systems, namely, STD, QbE-STD, and keyword spotting. In addition, we briefly discussed key contributions in this thesis, namely, GMM-based VTLN-warped Gaussian posteriorgram, posteriorgram from mixture of GMMs, feature reduction and segment-level BoAW for search space reduction, partial matching for non-exact query detection task, and the exploration of multiple detection sources and acoustic features. In the next chapter, the literature survey on QbE-STD problem is discussed.

CHAPTER 2
Literature Survey

2.1 Introduction

In Chapter 1, we briefly discussed different Spoken Content Retrieval (SCR) frameworks, namely, Spoken Term Detection (STD), Query-by-Example Spoken Term Detection (QbE-STD) and keyword spotting (KWS). In addition, we also discussed major contributions in the thesis. In this chapter, we will discuss a literature survey for various methods or approaches to design QbE-STD system. The organization of this chapter is as follows: Section 2.2 discusses the motivation behind QbE-STD and components of QbE-STD systems. Performance evaluation metrics are presented in Section 2.3. The details of components QbE-STD, such as, front-end subsystem, searching subsystem, and detection subsystem are presented in Section 2.4, 2.5, and 2.6, respectively. In Section 2.7, the research issues in QbE-STD systems are discussed. The brief details about QbE-STD submitted systems to MediaEval SWS and MediaEval QUESST evaluation campaign are given in Section 2.8.

2.2 Motivation and Components of QbE-STD

Query-by-Example (QbE) paradigm for STD has been introduced with two major motivations. One being the limitation of Automatic Speech Recognition (ASR) system for Out-of-Vocabulary (OOV) words as an alternative approach to subword recognizer [38]. The another is to extend the speech retrieval task for the low-resource scenarios, where ASR is not feasible due to lack of huge transcribed speech data [39]. Recent technological developments in the smartphone and digital devices allow a user to access the information via spoken media. This technological development motivated a new research direction for spoken content retrieval, i.e., to retrieve the audio document via spoken queries, which is called as Query-by-Example Spoken Term Detection (QbE-STD).

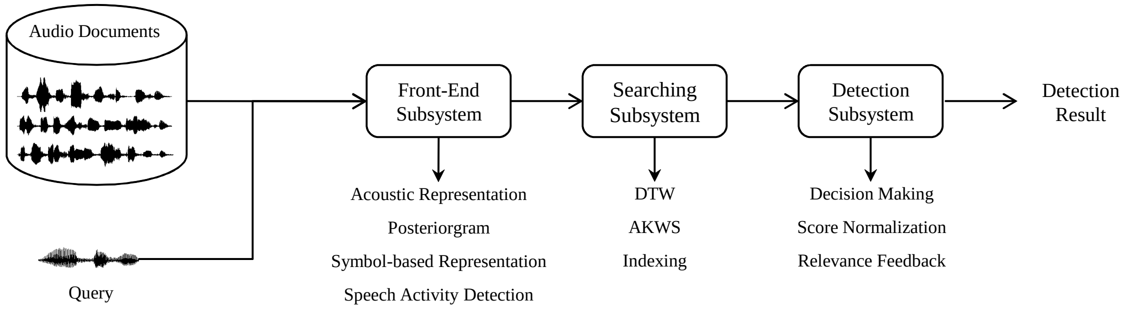


Figure 2.1: A schematic of generic block diagram of QbE-STD system. After [24].

Many information management tasks have adopted QbE paradigm. Earlier, the query-by-example paradigm was utilized to execute image retrieval task by computing the similarity between a query image and documents of images from database [40]. The QbE framework was also presented in [41] for music information retrieval (MIR) to extract the songs and other metadata, such as, singer, album, date of recording, etc. Recently, an example-based approach was employed for surgical activity detection task under context-based information extraction paradigm [42]. The stacked autoencoder representation of query was used with asymmetrical subsequence Dynamic Time Warping (DTW) search algorithm [42].

Due to diversified nature of QbE-STD, there are plenty of views and approaches regarding the solution. However, from the philosophical perspective, we may categorize the majority of the approaches into two broad categories, namely, low-resourced and zero-resourced, which are also referred to as supervised and unsupervised approaches, respectively. In low-resourced, acoustic models (from acoustically close language) can be used to obtain a symbolic form of acoustics. Further acoustic data adaptation and bootstrapping of an acoustic model can also be performed to tune the parameters of acoustic representation towards audio documents. In zero-resourced or unsupervised approach, no other resources from rich-resourced languages were adapted. These approaches are data-driven approaches, where the information about the acoustics is learned from the audio documents. To generalize these categories, we may consider the general schematic block diagram as shown in Figure 2.1. Basic components of QbE-STD system are as follows [24]:

(i) **Front-end subsystem:** The role of front-end subsystem is to represent audio documents and query into posteriorgram (i.e., frame-based posterior vectors) or symbol-based linguistic units. Front-end also perform speech *vs.* non-speech detection to remove silence present in the query.

(ii) **Search-subsystem:** The major role of search subsystem is to perform dynamic alignment between query and audio document realization to locate possible acoustic similarities. In addition, it also uses indexing to speed up the dynamic matching task.

(iii) **Detection subsystem:** This subsystem ranks the detection score and does the decision making. It also combines several evidences from multiple QbE-STD systems with effective score normalization and their score-level fusion. In pseudo-relevance feedback (PRF) scenario, the located query within the audio documents can be further utilized as the secondary query (i.e., pseudo-query). The acoustic representation from such pseudo-query can be further used to search within the audio documents, and the associated detection scores are modified, accordingly.

In STD framework for SCR, front-end subsystem corresponds to the ASR, whereas search and detection subsystems perform text-retrieval task onto the ASR output using text representation of a query to detect a query (keyword) (Please refer Figure 1.2). A more detailed aspect of each subsystem is discussed from next Section onwards. These research issues (which was discussed earlier in Chapter 1) are the acoustic representation, searching algorithm, search space reduction, search system combinations, etc.

2.3 Performance Evaluation Metrics

Evaluation of QbE-STD systems can be categorized as ranked and unranked evaluation [7]. The ranked evaluation displays the list of items based on the relevance w.r.t. query. The unranked evaluation assesses the performance based on thresholding, i.e., the audio documents are retrieved, whose relevance score is above a particular threshold. The performance of QbE-STD systems have been evaluated using following evaluation metrics [7,43]:

(a) **Precision@N**: If N is the number of queries present in the database (documents), precision@N, (i.e., p@N) is defined as [43]:

$$p@N = \frac{N_{corr}}{N} \times 100, \tag{2.1}$$

where N_{corr} = total number of correct occurrences found in top N items and N = total number of occurrences of the query. For an ideal search system, p@N should be *100* % indicating all top N_{corr} items are hit (correct detection).

Consider a scenario, where the query, q, appears 4 times in the documents and the top 10 potential matches are

$$(t, t, f, f, t, f, f, t, f, f),$$

where t and f corresponds to *true* occurrence and *false* alarm, respectively. Here, the total number of correct occurrences in top $N = 4$ items are 2 and hence, precision at N is, $p@N = \frac{2}{4} \times 100 = 50$ %.

(b) **Mean Average Precision (MAP):** Average precision (AP) is defined as [7]:

$$AP(q) = \frac{\sum_{k=1}^{n} p_q(k)\mathbb{1}_q(k)}{N}, \tag{2.2}$$

where $p_q(k)$ =precision at k for the query q, n = total number of audio documents, N = total number of occurrences of the query, p@k and $\mathbb{1}_q(k)$ = an *indicator* function indicating that the presence of a query (q) (i.e., relevant documents), which means $\mathbb{1}_q(k) = 1$ if k^{th} document is relevant otherwise $\mathbb{1}_q(k) = 0$. Mean Average Precision (MAP) is the mean value of average precision (AP) across different queries, i.e., $MAP = \frac{1}{Q}\sum_{q=1}^{Q} AP(q)$. Hence, AP shows the performance w.r.t. single query and mean average precision (MAP) is the mean value of AP across different queries indicating the overall system performance independent of queries. For the above scenario, $AP(q) = \frac{1}{4}\left(\frac{1}{1} + \frac{2}{2} + \frac{3}{5} + \frac{4}{8}\right) = 0.775$ (i.e., 77.50 %). For an ideal search system, MAP should be *100* %, indicating all the relevant documents are retrieved and none of the irrelevant documents are retrieved.

(c) **Maximum Term Weighted Value (MTWV):** Term Weighted Value (TWV) is a weighted linear combination of false alarm probability $P_{\text{fa}}(q, \theta)$ and miss probability $P_{\text{miss}}(q, \theta)$ at an operating threshold, θ, i.e.,

$$TWV(\theta) = 1 - \frac{1}{|Q|}\sum_{\forall q \in Q} \left(P_{\text{miss}}(q, \theta) + \beta P_{\text{fa}}(q, \theta)\right), \tag{2.3}$$

where Q is the set of queries and β is a constant (which is set empirically *66.66* for SWS 2013 and *12.49* for QUESST 2014 [29, 44]).

The Maximum Term Weighted Value (MTWV) is the maximum value of TWV obtained at an optimum value of threshold θ_{opt}. The ideal system should give *MTWV=1* that corresponds to $P_{\text{miss}} = 0$ and $P_{\text{fa}} = 0$, i.e., all the presence of query are detected with no false acceptance. The graphical view of miss, false alarm (FA) and hit are shown for a given query in Figure 2.2.

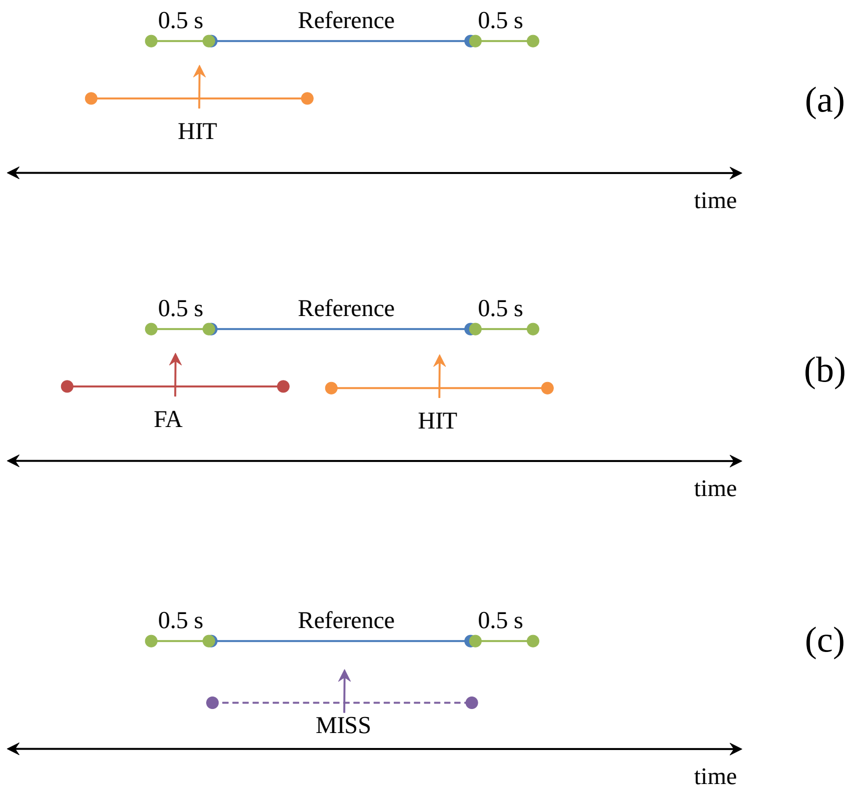


Figure 2.2: A graphical representation of hit, miss and false alarm (FA): (a) the detection within the ground truth (i.e., reference) that corresponds to *HIT*, (b) two detections within the ground truth that corresponds to one *HIT* and one *FA*, and (c) no detection within the ground truth corresponds to *MISS*. After [14,45,46].

(d) **Normalized cross-entropy $C_{\mathbf{nxe}}$:** It measures the fraction of information w.r.t. the ground truth, which is not provided by the system scores [44]. Scores are considered as log-likelihood ratios. A perfect system should give $C_{\text{nxe}} \approx 0$, indicating no randomness in the system detection and hence, it is well calibrated system, where target and non-target scores are far apart [47]. The cross-entropy can be expressed as follows:

$$C_{xe} = \frac{1}{\log 2}\left[\frac{p_{\text{target}}}{|T_{\text{true}}(S)|}\sum_{t\in T_{\text{true}}(S)} C_{\log}(llr_t) + \frac{1-p_{\text{target}}}{|T_{\text{false}}(S)|}\sum_{t\in T_{\text{false}}(S)} C_{\log}(llr_t)\right], \tag{2.4}$$

where $T_{\text{true}}(S)$ and $T_{\text{false}}(S)$ are the set of target and non-target trials, respectively and $C_{\log}(llr_t)$ is the logarithm of cost function. The empirical cross-entropy of trivial system (i.e., always rejecting or always accepting the trials, whichever gives the lower cost), which is prior entropy and it is given by [44]:

$$C_{xe}^{\text{prior}} = \frac{1}{\log 2}\left[p_{\text{target}}\cdot\log\frac{1}{p_{\text{target}}} + (1-p_{\text{target}})\cdot\log\frac{1}{(1-p_{\text{target}})}\right]. \tag{2.5}$$

From eq. (2.4) and eq. (2.5), the normalized cross-entropy is defined by [44]:

$$C_{nxe} = \frac{C_{xe}}{C_{xe}^{\text{prior}}}. \tag{2.6}$$

To minimize the normalized cross-entropy, affine transform is applied to the likelihood scores such that $l\hat{l}r = \gamma \cdot llr + \delta$, where γ and δ are the calibration parameters and minimum normalized cross-entropy is given by [44]:

$$C_{nxe}^{min} = \min_{\gamma,\delta}\{C_{nxe}\}. \tag{2.7}$$

(e) **Recall**: We use this metric to evaluate the performance of BoAW model. The recall is defined as [43]:

$$Recall = \frac{N_{rel \bigwedge ret}}{N_{rel}}, \tag{2.8}$$

where $N_{rel \bigwedge ret}$ is a total number of retrieved documents that are relevant to the query and N_{rel} is a total number of relevant documents. The ideal value of recall is *1* indicating all the retrieved documents are relevant.

(f) **Detection Error Trade-off (DET) curve:** The DET curve has been *de facto* standard performance evaluation metrics in speaker recognition literature [48]. The plot shows an trade-offs between false alarm probability (P_{fa}) and miss probability (P_{miss}) for various detection threshold. In STD and QbE-STD, the evaluation metric was introduced in [12], showing the performance of query detection task for each detection threshold.

2.4 Front-end Subsystem

The front-end subsystem is responsible for converting acoustic representation into either frame-based or symbol-based representation. Typically, a speech production knowledge is exploited to extract features from an acoustic realization of spoken audio. The acoustic realization is converted into parametric representation by signal-level feature extraction, which we refer to as *acoustic representation* [49]. The acoustic representation has some speaker-specific characteristics, which needs to be removed before searching an audio in multi-speaker scenarios. To address this issue, acoustic data is modeled and transformed into posterior representation. In order to avoid silence regions that are present in the spoken query, Speech Activity Detection (SAD) task is performed.

2.4.1 Acoustic Representation

In this thesis, we refer to acoustic representation as the first-level parameterization conventionally at short-time or at segment-level (within *20-30* ms interval). To characterize production and perception properties, linear prediction [50] and mel cepstrum [51], respectively, have been used as an acoustic representation. Next, the various acoustic representations explored in the QbE-STD problem are briefly discussed:

- Mel Frequency Cepstral Coefficients (MFCCs) [24, 52] : The human hearing mechanism has better frequency resolving capability at a lower frequency region than at a higher frequency region. To incorporate this into acoustic characteristics, nonlinearly-spaced subband filters (at mel frequency scale) are used. To extract MFCCs, Discrete Cosine Transform (DCT) is applied onto logarithm of subband energies (that are estimated at the output of mel filterbank having triangular shaped subband filters).

- Linear Prediction Cepstral Coefficients (LPCCs) [53]:
  The LPCCs are the cepstrum representation of Linear Prediction Coefficients (LPCs). LPC models the short-term power spectrum of speech using autoregressive all-pole model. Formants of speech acoustics are enhanced using all-pole model [50]. LPCs can capture the physiological characteristics of human speech production mechanism [54].

- Perceptual Linear Prediction (PLPs) [24,52]:
  In Linear Prediction (LP) analysis, we perform all-pole model spanning all possible frequency value. However, as discussed above human perception mechanism has better frequency resolution at lower frequency region than the higher frequency counterpart. Thus, human perception of hearing can discriminate lower frequencies better than the higher frequencies. In addition, hearing sensitivity is relatively more in middle frequency range (typically, 3100 Hz - 5000 Hz [55]). To incorporate a psychoacoustic observation of the human hearing process, PLP feature set was devised [55,56]. In general, the cepstral version of PLP is used to represent acoustic data from a short-time speech signal.

- Frequency-domain Linear Prediction (FDLP) [24,52]: The FDLP technique performs all-pole model in frequency-domain to obtain temporal envelopes. The DCT is used to get real-valued frequency-domain representation [57]. It was found that FDLP captures better acoustical characteristics under noisy and reverberation scenarios [58].

- Mel subband filters [59,60]: Subband filter energy has been used in the connections of DNN and Split Temporal Context Neural Network (STC-NN). Phoneme recognition system developed by the Brno University of Technology (BUT) use mel subband energy to train STC-NN for Czech (CZ), Hungarian (HU), Russian (RU) and English (EN) languages [61]. These phoneme recognizers have been extensively used in a low-resourced scenario for QbE-STD. The spacing of subband in frequency-domain is nonlinear, which typically follows mel frequency scale to imitate human perception process for hearing [51].

The features, such as, MFCCs, PLP, and FDLP, were used extensively as acoustic representation in QbE-STD system. This might be because of their wide usage in various speech research activities, and availability of various software tools. In addition to above mentioned acoustic representation, various other speech feature extraction schemes were used, such as, modulation spectrogram [62], modified group delay [34], MPEG-7 with low-level descriptors (such as, audio spectrum centroid, spectral flux, audio spectrum spread, audio spectrum envelope) [63], Power Normalized Cepstral Coefficients (PNCCs) [64], Local Binary Patterns (LBP) from spectrograms [65], etc. The design of suitable feature extraction scheme for QbE-STD task is an important research issue. The feature selection was performed using correlation and valley depth approaches [66,67]. In particular, the objective was to select the better features from the bunch of features (namely, a sum of auditory spectra, zero-crossing rate, frame intensity, loudness, Root Mean Square (RMS) energy, log-energy, MFCCs, mel filterbank, PLP, etc.) that discriminates two different words and hence, aids for audio search task.

2.4.2 Speech Activity Detection (SAD)

Speech Activity Detection (SAD) task is accomplished with the use of short-term energy. SAD is useful to remove the silence regions present at the begin and end of the spoken query. The top-hat algorithm is applied with window duration of *100* ms duration to avoid silence regions having less than *100* ms duration to prevent from small phrase breaks [68]. Speech *vs.* non-speech classification was performed by unsupervised clustering of MFCC features [28]. Zero-frequency filtered signal was used and a non-speech regions having more than *300* ms duration are ignored [69]. The Variance of Acceleration of MFCC (VAMFCC) rule-based approach was used to separate non-speech regions [70–72]. A Smith-trigger-based technique was used to trim start and end silence regions associated with the spoken query [73]. Multilayer Perceptron (MLP) was used to train the phone posteriors and posterior associated with non-speech region is used to discriminate speech *vs.* non-speech classification [59, 74–76]. To avoid false alarm, the query having less than *100* ms duration is ignored [74]. Principal Component Analysis (PCA) transformation is applied onto the speech segment, and then the threshold is applied on the corresponding eigenvalues of segmented speech to perform non-speech detection [71].

2.4.3 Posteriorgram Representation

The posteriorgram representation is *2*-dimensional (i.e., *2*-D) plots of time (speech frame index) *vs.* posterior probabilities. The posteriorgram is broadly categorized into two types, namely, supervised and unsupervised. The supervised posteriorgram are computed using the trained acoustic models using speech data and associated transcription, whereas the unsupervised posteriorgrams are computed without trained acoustic models using only speech data. For observations sequence, $\mathbf{O}$, having T acoustic feature vectors corresponding to segmental speech frames (namely, $\mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T$), the posteriorgram ($PG$) is defined as the sequence of posterior vectors [77]:

$$PG(\mathbf{O}) = [PG(\mathbf{o}_1), PG(\mathbf{o}_2), \cdots, PG(\mathbf{o}_T)]. \tag{2.9}$$

Each posterior vector can be obtained by [77]:

$$PG(\mathbf{o}_i) = \left[P(C_1|\mathbf{o}_i), P(C_2|\mathbf{o}_i), \cdots, P(C_{N_p}|\mathbf{o}_i)\right], \tag{2.10}$$

where C_j represents j^{th} class, N_p is the total number of classes and $PG(\mathbf{o}_i)$ is the posterior vector for i^{th} feature vector o_i. Hence, the posteriorgram for observation sequence, $\mathbf{O}$, can be represented as $N_p \times T$ matrix. Figure 2.3 shows an example of posteriorgram representation.

Here, class can be hypothesized as Gaussian components [77], acoustic segment [78], Restricted Boltzmann Machines (RBM) [79, 80] or phonetic unit [39] to compute posteriorgram. The posteriorgram representation takes real values between *0* and *1*, that corresponds to posterior probability (Please refer Figure 2.3). The acoustic representation, such as, MFCCs, PLPs, etc. are not sparse, taking multiple non-zero values for each speech frame.

2.4.3.1 Supervised Posteriorgram

Supervised posteriorgram represents an acoustic frame in terms of posterior probabilities w.r.t. phonetic units. For low-resourced scenarios, phoneme recogniz-

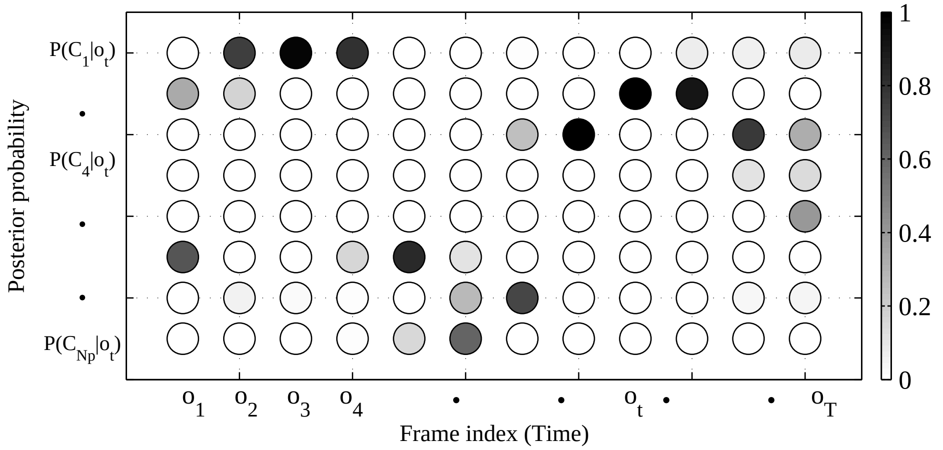
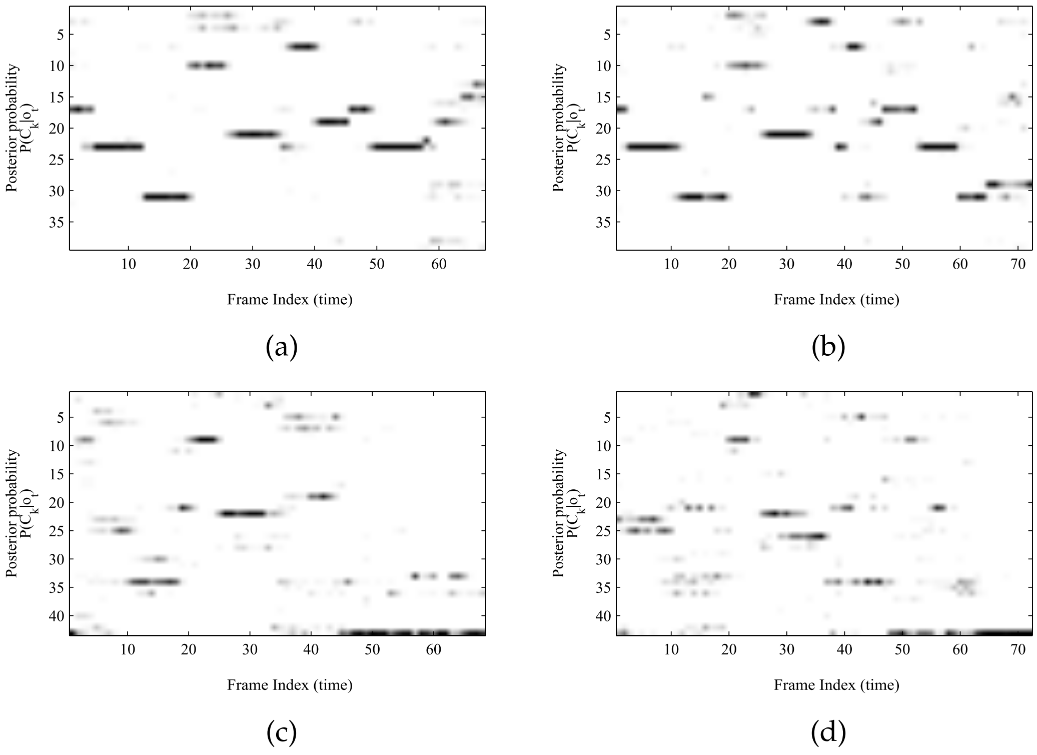


Figure 2.3: A schematic of posteriorgram plot.

ers from rich-resourced languages have been used to characterize phonetic symbols in terms of rich-resourced language. For low-resource languages, labeling of the speech data is the challenging and erroneous task. Hence, in order to use low-resource language speech, bootstrapping approach is employed as discussed in [24]. The trained phoneme recognizer is used to compute phonetic posteriorgram.

Posterior features were exploited in template-based ASR that was found to be comparable with HMM-based ASR. It was found that very less number of templates perform effectively in template-based ASR task [81]. The posteriorgram are found to be speaker-independent and different approaches were used to compute the posterior probability value in posteriorgram. For example, in supervised posteriorgram, Multilayer Perceptron (MLP) has been used to model the phonetic symbols [82]. Hence, the problem of acoustic representation design can be posed as the phonetic-level frame classification problem. However, this needs frame-level phonetic labels to train MLP. In low-resourced language scenarios, a language of phonetic recognizer might be different from that of audio documents. MLP can be trained with the help of rich-resourced language. It is assumed that this language is phonetically closer to that of audio documents and the query. Thus, for low-resourced speech sound units (that are represented in terms of MLP posteriors), speech need not be the part of the language (on which MLP is trained). In such cross-lingual decoding case, MLP representations can be assumed as language-independent [74, 82]. The phoneme recognizer developed by Brno University Technology (BUT) has been extensively used in the QbE-STD task [59,74]. These phoneme recognizers are based on the Split Temporal Context-Neural Netork (STC-NN), which is Multilayer Perceptron (MLP) framework. This framework exploits longer temporal context from the left and the right side and then merge them in MLP framework [61]. Figure 2.4 shows phonetic posteriorgram for word *intelligence* spoken by two different persons. In Figure 2.4 (a) and (b) show examples of phonetic posteriorgram with English phonetic recognizer. The language (as well as database, i.e., TIMIT) of spoken query and recognizer is the same. Thus, it can be observed that (a) and (b) posteriorgram plots are very much similar. Figure 2.4 (c) and (d) show cross-lingual posteriorgram, where the language of phonetic recognizer is different (i.e., Czech, (CZ)) than the language of spoken query (i.e., American English). Due to this language mismatch condition, the plots of posteriorgrams shown in Figure 2.4 (c) and (d) are not as identical as compared to Figure 2.4 (a) and (b). Thus, in low-resource scenario foreign language phoneme recognizer can be useful to represent the audio. The objective for using foreign language posteriorgram is to characterize the acoustic events of the audio. The objective is not to correlate this characterization into linguistic units, such as, phonemes. With this objective, many researchers have used cross-lingual foreign recognizer for QbE-STD [2,44,83,84].



(a)

(b)

(c)

(d)

Figure 2.4: Phonetic posteriorgrams examples for the spoken word '*intelligence*'. Posteriorgram with English phoneme recognizer: (a) male and (b) female speakers. Posteriorgram with Czech (CZ) phoneme recognizer (c) male and (d) female speakers.

Recently, articulatory features (from speech production viewpoint) that have an advantage of being language-independent were used for QbE-STD task [85,86].

The articulatory posteriorgram was computed by taking output as articulatory labels instead of phonetic labels. Three different MLPs were used to model vowel broad classes, place of articulation for consonants and manner of articulation for consonants inspired from IPA charts [86]. The suggested lower-dimensional articulatory posteriorgram gave an improvement of *2.87* % in terms of p@N as compared to the phonemic posteriorgram representation. In MLP construction, bottleneck features (BNF) from the middle hidden layers were also investigated for a QbE-STD task. The GP computed from articulatory BNF and FDLP features was found to be superior as posterior representation than the GP of FDLP features [85]. Furthermore, it was found that less amount of training data (about *30* minutes) for training BNF gave better performance than the phonetic posteriorgram and Gaussian posteriorgram, indicating convergence property of articulatory features [85]. In the cross-lingual QbE-STD framework, phonetic recognizer from one language is used to represent the spoken data of another language. The study conducted in [67] focuses the issue related to cross-lingual QbE-STD task. The phonetic posteriorgram trained in one language may not cover phonetically the target language (i.e., the language of audio documents and query). In such a scenario, few phonetic symbols might have redundant behavior, whereas few symbols might act as nuisance [67]. Thus, the study reported in [67] suggested feature selection approach by optimizing the DTW distance criteria and interestingly, observed that selected features gave better performance than the phoneme posteriorgram.

The study presented in [80] shows that deep architecture does not always help in performance improvement. In fact, the performance of QbE-STD saturates by using the only single hidden layer in DNN. Furthermore, it was observed that DNN with only *30* % of labels from TIMIT data could perform as good as entire training data [80]. The study was conducted on Tibetan corpus for QbE-STD using Chinese BNF [87]. The experimental results show *6* % improvement in F_1 score (F-score) over PLP features. The query detection was conducted using keyword model and garbage (background) model with GMM/HMM framework.

Knowledge-based information was used to design posteriorgram features [88]. The knowledge about speech sound units is integrated in terms of binary distinctive features. Earlier, distinctive features were found to be effective for event-based ASR task using landmark detection [89]. Each distinctive features can be trained using discriminative training approach. To that effect, support vector machines (SVMs) were utilized to perform binary classification for each distinctive features. This approach significantly improved the performance of QbE-STD as compared to other MLP phonetic posteriorgrams [88]. Recently, phone-time boundaries and data augmentation techniques were exploited in training of posteriorgram for QbE-STD [60,90]. The data augmentation technique produces a large amount of training data having acoustical variations obtained by convolving the speech signal with simulated room impulse response and additive noise [60]. The motivation behind data augmentation is to generate more the degraded speech in training phase in order to have a better match with degraded test conditions. The partial matching approaches suggested in [60] took phone boundaries into consideration as a guiding tool for the warping path of subDTW. Interestingly, data augmentation with stacked BNF does not hurt the performance rather do not give any substantial improvement either [90]. The reason might be QUESST 2014 does *not* have artificial noise and reverberation in audio data.

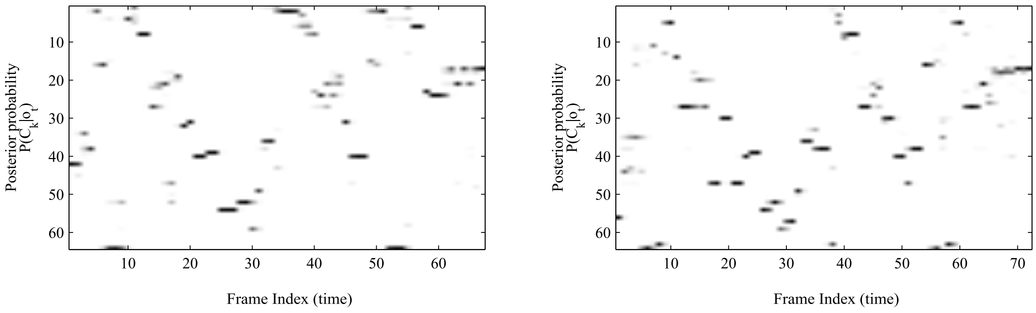
2.4.3.2 Unsupervised Posteriorgram

Unsupervised approaches for speech has been a popular area of research. In that context, the zero resource speech challenge 2015 (Zero-speech) was held at INTERSPEECH 2015 [91]. The primary objective of the challenge was to learn the subword units directly from a raw speech signal. The major motivation behind the task is to exploit the language acquisition performed by an infant. During the early infancy period, an infant gathers and process the speech and develop an acoustic and language model in an unsupervised manner [92]. Our traditional speech systems, in particular, ASR system uses a huge amount of acoustic and linguistic data. The researchers are keen to exploit this additional cues to improve the ASR performance. Unsupervised posteriorgram characterizes each feature vector by computing the posterior probability without any resources using features *only*. Such approaches are also referred to as zero-resources. Few unsupervised modeling techniques have been used, such as, Gaussian posteriorgram [25, 77], Acoustic Segment Model (ASM) [78, 93–96] posteriorgram and Restricted Boltzmann Machines (RBM) Posteriorgram [79,80] for QbE-STD. Next, we will describe these unsupervised posteriorgrams in brief.

- **Gaussian Posteriorgram:** Gaussian posteriorgram has been extensively used for QbE-STD task because it is easy to train and fewer parameters are required for tuning as compared to the ASM and RBM. Another advantage is that GMM can be easily adapted and initialize the deep learning networks, such as, Deep Belief Networks (DBN) [80]. The posterior probability $P(C_k|\mathbf{o}_t)$ (for k^{th} cluster and t^{th} speech frame index) of Gaussian Posteriorgram (GP) can be computed as follows:

$$\begin{aligned} P(C_k|\mathbf{o}_t) &= \frac{P(\mathbf{o}_t|C_k)P(C_k)}{P(\mathbf{o}_t)}, \qquad (2.11)\\ &= \frac{\pi_k \mathcal{N}(\mathbf{o}_t;\mu_k,\Sigma_k)}{\sum_{j=1}^{N_p} \pi_j \mathcal{N}(\mathbf{o}_t;\mu_j,\Sigma_j)}, \qquad (2.12)\end{aligned}$$

where N_p is the number of GMM components and $\{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{N_p}$ are the weights, mean vectors and covariance matrices, for each k GMM components. The weighted mean of Gaussian posteriorgram representation was explored for QbE-STD. A weighted mean representation is computed from a linear combination of mean vectors of GMM, whose weights are the posterior values associated with a posterior component. Smoothing operation on Gaussian posteriorgram was conducted to spread the posterior values across different components and frames. Reconstructed representation from smoothed Gaussian posteriorgram was found to be more efficient than the earlier weighted mean representation [97]. GMM posteriorgram-based QbE-STD was used for Telugu broadcast news dataset in [98]. Figure 2.5 shows phonetic posteriorgram for word *intelligence* spoken by two different persons. As compared to the phonetic posteriorgram (shown in Figure 2.4 (a) and (b)), the Gaussian posteriorgram (shown in Figure 2.5) is not as identical as phonetic posteriorgram because it may be due to unsupervised nature of GMM, i.e., obtained without any supervision (or transcription).



(a) (b)

Figure 2.5: Gaussian posteriorgrams examples for the spoken word '*intelligence*'. (a) male speaker and (b) female speaker.

The binary version of posteriorgram, which *BinaryGrams*, were used in zero-resource audio matching [68, 99]. In this representation, posterior probabilities are mapped to the binary values. The advantage of BinaryGrams representation is lesser storage and computational requirements employing Boolean logical distance computation [68, 99]. The human vocal tract system can be assumed to have the cascaded connections of linear cylindrical-shaped acoustic tubes or organ pipes (which is nothing but resonators corresponding to the formants). The Bessel functions are the solutions of the cylindrical wave equation. With that motivation, Fourier-Bessel Cepstral Coefficients (FBCCs) were used for QbE-STD [100]. FBCCs are expected to give better speech representation than the MFCCs (where frequency-domain is characterized by sinusoidal basis functions). FBCC gave *7* % (i.e., *0.07*) absolute improvement than the MFCCs for SWS 2012 QbE-STD task using Gaussian posteriorgram [100].

The spectral and temporal acoustic models were linked for QbE-STD task [101, 102]. In particular, GMM of the spectral acoustic model is initialized with TIMIT phonetic ground truth. *39* phones of TIMIT were modeled as *4*-component GMM, resulting in *156*-dimensional posterior representation. The Kullback-Leibler (KL) divergence w.r.t. each Gaussian of GMM was used to measure the dissimilarity between posterior vectors. The temporal acoustic model (TAM) captures the long temporal information of duration *150* ms that was taken from MFCCs representation. Combined spectral and temporal acoustic model gave *69* % improvement in MTWV as compared to the baseline approach.

Several modifications in GMM were proposed during training and effectiveness were studied for a QbE-STD task. Expectation Maximization-Maximum Likelihood (EM-ML)-based approach was used to train GMM. In this study, K-means is trained (which creates hard assignment) and then Gaussian distribution is fitted onto each cluster [99]. For better initialization, the GMM-based acoustic model trained on high-resourced language was adapted for training low-resourced language [101]. MLP is trained with either articulatory classes or phoneme labels and the BNF from the middle layer were used. The BNF with articulatory class labels at output were used with acoustic features to train GMM, and the GP derived posteriorgram outperformed phonetic posteriorgram [85]. Intrinsic Spectral Analysis (ISA) has been found to give speaker-invariant as well as phonetically distinctive representation in unsupervised method [103, 104]. The experimental results on TIMIT QbE-STD show that ISA features gave the relative improvement of *13.5* % over Gaussian posteriorgram, when temporal information is used in ISA [103].

- **Acoustic Segment Model (ASM):** The unsupervised model developed using GMM does not incorporate a *sequential* information into the account. To incorporate temporal dynamics across the frames, ASM was proposed, where segmentation and labelling-based approach was employed. ASM characterizes the temporal dynamics via HMM framework [78, 93, 95, 96, 105]. Figure 2.6

shows the steps for computing the ASM. To initiate HMM training, segmentation, and label assignment tasks are performed by Hierarchical Agglomerative Clustering (HAC) algorithm [106]. To label each acoustic segment, K-means clustering [78,93,105], Gaussian Component Clustering (GCC) and Multi-view Segment Clustering (MSC) were explored [96]. The initial labels are used to train ASM and decoding new labels. Again, the ASM is trained with this newly generated labels and this process iterates till convergence in label assignment is achieved. Conventionally, a number of ASM clusters were selected based on tuning on development set [78]. However, Minimum Description Length (MDL) criteria was used along with likelihood to select an optimum number of ASM components [93].

In unsupervised ASM framework, few studies discussed multi-level acoustic, phonetic and temporal granularity (resolution) [107,108]. In these studies, phonetic, acoustic and temporal resolutions were adjusted by varying number of classes (i.e., ASM units), number of Gaussian components and number of states in HMM. To perform QbE-STD, KL-divergence between the states of HMM (in terms of variational approximation) was used and DTW was used to align the sequence [108]. Furthermore, the content consistency across the multiple recognized hypothesis was inferred and re-labelling was conducted. This re-labelling scheme was found to be more effective than the one without re-labelling [107].

- **Restricted Boltzmann Machine (RBM)**:
RBM was explored to represent the audio signal in QbE-STD. Gaussian-Bernoulli RBM (GBRBM) was used to learn the distribution of feature vectors [79]. The joint energy for GBRBM is defined as follows [79]:

$$E(\mathbf{v},\mathbf{h}) = \sum_{i \in V} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i \in V, j \in H} \frac{v_i}{\sigma_i} w_{ij} h_j - \sum_{j \in H} h_j c_j, \tag{2.13}$$

where v_i and h_j represent the i^{th} component of visible layer and j^{th} component of hidden layer, respectively, w_{ij} is the weight associated with visible unit v_i and hidden unit h_j, b_i is the bias with visible unit v_i, c_j is the bias with hidden unit h_j and σ_i is the standard deviation with visible unit v_i.

The sigmoid activation obtained at first hidden layer was used as a posterior

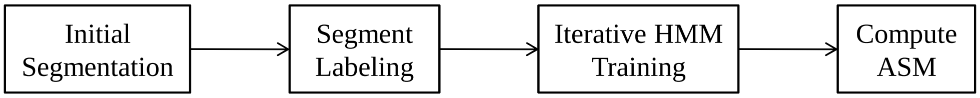


Figure 2.6: The schematic diagram for ASM computation. After [78,96].

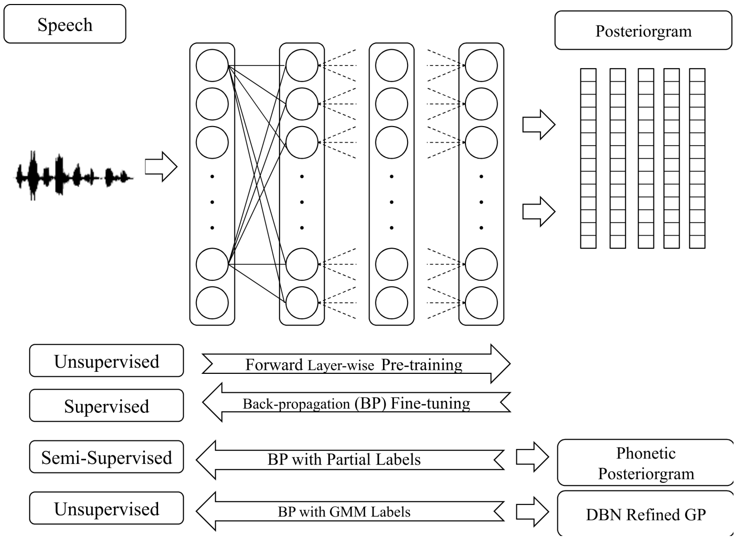


Figure 2.7: The schematic flow diagram of DBN posteriorgram feature extraction. After [109].

representation. The major motivation behind GBRBM was the non-Gaussian distribution of features. In GP, we generally assume diagonal covariance, which may be not the correct for different acoustic features. Since GBRBM posteriorgram considers this fact, it is expected to give better performance than the GP. The GBRBM posteriorgram gave *12* % improvement in p@N as compared to the MFCC representation. Deep Belief Network (DBN), the stack of RBMs, was used for posteriorgram building [80, 109]. DBN posteriorgram were used to tune Gaussian posteriorgram by assigning a class to each feature-based on the values attained by Gaussian posteriorgram. This unsupervised framework improves the search performance over Gaussian posteriorgram representation [80]. Procedure of DBN posteriorgram feature extraction is shown in Figure 2.7.

Recently, an autoencoder-based approach was used to learn the subword units in the speech signal. The autoencoder-based representation was found to be more efficient than the GMM posteriorgram for spoken query classification task [110]. Recently, Dirichlet Process Gaussian Mixture Model (DPGMM) is used in the connection with DNN for audio representation [111]. DPGMM has an ability to learn and adapt the suitable number of hyperparameters rather than being fixed as in the case of GMM. DPGMM is used to label the speech data in an unsupervised manner. The low-dimensional BNFs were computed in DNN framework. The performance of unsupervised BNF was found to be comparable with supervised BNF. The score-level fusion further improved *10* % relative improvement than the supervised BNF.

2.4.4 Symbol-based Representation

Along with frame-level posteriorgram representation, symbol-level representation was also used in QbE-STD. Phoneme decoders from trained recognizer [69, 112, 113], or unsupervised ASM trained under zero-resource scenarios [93, 108] were used to produce symbolic representation. Well trained phoneme recognizer and international phonetic alphabets (IPA) were used to represent spoken documents and query [114]. The phoneme labels obtained by Romanian phoneme recognition are mapped to the respective IPA symbols. Dynamic Time Warping String Search (DTW-SS) approach performs alignment on the smaller part of test utterance and query. DTW-SS is operated on symbol-level, and hence, it provides scalable and practically feasible in terms of computational aspect as compared to the frame-based approaches.

Given a spoken data, the unsupervised HMM training and labeling task can be executed in an iterative manner. Initially, the spoken data are transcribed as W_0 using segmentation and labeling approach [115]. The iterative model training and decoding steps are executed as follows:

$$\lambda_t = \underset{\lambda}{\operatorname{argmax}}\, P(O|\lambda, W_{t-1}), \tag{2.14}$$

$$W_t = \underset{W}{\operatorname{argmax}}\, P(O|\lambda_t, W). \tag{2.15}$$

HMM models are trained using the initial transcripts (labels) W_0 as in eq. (2.14) and then new transcription (labels) are obtained using trained HMM as in eq. (2.15). For every time t, the process can be repeated (as in eq. (2.14) and eq. (2.15)) iteratively, till there is no change across the old transcript W_{t-1} and the new transcript W_t.

In this Section, we studied front-end subsystem of QbE-STD system, where each audio waveform is converted into equivalent frame-based posterior representation. This sequence of posterior representation (i.e., posteriorgram) is used to execute matching between the audio documents and the spoken query. In the next Section, we discuss variants of search algorithm to perform audio matching.

2.5 Searching Subsystem

In this Section, we will discuss the search algorithm used for QbE-STD task for frame-based and symbol-based representation.

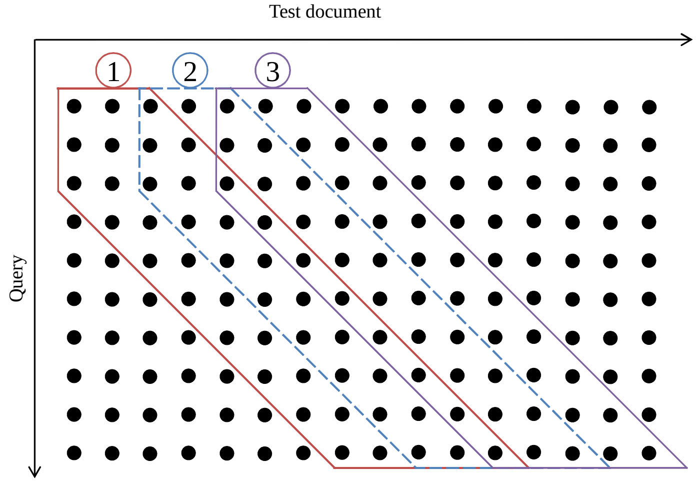


Figure 2.8: An example of segmental DTW with adjustment window length 2. Three segments are shown. After [25,77,109,119].

2.5.1 Search Subsystem for Frame-based Representation

The same query word spoken by the same or different speaker does have temporal alignment mismatch. Due to the speaking rate variation of individual speakers (i.e., the speakers may or may not prolong few vowels in any word), linear time normalization approach may not be suitable for the alignment task. DP-based nonlinear time normalization scheme was proposed in [116]. This nonlinear time-normalization technique has been popular as DTW. This Section discusses the majority of search algorithms used in QbE-STD problem, which are some variants of DTW.

(A) Segmental DTW (SDTW) is popular search technique, proposed for spoken term discovery [117]. In SDTW, constraints applied by considering the segment of nearly equal length to the query. DTW is performed to obtain an alignment cost for each segment [77, 117]. The number of comparison operation to execute the search task for a query having length N frames and test utterance having length M frames is of the order of $\mathcal{O}(MN^2)$, which is very time-complex operation. The number of segments in SDTW are proportional to number of test utterance. In some cases, a warping window was set in proportion to the length of query [118]. Figure 2.8 shows three segments of SDTW.

(B) Subsequence DTW (SubDTW) is used to find a subsequence of a query into reference. Here, endpoint constraint of classical DTW is relaxed [120]. In Sub-DTW, local distance is accumulated only once during warping path construction and hence, single DTW is executed (unlike multiple DTW in SDTW algorithm). The number of comparison operation to execute the search task for the query having length N frames and reference having length M frames is of the order of $\mathcal{O}(MN)$ that is far lesser than $\mathcal{O}(MN^2)$. Few modifications in the SubDTW algorithm were suggested mainly due to accumulation distance procedure.

(C) Other variants of DTW: SDTW and subDTW have been extensively used in QbE-STD, however, few research studies considered different variants of DTW w.r.t different local constraints or distance computation [53, 121, 122]. Non-segmental DTW (NSDTW) accumulates the distances and then normalized by a total number of hops to reach a current position from the starting frame index [53]. Slope-constraint DTW (SC-DTW) was proposed to overcome high-speaking rate distortion between two audio signals [121]. In SC-DTW, local slope constraints are adjusted between *0.5* and *2*, i.e., the warping path is adjusted such that no frame from query should be aligned to more than two frames of test utterance and vice-versa [25, 121]. Cumulative Dynamic Time Warping (CDTW) was proposed to introduce softmax operation in distance accumulation [122]. The detection score is obtained through logistic function. The parameters of logistic function can be derived using gradient-based optimization, where development data is used as train data.

2.5.1.1 Computational Improvement for Frame-based Approaches

The major issue with DTW-based approaches is high computational requirements. The simple, straightforward solution is to use advanced computing platform and parallel computing environment. Several such attempts were made in terms of the use of distributing the computational load to multiple CPU cores [123] for acoustic pattern discovery as well as the use of Graphical Processing Units (GPUs) [119, 124] for QbE-STD. In order to speed up, various improvements were suggested, which are as follows:

- **Lower bound estimate:** The lower bound (LB) is used to approximate LB of the exact DTW [125]. For the query Q and the segment of utterance U_S, the LB and DTW are related as follows:

$$LB(Q, U_S) \leq DTW(Q, U_S). \tag{2.16}$$

  The idea is to execute computationally cheaper LB values for each possible segment of an utterance, before the execution of DTW. The pair of a query and a test segment along with the LB values are ranked and stored in the queue, PQ. The top of the queue is pointing to the lowest LB value for all possible segments. The algorithm popped off the segments $U_{S_{top}}$ on the top of queue and

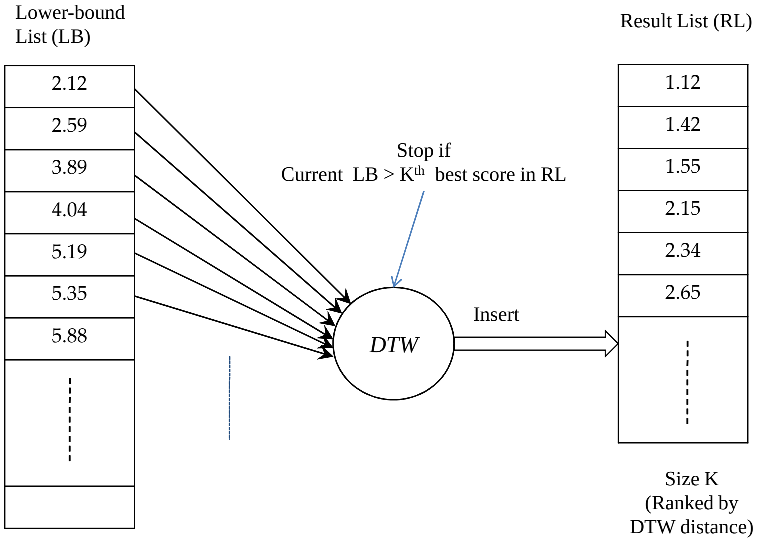


Figure 2.9: An illustration of KNN search with LB estimate to speedup SDTW. The LHS indicates the priority queue PQ and RHS indicates the result list RL. After [109].

execute DTW, i.e., $DTW(Q, U_{S_{top}})$. The result of DTW is stored into the result list RL. If result list RL does not contain the utterance associated with segment $U_{S_{top}}$, then RL is updated, else the better segment having lower DTW is kept in RL. The kth best match is set to $DTW(Q, U_{S_{top}})$. Again, the top of the queue PQ is popped off and compared against the kth best and the list RL is updated. Figure 2.9 shows the working of K nearest neighbors (KNN) search with LB estimate. Thus, KNN DTW method uses the lower bound technique to remove the unnecessary DTW operations [126]. In addition, tighter lower bound for DTW was also proposed in [127] that can be used for any distance unlike inner product distance as proposed in [126].

- **Segment-based approach:** In this approach, a segment is represented as the sequence of homogeneous frames efficiently. Thus, it can reduce the comparison operation and speed up the searching task at little cost [121]. The integrated segment and frame-based approach are presented from QbE-STD [121] that performs in two passes to detect the query. The first pass performs segment-based DTW and located the possible region that hypothesizes the presence of query. The distance between possible regions and query is recalculated using frame-level DTW. The detection performance is improved by *2.0* % in terms of MAP and CPU time is reduced by *45.2* % as compared to the frame-based DTW [121].

- **Syllable-based segmentation:** The study conducted in [128] presents the fast approach of QbE-STD via two-pass approach. At the first pass, new query instances derived using the actual query. In the second pass, the new instances along with actual query are used and normalized DTW distances were computed. The DTW alignment distance between syllable segments with a query and part of test utterance were used in the normalization process. The query detection task was conducted by segmental DTW, where the search is executed at only syllable boundaries and hence, it reduces the search computational complexity by a factor of nine than the segmental DTW approaches [128].

- **Fast DTW:** Inspiring from fast DTW approach [35], the study was conducted to speed up the NS-DTW by feature reduction [53]. The number of comparison operations to execute the search task for the query (having length N frames) and reference (having length M frames) is of the order of $\mathcal{O}(\frac{MN}{\beta^2})$, where $\beta > 1$ is the feature reduction factor. This means theoretically, search complexity reduces in a quadratic manner. Unbounded-DTW was proposed in [129] to reduce high computational cost. In this approach, possible alignment points are defined that are used to search time-warped matches. This results in a reduction in the exhaustive computational matrix. Start-end alignment points are found using forward-backward DP algorithm.
- **Information Retrieval DTW (IR-DTW)**: IR-DTW was proposed to match two subsequences of test utterance and spoken query in QbE-STD [130]. IR-DTW does not involve with dynamic alignment; it sequentially computes the similarity measure on an entire distance matrix. Hence, IR-DTW has a smaller memory or footprint size requirement as compared to the subDTW. In addition, IR-DTW offers an indexing for searching that makes this approach feasible for large scale subsequence matching scenario. One of the limitations of IR-DTW was the requirement of exhaustive search on all the test utterances for a given query.
- **Randomized Algorithm for Fast Template Matching:** For computational efficient DTW, randomized algorithms are used. Locality Sensitivity Hashing (LSH) technique is used to approximate sparse cosine similarity [131]. LSH converts high-dimensional raw speech representation into low-dimensional signature bits that can exploit Hamming distance as an approximating to some distance metrics in the feature space. A Randomized Acoustic Indexing and Logarithmic-time Search (RAILS) algorithm is rooted in randomized hashing and nearest neighborhood search operation [132]. The RAILS makes large-scale QbE-STD task feasible by combining stages of search-space reduction using approximating distance and an application of segmental DTW for further rescoring [132].
- **Bag of Acoustic Words Approach (BoAW) :** To reduce the search space, BoAW

is proposed in [133]. This method is derived from Bag of Words (BoW) method used in the text retrieval literature [43, 134]. In this study, acoustic words are associated with components of Gaussian posteriorgram. In BoAW, a spoken document is represented as an unordered acoustical units, such as, framewise phonetic contents, syllables or spoken lexical words. Each document is represented as a histogram or frequency count of acoustic words. Acoustic words are indexed using inverted indexing method [133]. In the BoAW method, temporal information in the speech signal is lost. Hence, DTW-based re-scoring is applied on the detection of BoAW inverted indexing method. The idea of BoAW was extended for bi-gram and tri-gram for indexing [135].

- **Graph-based similarity search :** To speed up the QbE-STD, several Graph-based Similarity Search (GSS) approaches were proposed, where the indexing graph is prepared *offline* and query detection is performed on the selected utterances [136–138]. A Degree Reduced k - nearest neighbor (k-DR) graphs are constructed from the GP representation. The GSS is a combination of Greedy Search (GS) and Depth-First Search (DFS). The GSS looks for the graph index, which is similar to the query and generates selected detection candidates with scores [136]. The DTW is further performed on these selected candidates. The GSS approach executes the search space reduction by *28* times faster than the LB-based approach, keeping almost the same p@N [136]. The extension of k-DR, i.e., hierarchical k-DR was proposed to perform Hierarchical Graph-based Similarity Search (HGSS). The HGSS reduces the CPU execution time about 40 % than the GSS, having almost the same precision [137]. A Double-Layer neighborhood Graph (DLG) index search method was proposed, which has two distinct layers, namely, an upper layer (which is analogous to express highway road) and a base layer (which is analogous to general roads). Both layers have different degree reduced k- nearest neighbor. Thus, the vertex on a base layer are posteriorgram segments that acts as general roads, and the top layer has a representative vertex that serves as an express highway. DLG reduces the CPU execution time by 40 % and more than 60 % as compared to the HGSS and GSS, respectively [138].

2.5.2 Search Subsystem for Symbol-based Representation

Symbol-based approach converts the query and spoken document in terms of a sequence of symbols. To do this, low-resourced approaches are employed to obtain phoneme symbols [139] or articulatory labels [69] from spoken document and query. However, few studies used zero-resourced approaches and exploited ASM to obtain symbols [93, 108]. There are two major approaches, namely, DTW on string and Viterbi alignment.

- **DTW :** Similar to frame-based approaches, DTW was also explored on the symbol-based representation. Various approaches used a different number of phone sets in order to handle the language mismatch between trained recognizer and the audio data. In addition, hyperphone, i.e., broad phoneme symbol-based approach was used as an indexing and searching [18]. The cost of alignment between the symbols were based on either flat binary [69], data-driven (i.e., derived from confusion matrix [107, 140]) or linguistically-motivated [18, 112]. In data-driven approaches, phoneme confusion matrix is normalized and cost of insertion, substitution and deletion are computed [140]. To compute the dissimilarities between the two symbols, KL-divergence averaged over all the states of HMM was used in [107]. To compensate the phoneme (symbol) decoding errors, N-best or lattice were used. However, this technique introduces speed *vs.* performance trade-offs.
- **Viterbi Decoding :** A keyword-background model-based AKWS approach has been popular for query detection task [33]. In AKWS model, each query is modeled as HMM (i.e., query HMM). Then, the likelihood difference between two HMMs is considered, namely, staying in background HMM and passing through query HMM sequence. However, this technique was designed for a pre-defined set of keywords (queries). In order to use this approach in spoken query, phoneme sequence from each query is generated, and then likelihood difference is calculated [30, 113]. Under ASM framework, entire audio is converted into ASM states, and a Viterbi algorithm is applied to compute the alignment cost. Furthermore, to introduce the slope-constraint in query alignment, Duration Constraint Viterbi (DC-Vite) was proposed [93]. In DC-Vite, the alignment of frames to HMM state is restricted by a number of frames per state, which is analogous to SC-DTW at frame-level matching [25]. The unsupervised HMM consists *3*-degrees of freedom, namely, a number of models, a number of states in each model, and a number of Gaussians in each state. They are referred to as phonetic granularity, temporal granularity, and acoustic granularity, respectively, in the study presented in [108]. Unsupervised HMM converts audio documents and spoken query into a sequence of unsupervised HMM labels. For TIMIT QbE-STD, this approach gave *16.16* % improvement in MAP as compared to the DTW-based approach.
- **Weighted Finite State Transducer (WFST) :** The lattice representation can be effectively transformed into WFST, and query audio is represented in terms of

Weighted Finite State Accepter [23]. WFST in QbE framework usually converts the phonetic symbols of lattice into the transducer, where time information is associated with a vertex of the graph and the likelihood is associated with a link. Later the weight pushing algorithm transforms the weights (likelihoods) into posterior probability [141]. The query detection is performed by composing the query transducer with utterance transducer. The resulting transducer gives all the possible occurrences of the query within utterance [142].

2.6 Detection Subsystem

In this Section, we will discuss the detection subsystem that performs decision making from search algorithm. In addition, the detection subsystem combines evidences from multiple sources, i.e., multiple query realization and multiple subsystems.

2.6.1 Score Normalization

The scores associated with each query detection might be different due to length and the content in the query. The QbE-STD system consists of single or multiword queries, as well as the presence of silence is different in each query. For these reasons, the scores from each query have different distributions [24]. Typically, DTW algorithm accumulates the distances between frame sequences and the scores taken from DTW distances follow Gaussian distribution having different means and variances. In order to consider a common threshold for the scores corresponding to all the queries, the scores per query were normalized to have zero-mean and unity-variance (*q*-norm). The concept of *q-norm* has been applied in many SWS systems [76, 118, 143, 144]. Query normalization problem was formulated as the logistic function, and the parameters were estimated using development set in [122]. Novel *m-norm* was proposed, where a maximum (mode) score is subtracted, and a standard deviation is divided for each query [113].

2.6.2 Query Selection

The selection of a query out of multiple examples of a query (i.e., query realization) has been very important. The query selection and combination of multiple examples were analyzed in [59]. Various strategies have been investigated to effectively select the example. Each spoken examples may exhibit different temporal dynamics. The sharpness in phonetic posteriorgram can be quantified in terms of *self* dot products and cross-entropy. In addition, to incorporate the relative behavior of query example w.r.t. other examples, pairwise DTW can be employed. After ranking, queries are fused with each other by the alignment. Multiple examples are fused in terms of graphical keyword model in ergodic HMM (EHMM) framework [144].

Posteriorgram stability, reliability, and local similarity were found to be effective metrics to select the best query [60]. It was analyzed that selected query via these metrics gave better precision over the score-level fusion and alignment over the longest query. The query selection schemes used in [59] and [60] were designed for phonetic posteriorgram. However, the longest query alignment method proposed in [74] applies to all the kinds of representation. The computationally simple scheme was suggested as to consider the longest query [74]. The average query representation is formed by the alignment of other queries onto the longest query. This technique is computationally cheaper because only single average example is utilized for searching task. Here, the longer duration query is used as a base and each feature from other query are aligned onto the base query to produce an average query. The average query X_{avg} for two different spoken queries $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{L_X}]$ and $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{L_Y}]$ can be performed as follows [74]:

$$X_t^{avg} = \frac{1}{1 + |\mathcal{S}_t|} \left(X_t + \sum_{v \in \mathcal{S}_t} v \right), \tag{2.17}$$

where $\mathcal{S}_t$ is the set of all the features of Y that are aligned to t^{th} frame of X (i.e., $\mathbf{x}_t$) and $1 \leq t \leq L_X$. In this thesis, we will use this approach for constructing the single average query example (please refer sub-Section 4.2.5.3).

2.6.3 Pseudo Relevance Feedback (PRF)

Inspired from the Information Retrieval (IR) literature, several studies in QbE-STD literature exploited the concept of relevance feedback. In relevance feedback scenario, the detection candidates (i.e., the part of spoken audio detected) at first few hits are assumed to be close to the query. Hence, these detected part of spoken audio may be treated as a query (which is referred to as *pseudo-query* or *pseudo-relevant example*) and the searching can be employed with this query. This approach is known as pseudo-relevance feedback (PRF) [93,121,145]. The scores obtained using pseudo query are merged with the scores using actual query. However, the weights associated with the pseudo query is low as compared to the weight of query because pseudo query might be the false detection (i.e., wrong detection by the actual query). Thus, lower weight associated with pseudo query might not affect the performance severely. Figure 2.10 illustrates the concept of pseudo-relevance feedback perform re-scoring for detection candidates. From Figure 2.10, it can be seen that top N ranked detection candidates are treated as pseudo-query examples. Since lower ranked pseudo query example is highly similar to the query than the higher ranked pseudo query. All the DTW distances between pseudo queries and detection candidates are used for re-scoring. This is achieved by taking the linear combination of DTW distances. However, the relevance of higher ranked pseudo query is less, the DTW distance should be assign lower weights. In Figure 2.10, the weights are set as $w_0 > w_1 > \cdots > w_N$. More details of pseudo-relevance feedback is given in [25]. We will use pseudo-relevant queries to re-score the detection hypothesis (please refer sub-Section 6.3.4).

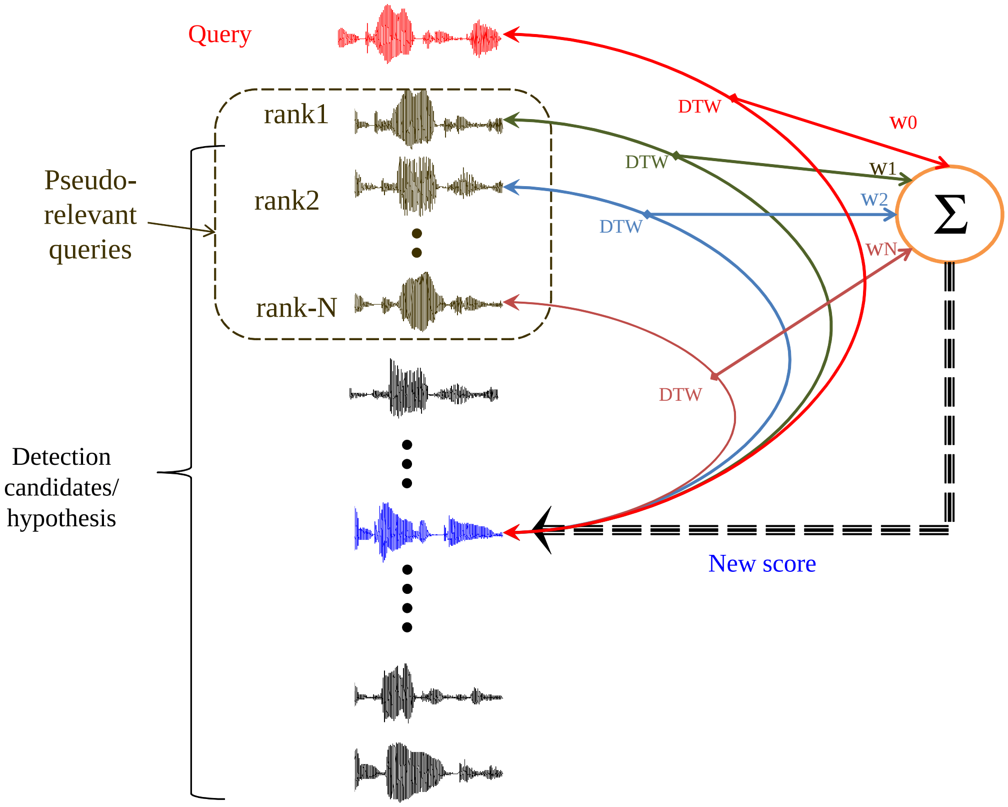


Figure 2.10: Illustration of re-scoring using the pseudo relevance feedback, top N detection candidates are treated as pseudo query examples. The relevance factors are $w_0, w_1, w_2, \cdots, w_N$, where $w_i > w_j, \forall i < j$. After [25].

PRF can be employed as a second pass search system along with first pass, such as, segment-based search system, whose execution is faster than the DTW [93, 121]. In addition to a pseudo-relevance example, the concept of pseudo-irrelevant example was used for ASM framework [93]. The pseudo-irrelevant example corresponds to the detection, which has the least similarity to the query. In this study, query HMM (i.e., Λ_R) and anti-query HMM (i.e., Λ_I) were built using the set of pseudo-relevant examples R and pseudo-irrelevant examples I. The likelihood-ratio w.r.t. two models were computed in order to score the hypothesis. Self-Similarity Matrix (SSM) along with DTW scores were combined to improve DTW detection score [146]. DTW compares the effective distance to align the frame sequence, whereas SSM takes into account the distance from entire spoken segments of the query and detected part in the test audio. In model-based approach, relevant or irrelevant examples are used to model the query and anti-query HMMs, respectively. An absolute improvement of *11.8* % was obtained with almost *50* % reduction in computational cost as compared to the SDTW on Mandarin broadcast news corpus [93, 105]. For query detection task, image processing-based approach was used in [147]. In this approach, distance matrix of a query and test utterance is processed with series of morphological operations.

2.6.4 Non-Exact Query Matching

MediaEval QUESST 2014 dataset contains different types of query that has variations at suffix, prefix, or word order. We performed modified DTW search to locate such presence of a query in the audio document. The kind of variations in spoken query retrieval task requires one to perform approximate/partial matching search rather than only the exact search. Given that *no prior* information about query type is given with QUESST 2014 database, the search algorithm should be general and incorporate all possible variations in the query matching task. The symbol-based search approach was adopted, and split phone sequence from query decoding was used to perform partial matching [60, 148]. In order to detect a non-exact type of query (i.e., variations at suffix, prefix, word order), DTW algorithm needs to be modified [83]. This technique considers different partial match and takes harmonic mean of DTW distance from these partial match. We considered slightly modified version from [83] with less number of backtracking. In the next part of chapter, we address the issue of computation related to search task. To that effect, we suggest two approaches, namely, feature reduction and segment-level BoAW model.

2.6.5 Calibration and Fusion of Multiple Search Systems

The discriminative calibration using logistic regression has been extensively used to fuse various heterogeneous QbE-STD systems [62]. To improve the fusion score and provide better calibration additional resources (such as, length of a query, non-silence frames, language identification (LID) scores) were used [149]. Given multiple examples of the same query, a reliable warping path on the test utterance is evaluated, where most of the minimum DTW warping path overlaps. The DTW scores from such reliable warping paths are taken into fusion procedures, and the rests are discarded [77].

Parallel DTW distance matrix fusion was proposed to combine the various frame-level representation [150]. In this distance matrix fusion, DTW has executed ones for all the systems as compared to the multiple executions of DTW per systems, and hence, this approach is computationally less complex. DTW local distance matrix combination gave *4.78* % relative MAP improvement over the score-level fusion approach [150]. In the next Section, we present various research issues in QbE-STD task.

2.7 Research Issues in QbE-STD System

QbE-STD is expected to bypass all the errors generated by ASR, since it offers ASR-free spoken query detection framework [25]. However, QbE-STD suffers from the following research issues:

- **Acoustic Representation:** In the realistic scenarios, the same word but different spoken realizations may have different statistics due to stochastic nature of speech production mechanism (i.e., speaker variations) and difference in speech acquisition (i.e., microphones and transmission channel characteristics). This naturally results in different acoustic characteristics for these different realizations. To represent these different acoustic characteristics as a similar pattern is a technological challenge. Various speaker normalization and noise-robust approaches were attempted in the history. The majority of approaches convert speech into either frame or symbol-based representation [24].
- **Design of Speech Activity Detection (SAD) System:** The spoken query often have the silence regions in the vicinity of spoken part, since it is recorded in isolation [151]. The presence of the silence region in the query might incorrectly detect the query within the audio documents. Thus, SAD system plays a vital role in the QbE-STD system.
- **Searching Execution:** The common approaches converts spoken query and test utterances into representative templates and execute the task of template matching. However, the size of audio documents (which contains test utterances) is huge as compared to the actual presence of query. Hence, a large amount of time is required to search the query within the test utterance that

does not contain the query. This time is effective since most of the search algorithms are based on DTW, whose time-complexity is directly proportional to time (in terms of a number of frames). To alleviate this issue, data abstraction [35,53] and indexing techniques [2,131,133] were used for QbE-STD.

- **Fusion of the search systems:** There is a need to combine the evidences from different search systems because of low performance of the individual QbE-STD system. The detection scores from majority voted detections were fused using the score-dependent weights. These weights of calibration are determined by logistic regression with the scores and available ground truth information [62]. Another approach suggested as to combine the local distance matrix for various templates obtained using different front-ends and execute the DTW onto the average local distance matrix [150]. This approach is quite computationally cheaper as compared to the combining the evidences from different QbE-STD systems as suggested in [62]. However, this approach does not consider the relative significance of each audio representation that is exploited in QbE-STD and might give wrong detection if majority of audio representation favors wrong detection [150].

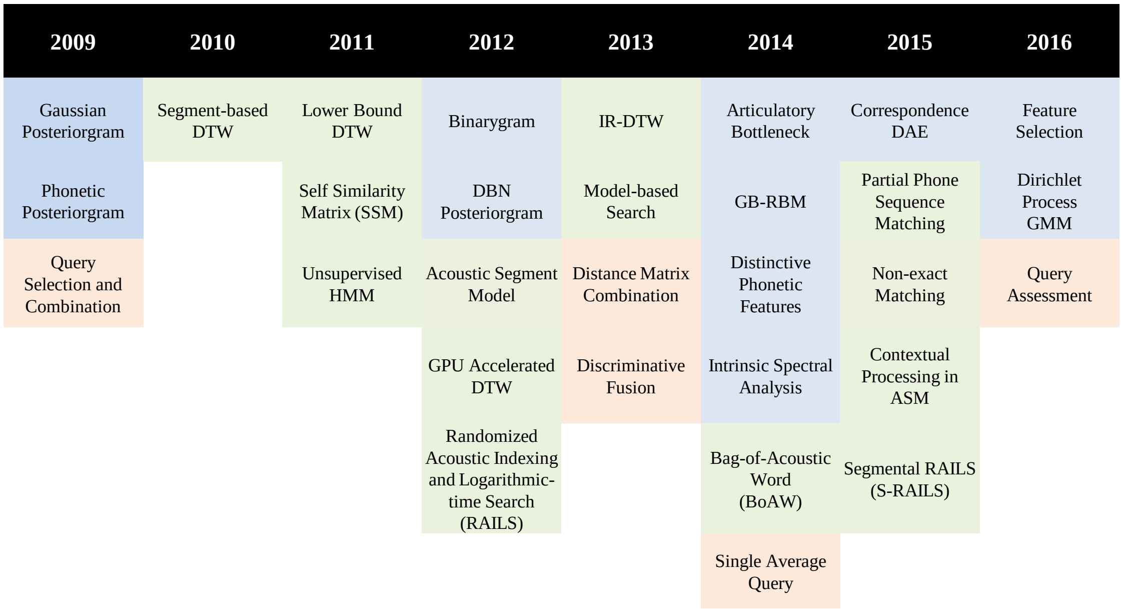
The overall summary of selected chronological progress in QbE-STD problem since 2009, is shown in Figure 2.11, which indicates that research activity is continuously evolved w.r.t. all the three different perspectives, i.e., representation, matching, and detection.

2.8 QbE-STD Submission in MediaEval

In this Section, we will discuss various submitted QbE-STD systems in MediaEval campaign in he years from 2011 to 2015. The details of QbE-STD systems submitted for MediaEval's SWS evaluations are shown in Table 2.1-Table 2.5.

2.8.1 Summary of MediaEval SWS 2011

To the best of author's knowledge, SWS 2011 evaluation was first time initiated by IBM, India [27] to develop STD systems on Indian languages (namely, Indian English, Hindi, Gujarati and Telugu). Audio documents/test utterances are having duration of 4-30 sec. Dev set contains 400 queries (100 per language) and Eval set contains 200 queries (50 per language) [27]. Most of the audio is taken in spontaneous recording mode in the real-life settings [151]. The performance metric used is MTWV. Table 2.1 shows the QbE-STD submission to MediaEval SWS 2011



2009	2010	2011	2012	2013	2014	2015	2016
Gaussian Posteriorgram	Segment-based DTW	Lower Bound DTW	Binarygram	IR-DTW	Articulatory Bottleneck	Correspondence DAE	Feature Selection
Phonetic Posteriorgram		Self Similarity Matrix (SSM)	DBN Posteriorgram	Model-based Search	GB-RBM	Partial Phone Sequence Matching	Dirichlet Process GMM
Query Selection and Combination		Unsupervised HMM	Acoustic Segment Model	Distance Matrix Combination	Distinctive Phonetic Features	Non-exact Matching	Query Assessment
			GPU Accelerated DTW	Discriminative Fusion	Intrinsic Spectral Analysis	Contextual Processing in ASM	
			Randomized Acoustic Indexing and Logarithmic-time Search (RAILS)		Bag-of-Acoustic Word (BoAW)	Segmental RAILS (S-RAILS)	
					Single Average Query		

Figure 2.11: Selected chronological progress during last few years in QbE-STD research problem.

Table 2.1: QbE-STD submission in SWS MediaEval 2011

Study	Front-end	Search Algorithm	MTWV
Telefonica [68]	Binarization of posterior features	SubDTW	0.222
Telefonica [68]	EM-ML GP	SubDTW	0.173
BUT-HCTLabs [152]	GMM/HMM query model	LLR	0.131
MUST, HLT, Microsoft [112]	Hindi AM, PhnRec	DP	0.114
Irisa [153]	MFCC, GP, PP	SLNDTW with SSM	0.100
MUST, HLT, Microsoft [112]	Hindi AM, lattice	DP	0.086
BUT-HCTLabs [152]	PAKWS	LLR	0.033
BUT-HCTLabs [152]	PP	DTW	0.014
IIIT-H [69]	Articulatory-triphone	Sliding DTW	0.000

EM-ML: Expectation-Maximization Maximum-Likelihood, AM = Acoustic Model, PAKWS = Parallel Acoustic Keyword Spotting, LLR= Log likelihood Ratio, DP = Dynamic Programming, SLNDTW = Segmentally Local Normalized DTW, GP = Gaussian Posteriorgram, PP = Phonetic Posteriorgram, PhnRec= Phoneme Recognition. All MTWV values are rounded off to 3 decimal places.

campaign. A brief summary of submitted systems at MediaEval SWS 2011 is as follows.

- Articulatory decoders were used to convert speech into articulatory symbols, and each articulatory tags have corresponding phonemic symbols in Telugu database. Miss and false alarm probabilities on evaluation data was *96* %-*98* % and *0.1* % - *0.2* % [69].
- Performance of MFCC, GP and BUT phoneme recognizer was presented at frame-level audio matching [153]. The system presented novel SSM technique

along with DTW scores.

- Hindi as an Indian language should be acoustically close to other Indian languages, which are part of SWS 2011. With this motivation, Hindi phoneme recognizer was used in [112]. Lattice and DP-based search were used.
- GMM/HMM keyword search model was found to be effective for SWS 2011 QbE-STD task. However, language-specific AKWS model did not give better performance [152].
- Unsupervised hard clustering technique was used after GMM training in [68]. In addition, a binary representation of posteriorgram with Boolean distance metric was explored.

2.8.2 Summary of MediaEval SWS 2012

SWS 2012 targets four African languages, namely, isiNdebele, Siswati, Tshivenda, and Xitsonga [151]. Dev set contains 1580 audio documents (395 per language) and 100 query examples. Eval set contains 1660 audio documents and 100 queries [151]. The performance metric used is again MTWV. Table 2.2 shows the QbE-STD submission to MediaEval SWS 2012 campaign. A brief summary of submitted systems at MediaEval SWS 2012 is as follows [24]:

- The use of phoneme decoders and their mapping using IPA symbols were presented in [154, 155].
- The use of AKWS (the likelihood ratio of a query HMM and background model) gave the promising results than the BNF-based DTW search system [139].
- To get the benefits from complementary information from different posterior representations (from different tokenizer), a parallel tokenizer followed by DTW detection (PTDTW) framework was explored in [118].
- RAILS algorithm was used to obtain scalable zero-resource search system by representing the features as bit signatures and computing the matching using image processing approaches [143].
- The majority voting scheme was introduced to keep the detection candidates supported by at least two search systems [156].
- The novel CDTW was proposed, where the softmax operation is used instead of hard maximum (or minimum) for accumulated distance computation [122].
- The SVM-driven unsupervised classification framework was proposed in [63], which used the alignment cost (obtained using DTW) to label the segment pairs.

Table 2.2: QbE-STD submission in SWS MediaEval 2012. The results of evaluation in MTWV is reported

Study	Front-end	Search Algorithm	MTWV
CUHK [118]	MFCC-GMM, MFCC-ASM; PT: CZ-GMM, HU-GMM, RU-GMM, MA-GMM, EN-GMM	Segmental DTW, PRF and score normalization	0.740
CUHK [118]	PT: CZ-GMM, HU-GMM, RU-GMM, MA-GMM, EN-GMM	SDTW, and score normalization	0.720
CUHK [118]	MFCC-GMM, MFCC-ASM	SDTW, PRF and score normalization	0.640
BUT [139]	Mel filterbank, ANN/HMM decoder	LLR	0.530
L2F [156]	PLP-RASTA, modulation spectrogram, MLP	AKWS, MV fusion	0.523
BUT [139]	Mel filterbank, BNF	DTW	0.488
JHU-HLTCOE [143]	FDLP	RAILS with SDTW	0.369
Telefonica [157]	MFCC, GP	IR-DTW	0.342
SpeeD, LAPI [154]	Romanian PhnRec	DTWSS	0.310
TUM [122]	MFCC	CDTW	0.296
Telefonica [157]	MFCC, GP	SDTW	0.311
GTTS [155]	BUT PhnRec	Approximating string matching	0.081
TUKE [63]	MFCC	Supervised SVM with MCA	0.000

PT: Phonetic Tokenizer, MCA : Minimum Cost of Alignment, MV: Majority Voting, AM = Acoustic Model, PAKWS = Parallel Acoustic Keyword Spotting, LLR= Log likelihood Ratio, GP = Gaussian Posteriorgram, PP = Phonetic Posteriorgram, PhnRec= Phoneme Recognition. All MTWV values are rounded off to 3 decimal places.

2.8.3 Summary of MediaEval SWS 2013

SWS 2013 targets nine languages [28]. These nine languages include four African languages (namely, Isixhosa, isiZulu, Sepedi, and Setswana), Albanian, Romanian, Czech, Basque and non-native English. The number of utterances are not uniform for each language and the duration of database is almost 5 times than the MediaEval SWS 2012 [28]. Dev set and Eval set contain 505 and 503 unique spoken query examples, respectively. The performance metric used is again MTWV. Table 2.3 shows the QbE-STD submission to MediaEval SWS 2013 campaign. Brief novelties of submitted systems at MediaEval SWS 2013 are as follows:

- To expand the spoken query, Pitch Synchronous Overlap and Add (PSOLA) technique was used in [158], where the query is expanded by two time-scale factors, namely, 0.7 and 1.3.
- To exploit the multiple query examples and perform the single DTW to save the execution time, the average query computation was used in [76]. The longest duration query example was considered, and all the queries were aligned to the frames of longest query example.

Table 2.3: QbE-STD submission in SWS MediaEval 2013. The results of evaluation in MTWV is reported

Team	Front-End	Search Algorithm	MTWV
GTTS [76]	BUT PhnRec	subDTW, Score normalization and calibration/fusion	0.399
L2F [62]	PLP-RASTA, Modulation Spectrogram; MLP, ANN/HMM network	AKWS, DTW and calibration/fusion	0.342
GTTS [76]	BUT PhnRec	subDTW	0.346
CUHK [158]	MFCC, GCC and CD-DNN	DTW , Distance matrix fusion, PSOLA for query expansion and Score normalization (z-norm)	0.306
BUT [113]	13 atomic systems	AKWS, score normalization (m-norm)	0.304
BUT [113]	13 atomic systems	AKWS, DTW, score normalization (m-norm)	0.297
CMTECH [102]	SAT, TAM	subDTW, score normalization (CDF-equalization)	0.269
CMTECH [102]	SAT, TAM	subDTW, score normalization (z-norm)	0.264
IIIT-H [159]	GP of FDLP+PH-BNF	NSDTW	0.249
IIIT-H [159]	GP of FDLP+AR-BNF	NSDTW	0.241
ELIRF [73]	MFCC	subDTW	0.159
Telefonica [157]	MFCC, GP	IR-DTW	0.093
Georgia tech [144]	MFCC, ergodic HMM model	Viterbi search	0.084
SpeeD [160]	Romanian PhnRec	DTWSS	0.059
UNIZA [161]	MFCC, quasi-phoneme HMM	Viterbi search	0.001
TUKE [70]	MFCC, audio segment unit	GMM-FST, SDTW	0.000

SAM : Spectral Acoustic Model, TAM : Temporal Acoustic Model, AR-BNF: Articulatory BNF, GMM-FST : GMM-Finite State Transducers, CD-DNN : Context-dependent DNN. All MTWV values are rounded off to 3 decimal places.

- The KL divergence, which considered the inter-cluster and intra-cluster variabilities in the distance was used in [102].
- The set of *i*-vectors for the query and the test utterance was used in [162]. The alignment between two *i*-vectors of the query and the test utterance were computed using DTW and used to detect the query.
- The unsupervised HMM, in particular, Semi-continuous density HMM [161] and EHMM [144] were used for QbE-STD. The Viterbi algorithm was used to detect the query.
- The novel *m*-norm was proposed to normalize the scores of the different query [113]. In *m*-norm, the scores are subtracted from the mod (maximum) value and divided by the standard deviation. To fuse several detection evidences, 13 search systems (i.e., atomic systems) were used [113].
- The articulatory information were used to train the MLP, and their BNF were used for QbE-STD because of its language-independent characteristics [159]. The Gaussian posteriorgram of BNF with raw FDLP features gave the improve-

ment than the MLP phonetic posteriorgram features.

2.8.4 Summary of MediaEval QUESST 2014

MediaEval QUESST 2014 database targeted six languages, namely, Albanian, Basque, Czech, non-native English, Romanian, and Slovak. In addition to the exact lexical match, this evaluation contains query with lexical variation at start or end as well as word reordering in multiple words [29]. The audio documents consists duration of 23 hours (12492 files). The Dev and Eval sets contain 555 and 560 queries, respectively [29]. The primary evaluation metric used in QUESST 2014 is C_{nxe}. Table 2.4 shows the QbE-STD submission to MediaEval QUESST 2014 campaign. Brief novelties of submitted systems at MediaEval QUESST 2014 are as follows:

- MSC is used to perform labeling to segments of posterior feature vectors. After initial label estimation, HMM training and decoding are performed iteratively until label sequence converges [163].
- The raw and choi processed MFCCs were used in a zero-resource approach in [164]. The results showed that choi processed MFCC did not give any improvement than the MFCC. Similarly, PNCCs were used in [64] as noise-robust acoustic representation.
- Various supervised [64, 140, 149, 165] and unsupervised [163, 165, 166] feature representation were used to improve the detection performance.
- The split query with AKWS was used to improve the performance of non-exact matched type of query, which is having lexical variation either at start, end or middle [149].
- Partial matching schemes were used to perform non-exact query matching task to deal with truncation, filler and re-ordering cases [167].

2.8.5 Summary of MediaEval QUESST 2015

MediaEval QUESST 2015 focused on seven languages, namely, Albanian, Czech, English, Mandarin, Portuguese, Romanian, and Slovak. QUESST 2015 evaluation comprises about *450* spoken queries in Dev and Eval sets. These queries have exact as well as non-exact match having filler and reordering. The noise and room reverberation effect were artificially introduced in recorded queries by simulation [30]. Most of the QbE-STD systems developed by the participants uses more

Table 2.4: QbE-STD submission in QUESST MediaEval 2014. The results of evaluation in C_{nxe}^{min} is reported

Team	Front-End	Search Algorithm	Results (Eval)
BUT [149]	10 PhnRec	AKWS, DTW; score normalization (m-norm); calibration/fusion	0.464 (0.323 / 0.470 / 0.660)
BUT [149]	10 PhnRec	AKWS, DTW; score normalization (m-norm); calibration/fusion with side information.	0.465 (0.310 / 0.461 / 0.673)
SPL-IT [167]	BUT PhnRec (CZ, HU, RU)	Partial matching DTW	0.5080
GTTS [75]	BUT PhnRec	subDTW, calibration/fusion	0.5994 (0.440 / 0.641 / 0.770)
NNI [166]	PhnRec	DTW, WFST-based AKWS	0.6023 (0.507 / 0.621 / 0.725)
CUHK [163]	GP, ASM; PhnRec: CZ, HU, RU, MA, EN	DTW, distance matrix fusion	0.659 (0.486 / - / -)
NNI [166]	9 PhnRec	DTW	0.6925 (0.573 / 0.730 / 0.803)
TUKE [165]	Slovak PhnRec	Weighted fast sequential variant of DTW	0.891
IIIT-H [140]	FDLP, GP of FDLP+AR-BNF	NS-DTW	0.922 (0.812 / 1.021 / 1.001)
TUKE [165]	MFCC, Viterbi decoding	Weighted fast sequential variant of DTW	0.934
IIIT-H [140]	Telugu PhnRec	NS-DTW	0.949 (0.933 / 0.960 / 0.964)
ELIRF [164]	MFCC, GP	subDTW	0.965
ELIRF [164]	MFCC (choi processing), GP	subDTW	0.967
SPEED [64]	PNCC; Romanian, Albanian, English PhnRec	DTWSS	0.972 (0.972 / 0.970 / 0.963)

PhnRec : Phoneme recognizer, MA : Mandarin, AR-BNF : Articulatory BNF, DNN : Deep Neural Network, LSTM : Long Short-Term Memory, RNN : Recurrent Neural Network, TRAP : TempoRAl Pattern, BNF : Bottleneck Features, SBNF : Stacked BNF. All C_{nxe}^{min} values are rounded off to 3 decimal places. The dash (-) symbol indicates no values reported in the respective papers.

than the single phoneme recognizer (PhnRec) front-end and subsequence DTW. Finally, discriminative calibration was performed to execute score-level fusion. Table 2.5 shows the QbE-STD submission to MediaEval QUESST 2015 campaign. Brief novel concepts that were the outcome of QUESST 2015 evaluation are as follows:

- Noise filtering and pre-processing were introduced before feature extraction stage. The spectral subtraction was used to remove noise from audio by estimating the power spectral density (PSD) of noise [168, 172]. The Wiener filter approach was used in [177] to reduce the noise. In addition, data-augmentation approach was used, where noise is artificially added in training to establish the matching condition with QUESST 2015 [177].
- Several deep learning frameworks were used for zero resource and low resource scenarios. The zero-approach Recurrent Neural Networks (RNN)-based

Table 2.5: QbE-STD submission in MediaEval QUESST 2015. The results of evaluation in C_{nxe}^{min} is reported

Team	Front-End	Search Algorithm	Results (Eval)
NNI [166]	Phonetic symbol, BNF, SBNF	DTW; score-level fusion of 66 systems	0.747 (0.577 / 0.831 / 0.769)
SPL-IT-UC [168]	5 PhnRec	SubDTW with 6 different partial matching strategies; calibration/fusion, side information	0.781
BUT [169]	7 PhnRec: SpeechDat, GlobalPhone: 1 stacked BNF	DTW with slope constraint; Partial match in 2 and 3 bands	0.812 (0.742 / 0.845 / 0.821)
BUT [169]	7 PhnRec: SpeechDat, GlobalPhone: 1 stacked BNF	DTW with slope constraint	0.818 (0.741 / 0.852 / 0.831)
BUT [169]	7 PhnRec: SpeechDat, GlobalPhone	DTW; majority voting calibration/fusion with side information	0.826 (0.757 / 0.865 / 0.834)
ELiRF [170]	BUT PhnRec	SubDTW; calibration/fusion	0.875
GTM-Uvigo [171]	11 PhnRec: DNN, LSTM, TRAP	Phoneme unit selection, DTW; calibration/fusion	0.905 (0.861 / 0.928 / 0.904)
IIT-B [172]	3 PhnRec, GP	DTW; majority voting calibration/fusion	0.908 (0.868 / 0.911 / 0.921)
TUKE [173]	4 PhnRec: GMM training	DTW; max score merging	0.971
CUNY [174]	MFCC	LB Keogh for search score reduction followed by subDTW	0.985
SPEED [175]	Multilingual common phones	DTWSS	0.993
NTU [176]	MFCC: zero-resource RNN	DTW	0.997

PhnRec : Phoneme recognizer, MA : Mandarin, AR-BNF : Articulatory BNF, DNN : Deep Neural Network, LSTM : Long Short-Term Memory, RNN : Recurrent Neural Network, TRAP : TempoRAl Pattern, BNF : Bottleneck Features, SBNF : Stacked BNF. All C_{nxe}^{min} values are rounded off to 3 decimal places.

approach was presented in [176]. The phoneme posteriorgram trained using a long-short-term memory (LSTM) neural networks and DNN were presented in [171].

- To address non-exact matching, several partial matching strategies were employed at frame-level or at symbol-level. In particular, the phone sequence approximate matching was used in [177] and the partition in DTW distance matrix at different locations were used in [168, 169].
- The most common relevant phoneme unit selection approach to improve search performance was presented in [171].
- One common observation was the problem of non-exact matching is too hard, and many times, while dealing with non-exact matching, the performance of an exact match $T1$ type of query got affected.

2.9 Chapter Summary

In this chapter, we discussed the literature survey of various methods for QbE-STD task, where the spoken content retrieval task is attempted using spoken form of a query. The STD uses the text-based query and retrieval is conducted using the ASR and information retrieval system. Due to a scarcity of adequate transcribed data in low-resourced languages, ASR is not feasible in many such languages. To retrieve spoken content in the audio of such languages, a spoken example is exploited and is regarded as a query in the QbE-STD framework. This chapter summarized various subsystems of QbE-STD system, namely, front-end subsystem, searching subsystem, and detection subsystems. Finally, we presented various submission of QbE-STD systems in MediaEval from years 2011 to 2015. In the next chapter, we will present the brief details of experimental setup that is used in this thesis to perform QbE-STD task.

CHAPTER 3
Experimental Setup

3.1 Introduction

In this chapter, the details of an experimental setup used in QbE-STD system is discussed. The organization of this chapter is as follows. Section 3.2 gives the details of databases used in the QbE-STD. In this thesis, we have used three databases, namely, SWS 2013 and QUESST 2014. As discussed earlier in Section 2.2, QbE-STD system constitutes many components (i.e., sub-systems), namely, front-end subsystem, searching sub-system and detection sub-system. In that framework, the thesis considers on the frame-based posteriorgram representation and subDTW as a searching algorithm. Section 3.3 presents the details of acoustic representation and posteriorgram representation. Section 3.4 presents the details of the subDTW algorithm, which is extensively used in this thesis. Section 3.5 presents the details of score normalization in detection subsystem along with, the effect of local constraints and dissimilarity functions (or distance metrics).

3.2 Databases Used

In this thesis, three databases are used for QbE-STD task, namely, (A) SWS 2013 database, and (B) QUESST 2014 database. Next, the details of these databases, such as, the number of keywords (queries), their instances, duration (average), the number of utterances in test dataset, etc. are described in brief.

(A) SWS 2013 database

MediaEval SWS 2013 dataset is used for unranked evaluation, where the detection of query is made based on the threshold value. All audio recordings are having *8* kHz sampling rate with *16* bits/sample PCM encoded **.wav* format. The statistics of the database is given in Table 3.1. Two sets of a query are categorized as Devel-

Table 3.1: Statistics of MediaEval SWS 2013 database. After [178]

Data	# Utterances	Average duration (in sec.)
Test	10762	6.67
Dev query	505	1.35
Eval query	503	1.38

Table 3.2: Statistics of MediaEval QUESST 2014 database. After [31]

Data	# Utterances	Average duration (in sec.)
Test	12492	6.65
Dev query	560	2.18
Eval query	555	2.10

opment (Dev), and Evaluation (Eval) sets. Common test audio data is provided, which is used for both the query types. The objective of development set is to fine tune the parameters in the design of QbE-STD system. The performance of evaluation set is investigated under this tuned parameters. SWS 2013 data consists of two types of queries, namely, normal or basic type (which involve only one example per query), and extended type (which includes multiple examples per query). We use the cepstral-domain features from test data for GMM training.

(B) QUESST 2014 database

MediaEval QUESST 2014 consists of *23* hours total search data. All audio recordings are having *8* kHz sampling rate with *16* bits/sample PCM encoded **.wav* format [31]. The statistics of QUESST 2014 database is given in Table 3.2. Findings from SWS 2013 database suggests that though Czech (CZ) have a similar acoustic condition in audio documents and query, it was performing worse. One of the possible reasons could be rapid variations in speaking rate of conversational speech that might give short query, when excised through speech cuts using forced alignment [31]. This motivates to record the query from speech directly from the user for QUESST 2014.

The major distinctive characteristics of this database lies into the form of query. The structure of the query can motivate to match audio using approximate search rather than only the exact search. The matching between audio can be categorized into three different types, namely, Type 1, Type 2 and Type 3.

- **Type 1 :** This type of audio matching refers to the exact match. For example, if a query is *Funny joke*, it should match the audio document containing '*This is a* *funny joke*.'

- **Type 2 :** This type of audio matching refers to the lexical variation of query either at the begin or at the end. The ground truth of such query is prepared such that matching part should be more than 250 ms than a non-matching part. In addition, the matching part should be more than a non-matching part. For example, if a query is '*perform*', it should match the audio document containing, '*That was a nice performance*' and vice-versa. In addition, if a query is, '*encounter*', it should match the audio containing, '*Please! come at the first counter*' and vice versa.

- **Type 3 :** This type of audio matching refers to the reordering and filler cases of a query. The audio document contains all the words of multi-word query; however, the sequence of words may be different than the query. In addition, some filler words can be present the audio contents. For example, if a query is, '*Funny joke*,' it should match the audio document containing, '*This joke is funny*.'

3.2.1 Challenges in Databases

QUESST 2014 requires to perform non-exact matching of spoken query, which is not straightforward. Thus, subDTW does not able to perform non-exact match, and thus, the results of a non-exact match are poor as compared to the exact match with the subDTW search algorithm. We will discuss the subDTW search algorithm in Section 3.4.

3.3 Front-end Subsystem

3.3.1 Acoustic Representation

In this thesis, Mel Frequency Cepstral Coefficients (MFCC) [51], Perceptual Linear Prediction (PLP) cepstral coefficients [55] and MFCC-TMP [179] features are used. The objective of using different types of cepstral-domain features is to analyze the proposed representation, namely, VTL-warped Gaussian posteriorgram and posteriorgram using a mixture of GMMs are not biased with any particular type of representation. PLP features are better for formant matching across different age groups as reported in [55] and hence, expected as a good cepstral representation of speech.

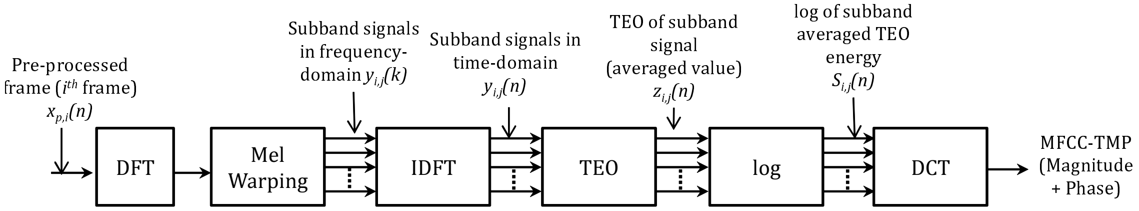

Pre-processed
frame (i^{th} frame)
$x_{p,i}(n)$
DFT
Mel Warping
Subband signals in frequency-domain $y_{i,j}(k)$
IDFT
Subband signals in time-domain
$y_{i,j}(n)$
TEO
TEO of subband signal (averaged value)
$z_{i,j}(n)$
log
log of subband averaged TEO energy
$S_{i,j}(n)$
DCT
MFCC-TMP (Magnitude + Phase)

Figure 3.1: Schematic block diagram of MFCC-TMP feature extraction. After [179, 182].

3.3.1.1 MFCC and PLP

We have used 26 subband filters spanning 0-4000 Hz frequency range and 13 DCT coefficients (Type III of DCT) for feature extraction. Features are extracted on *25* ms window duration with *10* ms frame shift. Here *13* coefficients along with their delta (Δ) and delta-delta (Δ^2) features are considered. The details of the feature extraction computation and feature vector formation scheme are as follows. MFCC and PLP features are extracted using Hidden Markov Model Toolkit (HTK) [180].

3.3.1.2 MFCC-TMP

MFCC-TMP stands for Mel Frequency Cepstral Coefficients where subband energy is computed via Teager Energy Operator (TEO) considering Magnitude and Phase part of subband signal. In MFCC feature extraction, conventional l^2 norm is used for subband energy computation. Hence, the energy of subband signal is equal to the sum of squared values of magnitude spectrum [51], whereas, in MFCC-TMP, Teager energy (which is running estimate of signal's energy) is used instead of l^2 energy. The time-domain signal is used to compute TEO of subband signal [179]. A nonlinear energy tracking operator referred to as Teager Energy Operator (TEO) (denoted as ψ) for discrete-time signal, $x(n)$, is defined as [181]:

$$TEO\{x(n)\} = \psi\{x(n)\} = x^2(n) - x(n+1)x(n-1). \tag{3.1}$$

The feature extraction procedure for MFCC-TMP is shown in Figure 3.1. Finally, normalized subband energy is computed followed by logarithm and Discrete Cosine Transform (DCT) operations to get proposed feature set, namely, MFCC-TMP, i.e.,

$$MFCC - TMP_i(k) = \sum_{j=1}^{N_F} Sl_{i,j} cos\left(\frac{k(j-0.5)\pi}{N_F}\right), \tag{3.2}$$

where $k = 1, 2, \cdots, N_c$, $N_c =$ dimensions of feature vector (*13* in this work), $N_F =$

number of filters used in the Mel filterbank (*26* in this work) and $S_{i,j}$ = logarithm of subband energy for i^{th} frame index and j^{th} subband filter.

3.3.2 Posterior Representation

These acoustic representations are transformed into the posterior representation for better query detection. Conventionally, Gaussian posteriorgram are computed with the help of acoustic representations (such as, MFCC, PLP, MFCC-TMP).

3.3.2.1 Motivation for GMM

There are two major motivations behind using GMM for cepstral representation. The first reason is the belief that each component of multi-modal distributions, such as GMM represents the distinct acoustical events in the speech [183]. Each such acoustic events are caused due to average vocal tract configuration, which is characterized by the mean component μ_i. The variation in vocal tract structure is characterized by the covariance Σ_i. The unbalanced number of different acoustic classes can be characterized in terms of the weights in GMM. The second reason is the approximation capability of GMM that a GMM can be useful to approximate any arbitrary distribution without any labels [183]. Thus, we can also represent the cepstral-domain features with the help of a weighted linear combination of mean vectors.

The details of Gaussian posteriorgram is presented in sub-Section 2.4.3.2. We will discuss VTLN-warped Gaussian posteriorgram and mixture of GMMs posteriorgram in the Chapter 4.

As discussed earlier in sub-Section 2.4.3.2, the Gaussian posterior probability $P(C_k|\mathbf{o}_t)$ (for k^{th} cluster and t^{th} speech frame index) is:

$$P(C_k|\mathbf{o}_t) = \frac{\pi_k \mathcal{N}(\mathbf{o}_t; \mu_k, \Sigma_k)}{\sum_{j=1}^{Np} \pi_j \mathcal{N}(\mathbf{o}_t; \mu_j, \Sigma_j)}, \tag{3.3}$$

where the likelihood of feature $\mathbf{o}_t$ being in k^{th} cluster is,

$$\mathcal{N}(\mathbf{o}_t; \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_k|}} \exp\left(-\frac{1}{2}(o_t - \mu_k)^T \Sigma_k^{-1} (o_t - \mu_k)\right). \tag{3.4}$$

The GMM parameters are estimated using Expectation-Maximization (EM) algorithm. The initial parameters are set by the vector quantization (VQ) codebook computed via Linde-Buzo-Gray (LBG) algorithm [184]. The procedure of VQ codebook preparation is demonstrated in Figure D.1.

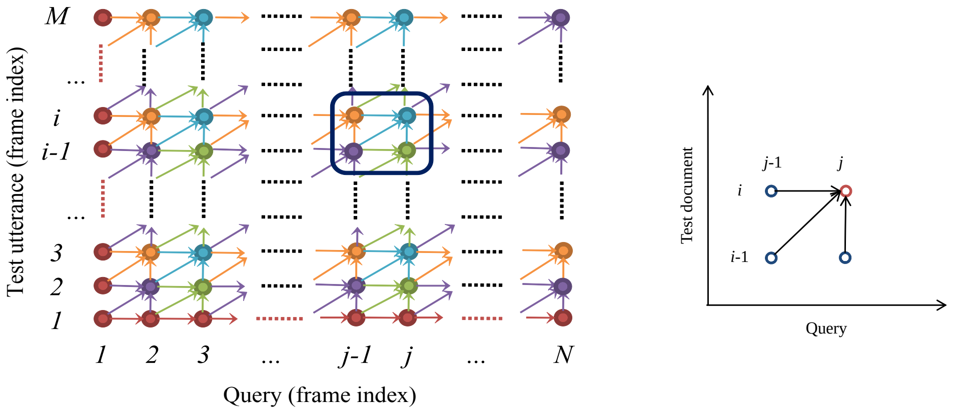
In addition to Gaussian posteriorgram, we have also used phonetic posteriorgram. The phone posterior is obtained using open source Brno University's phoneme recognizer [61]. Czech (CZ), Hungarian (HU) and Russian (RU) phonetic recognizer systems were trained on the SpeechDat-E databases. We merge the state posterior probability into single as performed in the study presented in [74]. Furthermore, we perform Speech Activity Detection (SAD) using all the phone posteriorgram (i.e., CZ, HU, and RU). Speech Activity Detection (SAD) separates the speech and non-speech part from the spoken audio. Non-speech regions may be background noise, babble or silence regions present in the speech signal. In QbE-STD task, the non-speech region of the speech is not important in the detection and consumes unnecessary search processing time. In addition, silence region or babble may resemble the similarity in posteriorgram features, and hence, it may create an ambiguity in the query detection and reduces the search performance. We considered the average of the posterior probability of non-speech units (such as, *pau*, *int*, and *spk*) from CZ, HU and RU to perform speech activity detection (SAD). Thus, we have 43, 59 and 50 speech units, corresponding to CZ, HU and RU phoneme posteriorgrams, respectively [74].

In the thesis, we proposed VTL-warped Gaussian posteriorgram and mixture of GMMs posteriorgram. VTL-warped Gaussian posteriorgram removes the speaker variability caused due to spectral scaling variations (Please refer Section 4.2 in Chapter 4). The mixture of GMMs brings broad phoneme posterior probability during while training of GMM. Hence, this might be useful to emphasize broad phoneme class-related information into the posteriorgram (please refer Section 4.3 in Chapter 4).

3.4 Searching Subsystem

The searching subsystem consists of subsequence DTW (subDTW) as searching algorithm [120]. Let the dimension of posteriorgram be N_p and a posteriorgram feature vector sequence for spoken query, $q_y = (q_y^1, q_y^2, \cdots, q_y^N)$ and a test utterance, $t_x = (t_x^1, t_x^2, \cdots, t_x^M)$. The local distance between two posterior vectors, namely, t_x^i and q_y^j, is computed using the symmetric Kullback-Leibler (KL) divergence and is defined as [82, 98]:

$$Dis_{t_x^i, q_y^j} = \sum_{k=1}^{N_d} t_x^i(k) \log \left(\frac{t_x^i(k)}{q_y^j(k)} \right) + \sum_{k=1}^{N_d} q_y^j(k) \log \left(\frac{q_y^j(k)}{t_x^i(k)} \right). \tag{3.5}$$



(a) (b)

Figure 3.2: (a) Generation of warping path using subDTW and (b) local constraints used in subDTW. The red circles show initial points and the rest of the circles are computed recursively. The values of accumulated distance matrix at an arbitrary node (i, j) can be determined by the adjacent nodes, which is shown in Figure 3.2 (a). After [2].

As posteriorgram represents the probability density function (*pdf*) across different phonetic class, KL-divergence between two posterior vectors corresponds to the divergence between two *pdf*s [82, 98]. The KL divergence between two *pdf*s shows the relative entropy, which does not hold symmetric property. However, we have used symmetrical version of KL divergence to find the distance between two posteriorgram vectors. The KL divergence-based local distance was found to be effective on posterior feature vectors [52]. This might be because of the nature of posterior feature vectors, which can be regarded as *pdf* or probability mass function (*pmf*). We consider the local constraints as shown in Figure 3.2.

Each cell represents the pair of test utterance frame and query frame. The cells on rows and columns indicate frames associated with test utterance and query, respectively. The matrix S stores the accumulated distance for the optimal warping path and the frame-counting matrix T stores the length of the optimal warping path. The starting frame indicator matrix P is used to store the starting frame index for the corresponding warping path, which removes the need of backtracking. The procedure is used to execute subsequence DTW with local constraints is specified as in Figure 3.2. For a single query, q_y, and test utterance pair, t_x, the local distance matrix $D = Dis_{t_x,q_y}$.

Initialization:

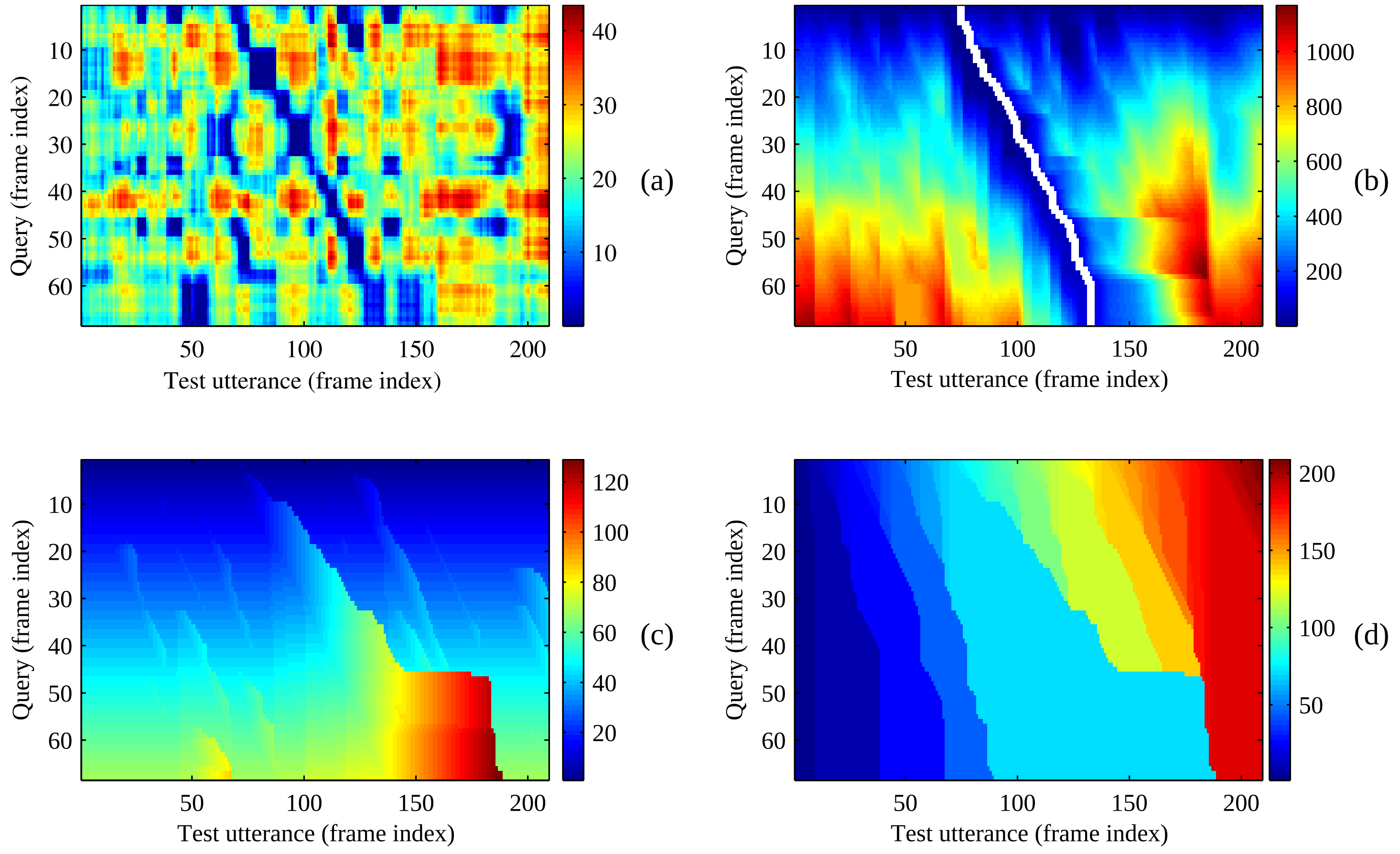


Figure 3.3: The graphical representation of matrices associated in subDTW operation using local symmetrical local constraints as shown in Figure 3.2 (b): (a) local distance matrix D, (b) accumulated distance matrix S (white line shows the warping path obtained by the algorithm), (c) frame counting matrix T, and (d) starting frame indicator matrix P. Note that image plots are rotated as row indicates query frame index and column indicates test utterance frame index. After [2].

For $j = 1, i = 1, 2, \cdots, M$:

$$\begin{aligned} S(i,j) &= D(i,j), \\ T(i,j) &= 1, \\ P(i,j) &= i. \end{aligned} \tag{3.6}$$

Path-tracing: For $i = 1$:

$$\begin{aligned} S(i,j) &= \sum_{k=1}^{j} D(i,k), \\ T(i,j) &= j, \\ P(i,j) &= i = 1. \end{aligned} \tag{3.7}$$

For $j > 1, i = 2, 3, \cdots, M$:

$$\Omega = \{(i, j-1), (i-1, j-1), (i-1, j)\}, \tag{3.8}$$

$$(r,s) = \underset{(a,b)\in\Omega}{\operatorname{argmin}}\, S(a,b), \tag{3.9}$$

$$S(i,j) = S(r,s) + D(i,j), \tag{3.10}$$

$$T(i,j) = T(r,s) + 1, \tag{3.11}$$

$$P(i,j) = P(r,s). \tag{3.12}$$

An algorithm 1 presents the pseudo-code for subDTW with symmetrical local constraint. Figure 3.3 (a) shows a local distance obtained using eq. (3.5) between each frame of query and test utterance. Figure 3.3 (b) shows the accumulated distance (computed as per eq. (3.10)) indicating the diagonal trace indicating the presence of the query. The warping path along distance accumulation can be traced with backtracking (which is shown as a white colored path). To count the number of cell on the white colored warping path (as shown in Figure 3.3 (b)), we used frame counting matrix T (computed as per eq. (3.11)). The image plot for frame counting matrix T is shown in Figure 3.3 (c). It can be that as query frame index increases, the value of T matrix increases indicating that more number of cells traced as query frame index increases. The backtracking requires additional computation, and the exact alignment path is not required rather the start and end time stamps are important. With this consideration, we used start frame indicator or path tracing matrix P. Figure 3.3 (d) shows the path tracing matrix P that contains few distinct colors indicating different warping paths for different starting

Algorithm 1 An algorithm for matrices S, T and P computation from the matrix D for *symmetrical* local constraint, LC_1 (as specified in Figure 3.2 (b).). After [120].

Input: Matrix D

Output: Matrices S, T and P

```
    Initialization :      # 1^{st} column, i.e., $j = 1$
for $i = 1$ to $M$ do
    $S(i, 1) = D(i, 1)$
    $T(i, 1) = 1$
    $P(i, 1) = i$
end for
    Path tracing :        # 1^{st} row, i.e., $i = 1$
for $j = 2$ to $N$ do
    $S(1, j) = S(1, j - 1) + D(1, j)$
    $T(1, j) = j$
    $P(1, j) = 1$
end for
    Path tracing :        # For the rest: $i > 1$ and $j > 1$
for $i = 2$ to $M$ do
    for $j = 2$ to $N$ do
        $\Omega = \{(i, j - 1), (i - 1, j - 1), (i - 1, j)\}$
        $(r, s) = \underset{(a,b)\in\Omega}{\text{argmin}}\, S(a, b)$
                          # Selecting the predecessor
        $S(i, j) = S(r, s) + D(i, j)$
        $T(i, j) = T(r, s) + 1$
        $P(i, j) = P(r, s)$
    end for
end for
```

frame index. In this thesis, warping path interval less than twice the length of the query length (i.e., *2N*), and greater than half of the query length (i.e., $N/2$) are considered as valid warping paths, and hence, remaining warping paths, which do not satisfy this condition are not considered here.

For SWS 2013 and QUESST 2014 details, we employ different strategies to obtain DTW distance due to different nature of the task. There are different warping paths across the test utterance. For SWS 2013, we select the warping path having the least average DTW distance. In this way, we considered maximum seven warping path intervals and the associated distances for each query and test utterance pair. In practice, the execution of selection of seven warping paths for each query and test utterance pair is given in Algorithm 2. Each endpoint and corresponding start points corresponds to warping paths (determined by the matrix P) and warping cost (determined by the matrices S and T). We select the warping paths having length lesser than the twice and greater than the half of the query length. Later, we select the minimum cost warping paths across different groups and store them (as valid start point (spv), end point (epv) and DTW distances (dv)) as shown in Algorithm 2. The more than one presence of a query in test utterance (in SWS 2013 dataset) demands to consider more than one warping path. The selection of seven warping paths was made based on the optimal performance on the Dev set. The top N_{top} *=1000* minimum distance values are considered. The rationale behind taking maximum seven warping paths for each query and utterance pair and N_{top} =1000 distance values, is due to the high performance gain with these settings as reported in [74]. For QUESST 2014, the objective is to retrieve the test utterance rather than detecting the location (time stamp) of a query. Hence, we consider distances for all the test utterance with a query. Thus, for QUESST 2014, N_{top} = 12492, i.e. , total number of test utterances (as stated in Table 3.2).

3.5 Detection Subsystem

The distance values per query are taken and the negative of their normalized distance are treated as *scores*. For instance, consider the top N_{top} distances based on their minimum value per query are, i.e., $ds_1, ds_2, \ldots, ds_{N_{top}}$. Now, score normalization is performed to the obtained normalized distance values $\tilde{ds}_1, \tilde{ds}_2, \ldots, \tilde{ds}_{N_{top}}$, respectively, where $\tilde{ds}_i = \frac{ds_i - \mu_q}{\sigma_q}$ and μ_q and σ_q indicate the mean and the standard deviation of $ds_1, ds_2, \ldots, ds_{N_{top}}$, respectively. The respective scores associated with each detection are $s_1, s_2, \ldots, s_{N_{top}}$, where $s_i = -\tilde{ds}_i$. Algorithm 3 shows the proce-

Algorithm 2 An algorithm for warping path selection for each test utterance having M frames and query having N frames pair for SWS 2013 database.

Input: Matrices S, T and P.

Output: $Nbst$ or less warping paths (starting point spv and ending point epv) with their distances (dv). In this thesis, $Nbst = 7$.

```
Group warping paths :
for $i = 1$ to $M$ do
    $dist(i) = \frac{S(i,N)}{T(i,N)}$  # Path length normalized DTW distance
    $wr(i) = \frac{(i-P(i,N))}{N}$  # Slope constraint
    Check valid warping path :
    if $(wr(i) \leq 2)\&(wr(i) \geq \frac{1}{2})$ then
        $V_{wr}(i) = 1$
    else
        $V_{wr}(i) = 0$
    end if
    $G = \{0\}_{M \times 1}$  # Initialize group assignment as 0 (no group)
    $CG = 1$  # Current group
    if $V_{wr}(i) = 1$ then
        $G(i) = CG$
    else
        $CG = CG + 1$
    end if
end for
Select best warping paths :
for $k = 1$ to $\max(G)$ do
    Let, the set $Sk := \{i|G(i) = k\}$
    $dv(k) = \min_{i \in S_k} dist(i)$
    $epv(k) = \operatorname{argmin}_{i \in S_k} dist(i)$
    $spv(k) = P(epv(k), N)$
end for
Sort and select $Nbst$ best warping paths based on minimum distance values.
If $max(G) \leq Nbst$ then select all warping paths.
```

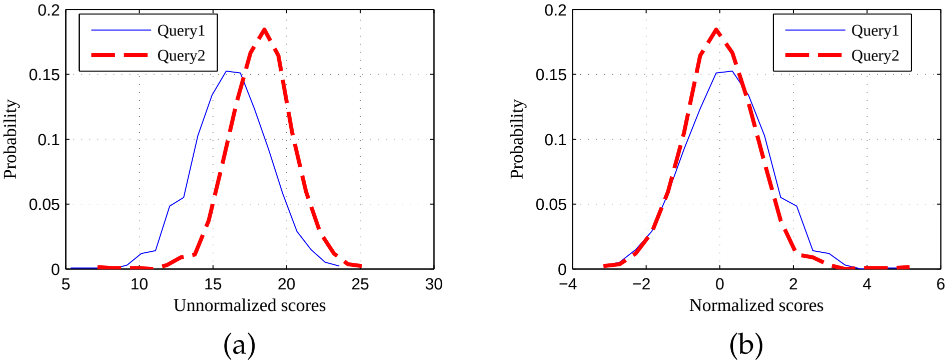


Figure 3.4: The distribution of unnormalized and normalized scores. (a) Unnormalized scores (DTW distance/cost), and (b) normalized scores. The PDF is approximated from the histogram with 20 bins.

dure to execute score normalization for a single query, that is adapted from [62].

Algorithm 3 An algorithm for score normalization for a single query. Adapted from [74].

```
Input: Unnormalized $N_{top}$ distance values: $ds_1, ds_2, \ldots, ds_{N_{top}}$.
Output: Normalized $Ntop$ score values: $s_1, s_2, \ldots, s_{N_{top}}$.
1: Sample mean: $\mu_q = \frac{1}{N_{top}} \sum_{k=1}^{N_{top}} ds_k$
2: Sample standard deviation: $\sigma_q = \sqrt{\frac{1}{N_{top}-1} \sum_{k=1}^{N_{top}} (ds_k - \mu_q)^2}$
3: Normalized scores: $s_i = -\frac{(ds_i - \mu_q)}{\sigma_q}$, for $1 \leq i \leq N_{top}$
```

Figure 3.4 (a) shows the probability density function (*pdf*) for unnormalized scores (DTW alignment cost/distance) of the same query (‘*intelligence*’) spoken by two different speakers. It can be observed from Figure 3.4 (a), that the distribution of unnormalized scores follows Gaussian distribution. The reason could be explained as follows. Note that, the distance value computed from subDTW are the accumulation distance over warping. The accumulation process is summing the distances across the warping path. With an assumption that the distribution of local distance matrix values has finite mean and variance, the unnormalized scores (DTW distances) follow a Gaussian distribution according to the law of large numbers [185]. Figure 3.4 (b) shows the probability density function for normalized scores for the query (‘*intelligence*’) spoken by two different speakers. The distribution seems identical after score normalization and also both *pdf*s in 3.4 (b) are in the vicinity of 0 indicating the distribution is centralized to zero (i.e., mean equals to zero).

As discussed in sub-Section 1.2.1, the score normalization is important because the threshold for query detection should not be biased to particular sets of queries. After normalization, we used an arbitrary threshold value 2, for Dev set. The operating threshold at which Dev set gives MTWV is used for Eval set. For QUESST 2014, we optimized the threshold to achieive MTWV on Dev set and calibrate the scores that minimizes C_{nxe}. We use Bosaris toolkit to compute the C_{nxe} and weights to obtain minimum C_{nxe} across all types of queries [186]. The details related to the threshold θ selection is given in Appendix E.

3.5.1 Score-level Fusion of Multiple Systems

Earlier different speaker and language recognition systems were fused at score-level to improve the performance of speaker and language recognition task [187, 188]. The score-level fusion approaches assume that the scores are available for each trial, which is not in the case of QbE-STD. All the detection candidates from several QbE-STD systems may not be synchronized, i.e., having different (time-stamps) start and end positions that hypothesize the location of the query in the utterance. For instance, in Figure 3.5 (a) shows three QbE-STD systems that does not have the exact time-synchronous detection candidates. However, they are aligned as part of the each detection candidates may have overlap with another. Thus, the time stamps that covers all the detection candidates are taken. In some cases, few QbE-STD systems might not produce the output, i.e., does not give the detection scores, whereas other QbE-STD systems produce the detection scores for that detection candidate. As shown in Figure 3.5 (b) that detection candidates from two QbE-STD systems are aligned, and the system-3 does not produce the score for that detection candidate. The detection score (missing score) for such detection candidate is the default score, which is minimum score per query or minimum score per system. Thus, the detection regions from various search systems are aligned such that their time-stamps overlap based on the majority voting decision. In few of the cases, if there is no detection region for a particular search system (i.e., missing scores), a default score is assigned. In this thesis, the missing score is considered as the minimum score per query.

After the alignment, scores are calibrated using logistic regression, where inferences (i.e., the ground truth) are taken from the Dev set. The scores obtained from different systems are combined using the discriminative fusion approach presented in [62]. For given NS systems having t trials are fused as [62]:

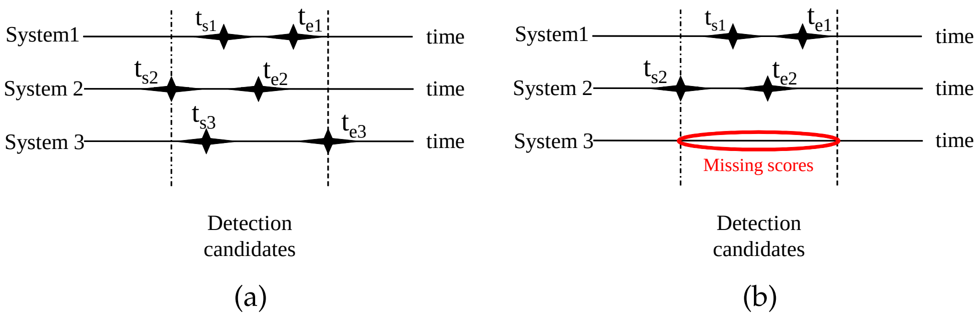


Figure 3.5: Illustration of fusion for three QbE-STD systems: (a) all the detection candidates are having overlap, and (b) missing detection candidates. The t_{si} and t_{ei} indicate the start and end times of detection candidate for i^{th} QbE-STD system. After [62].

$$\hat{s}_t = \xi_0 + \sum_{i=1}^{NS} \xi_i s_t^i, \tag{3.13}$$

where s_t^i is the score of i^{th} system for t^{th} trial, ξ_i's are fusion/calibration coefficients, which are estimated by binary logistic regression [62]. The scripts for fusion multiple QbE-STD systems is available online [189].

3.6 Effect of Local Constraints (LC)

The relative local temporal mismatch between a query and utterance due to different speaking rates by various speakers may require additional treatments in the search algorithm. In particular, the locality constraints considered during DTW distance accumulation has to be adjusted. The feature alignment is performed by similarity matching of consecutive features by considering different local constraints. We analyze the performance of the QbE-STD task for various local constraints in subDTW. To that effect, Figure 3.6 shows three different local constraints for DTW-based searching. To use these local constraints, we need to change the initialization of subDTW algorithm and modify the eq. (3.6) and eq. (3.7). The rest of the computation remains the same for all the local constraints that are used in this theses. In the analysis of DTW presented earlier in Figure 3.3, we used the local constraint, LC_1. The pseudo codes for other asymmetrical local constraints, i.e., LC_2 and LC_3, pseudo codes are described in Algorithm 4 and Algorithm 5, respectively (modified after [120]).

The relative temporal mismatch between the query and the instance of query, which is present in the utterance (due to different speaking rates by the various speakers) may require additional treatments in the search algorithm. In particular,

Algorithm 4 An algorithm for matrices S, T and P computation from the matrix D for *asymmetrical* local constraint, i.e., LC_2 (as specified in Figure 3.6 (b).) Note that, in local constraints LC_2, the values in matrix T depends only on the number of frames in query, i.e., N.

Input: Matrix D

Output: Matrices S, T and P

```
    Initialization :        # $1^{st}$ column, i.e., $j = 1$
for $i = 1$ to $M$ do
    $S(i,1) = D(i,1)$
    $T(i,1) = 1$
    $P(i,1) = i$
end for
    Path tracing :        # $1^{st}$ two rows, i.e., $i = 1,2$
for $i = 1$ to 2 do
    for $j = 2$ to $N$ do
        $S(i,j) = S(i,j-1) + D(i,j)$
        $T(1,j) = j$
        $P(1,j) = i$
    end for
end for
    Path tracing :        # For the rest: $i > 2$ and $j > 1$
for $i = 3$ to $M$ do
    for $j = 2$ to $N$ do
        $\Omega = \{(i,j-1),(i-1,j-1),(i-2,j-1)\}$
        $(r,s) = \operatorname{argmin}_{(a,b)\in\Omega} S(a,b)$        # Selecting the predessor
        $S(i,j) = S(r,s) + D(i,j)$
        $T(i,j) = T(r,s) + 1$
        $P(i,j) = P(r,s)$
    end for
end for
```

Algorithm 5 An algorithm for matrices S, T and P computation from the matrix D for *asymmetrical* local constraint, i.e., LC_3 (as specified in Figure 3.6 (c).)

Input: Matrix D

Output: Matrices S, T and P

```
Initialization :        # 1^{st} two columns, i.e., $j = 1, 2$
for $j = 1$ to 2 do
    for $i = 1$ to $M$ do
        $S(i, j) = D(i, j)$
        $T(i, j) = j$
        $P(i, j) = i$
    end for
end for
Path tracing :        # 1^{st} row, i.e., $i = 1$
for $j = 2$ to $N$ do
    $S(1, j) = S(1, j-1) + D(1, j)$
    $T(1, j) = j$
    $P(1, j) = 1$
end for
Path tracing :        # For the rest: $i > 1$ and $j > 2$
for $i = 2$ to $M$ do
    for $j = 3$ to $N$ do
        $\Omega = \{(i-1, j-2), (i-1, j-1), (i-1, j)\}$
        $(r, s)$ = $\underset{(a,b)\in\Omega}{\text{argmin}}\ S(a, b)$        # Selecting the predessor
        $S(i, j) = S(r, s) + D(i, j)$
        $T(i, j) = T(r, s) + 1$
        $P(i, j) = P(r, s)$
    end for
end for
```

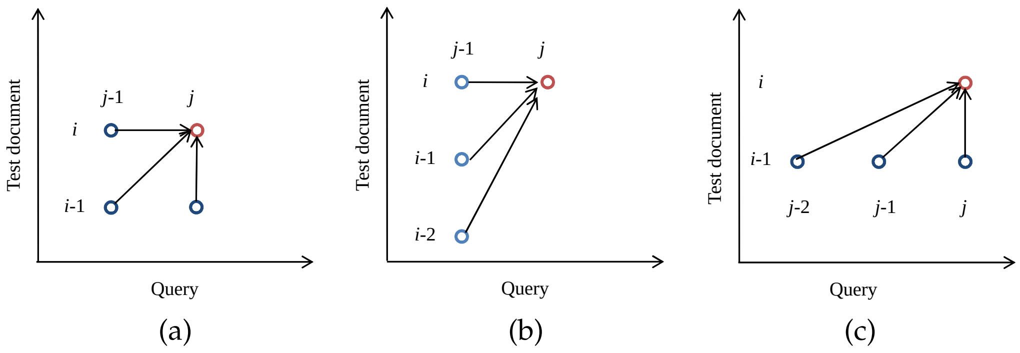


Figure 3.6: Types of local constraints (LC) used: (a) LC_1, (b) LC_2, and (c) LC_3. After [2].

Table 3.3: Performance of SWS 2013 QbE-STD system for various local constraints (in TWV). The numbers in the brackets indicate ATWV. After [2]

Feature Vector	Dev Set			Eval Set		
	LC_1	LC_2	LC_3	LC_1	LC_2	LC_3
CZ	0.348	**0.375**	0.356	0.318 (0.313)	0.342 (**0.340**)	0.324 (0.323)
HU	0.357	**0.374**	0.368	0.331 (0.325)	0.341 (**0.340**)	0.342 (0.336)
RU	0.367	**0.386**	0.378	0.340 (0.338)	0.359 (**0.357**)	0.347 (0.347)

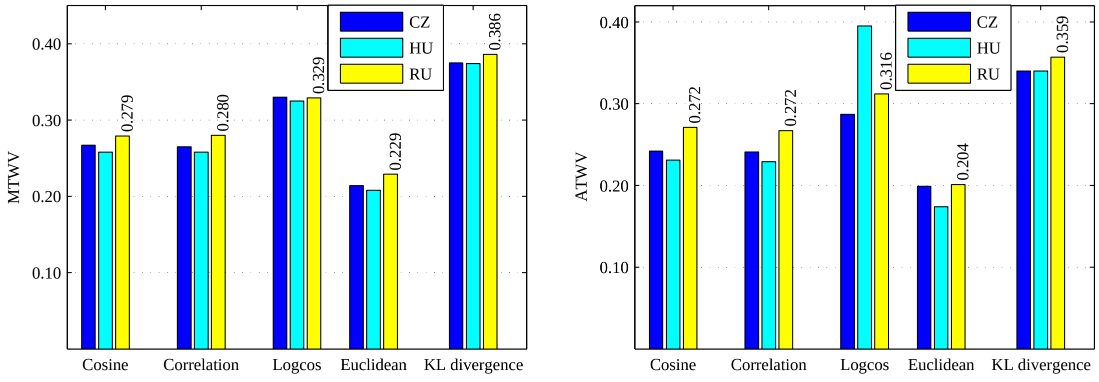
locality consideration during computation of accumulated distance matrix. The feature alignment is performed by similarity matching of consecutive features by considering different local constraints. Table 3.3 shows the performance of different local constraints for CZ, HU, and RU posteriorgrams. It can be seen from Table 3.3 that the local constraint LC_2 gives relatively better performance than the LC_1 and LC_3 [2,53]. The local constraint LC_2 allows more frames to be inserted from test utterance to the query, which is suited for QbE-STD [53]. Hence, we have used, LC_2, for the rest of the experiments reported in this thesis if not specified.

We do not perform length normalization for every transition (i.e., on-the-fly length normalization) as opposed to the study reported in [53, 148]. It was observed that on-the-fly length normalization (OLN) prefers the longer alignment path over shorter alignment path [45, 148]. In subDTW, only single DTW is performed for each query and test utterance pair. The query can start at any time instant within test utterance. The warping path is selected based on the adjacent accumulated distances. The on-the-fly length normalization (OLN) is performed by selecting the warping path based on adjacent accumulated local distance normalized by the path length. The performance w.r.t. OLN using local constraints LC_1 is shown in Table 3.4. It can be observed that a slight improvement in LC_1 is obtained. However, the performance is still not better than the LC_2.

In the search algorithm, the matrix S is normalized independently of the ma-

Table 3.4: Performance of SWS 2013 QbE-STD for on-the-fly length normalization (OLN) for local constraint LC_1 (in MTWV). The numbers in the brackets indicate ATWV. After [2]

Feature Vector	Dev Set		Eval Set	
	no OLN	OLN	no OLN	OLN
CZ	0.348	**0.357**	0.318 (0.313)	0.324 (**0.322**)
HU	0.357	**0.359**	0.331 (0.325)	0.334 (**0.332**)
RU	**0.367**	0.360	0.340 (0.338)	0.349 (**0.346**)



(a)

(b)

Figure 3.7: Performance of various dissimilarity functions on SWS 2013 QbE-STD task: (a) Dev set, and (b) Eval set. The number on the top of bars indicate MTWV. After [2].

trix T and then the matrix T is applied to normalized matrix S in order to compute the DTW alignment distance. For the local constraint, LC_2, the optimization of the matrix S does not involve the normalization by the length matrix T. This is because of the fact that for LC_2, $T(a,b) = b = j-1$ and hence, for the local constraint, LC_2, the optimization of the matrix S does not involve the normalization by the length matrix T. This is because of the fact that for LC_2, $T(a,b) = b = j-1$ and hence,

$$(r,s) = \underset{(a,b)\in\Omega}{\operatorname{argmin}} \frac{S(a,b)}{T(a,b)} = \underset{(a,b)\in\Omega}{\operatorname{argmin}} \frac{S(a,b)}{j-1} = \underset{(a,b)\in\Omega}{\operatorname{argmin}} S(a,b). \tag{3.14}$$

After complete path tracing, the accumulated distance matrix S is normalized by the path length matrix T to compute the DTW alignment cost. Thus, for local constraint, LC_2, OLN does not make any difference because of the constant denominator in minimization.

3.7 Effect of Dissimilarity Functions

Various studies in QbE-STD used different dissimilarity functions (such as, cosine distance, correlation distance, log-cosine distance, Euclidean distance and KL-divergence) to compute the local distance matrix. For example, the *logcosine* distance was used in [39, 77, 121, 126, 148]. The cosine measure was used in [103, 111, 148]. The Pearson's correlation coefficient was used in [66, 148]. The study presented in [52,86,98] uses KL divergence metrics. As shown in the Figure 3.7, the symmetrical version of KL divergence gave relatively better performance over other distance metric (for both Dev and Eval sets), followed by the negative logarithm of cosine similarity. KL divergence is better suited because the posterior probabilities have a flatter distribution [86]. The performance obtained using cosine and correlation distance metric is similar. The Euclidean distance metric is not suitable for posterior template matching, as suggested in [81].

3.8 Chapter Summary

In this chapter, we discussed the experimental setup for QbE-STD systems used in this thesis. The front-end component converts speech signal into frame-level representation (such as, acoustic features or posteriorgram of acoustic features). It also performs removal of non-speech regions with the help of SAD. The searching subsystem performs matching between the query representation and the representation of the utterance. Detection subsystem pools the distances from several detection candidates and normalizes them, which are interpreted as detection scores. Performance evaluation metrics are mainly p@N, recall and MAP for ranked evaluation task. For unranked evaluation task, the performance is evaluated with MTWV and C_{nxe}^{min}. In the next chapter, we will discuss the representation perspective for the design of QbE-STD system.

CHAPTER 4

Representation Perspective

4.1 Introduction

This chapter presents the details of acoustic representation for QbE-STD system. As discussed in Section 2.4, the front-end subsystem transforms speech (acoustic) signal into an appropriate representative vector, which is used by the search subsystem for audio matching. The acoustic representation should be speaker and channel-invariant and resemble the similar behavior for the same spoken content (same word or same sentence). In this chapter, we will discuss the use of Vocal Tract Length Normalization (VTLN) warping factor estimation approach and mixture of GMMs approach to representing the speech signal with restricting the posterior probability w.r.t. broad phoneme classes for the QbE-STD task. Section 4.2 discusses the VTLN warping factor estimation with GMM and VTL-warped Gaussian posteriorgram. Section 4.3 presents the mixture of GMMs framework for posteriorgram design.

4.2 Vocal Tract Length Normalization (VTLN)

It has been studied in the speech processing literature that for a uniform vocal tract model, the formants of the vocal tract are *inversely* related to the length of the vocal tract [190]. The formant frequencies of the vocal tract are given by [190]:

$$F_n = \frac{(2n-1)v}{4L}, n = 1, 2, \cdots, \tag{4.1}$$

where L = length of the vocal tract (which is typically *13* cm to *18* cm [191]) and v = velocity of the sound wave ($\approx$ *344* m/s, at sea-level and *70*^{o} F [190]). For instance, formant frequencies of two speakers, namely, speaker A and speaker B having an average vocal tract length, L_A and L_B, respectively, are $F_A \propto \frac{1}{L_A}$ and $F_B \propto \frac{1}{L_B}$. This results into $F_A = \alpha_{AB} F_B$, where α_{AB} represents VTLN warping

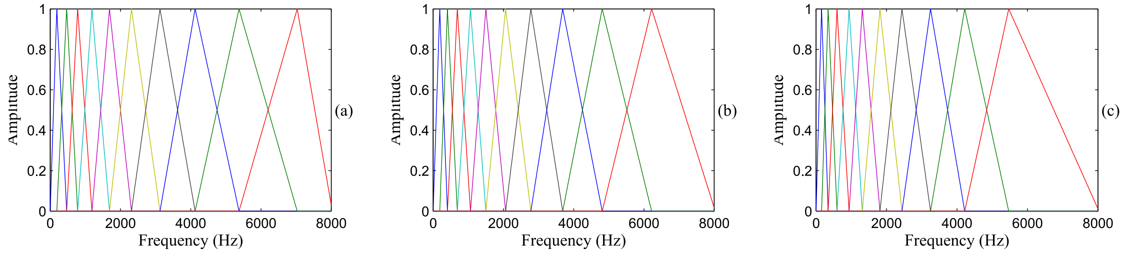


Figure 4.1: Mel filterbank for different VTLN warping factors: (a) $\alpha = 0.88$, (b) $\alpha = 1$ and (c) $\alpha = 1.12$.

factor associated with only two speakers, namely, speaker A and speaker B. This is also referred to as *uniform scaling* in the frequency domain spectra vowel produced (spoken) by two different speakers [192]. In practice, the VTLN warping factor is estimated from each utterance w.r.t. to a general speaker model. Human vocal tract length can vary from nearly *13* cm for adult female up to *18* cm for adult male [191]. Due to this, formant frequencies can deviate by *25* % among various speakers. To reflect this deviation, the VTLN warping factor is generally taken from a set of *13* distinct values (at equally-spaced points between *0.88* and *1.12*) [191]. The introduction of the VTLN warping factor creates an adjustment in the frequency analysis to cope with such spectral scaling variations. In general, this is performed by considering different versions of the Mel filterbank (whose center frequencies are scaled linearly). Figure 4.1 shows the Mel filterbank for different VTLN warping factors (namely, $\alpha = 0.88, 1$, and *1.12*). In practice, warping factors are obtained via statistical modeling framework, i.e., Maximum Likelihood Estimation (MLE) [193]:

$$\hat{\alpha} = \underset{0.88 \leq \alpha \leq 1.12}{\operatorname{argmax}} \; P(X^{\alpha} | \lambda_T, W). \tag{4.2}$$

Since the closed form expression of Eq. (4.2) is not available, MLE is computed for all the different warped feature vector X^{α} against model λ_T for a given transcription, W. In this thesis, we refer to this estimation approach as HMM-based VTLN warping factor estimation (because this approach requires HMM model λ_T and the transcription W). The transcription is either known (i.e., supervised) or estimated first. This framework computes the alignment cost of features to the states of HMM. To investigate the effectiveness of warping factor estimation, we estimate the α from *10* male and *10* female subjects of Gujarati (G) and Marathi (M) languages [194]. As shown in Figure 4.2, higher values of α is for male speakers (indicating longer VTL) and smaller values of α are for female speakers (indicating shorter VTL).

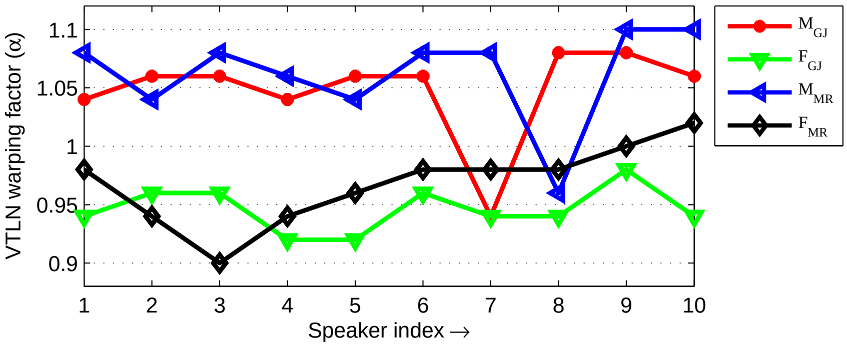


Figure 4.2: Plot of VTLN warping factor α for 20 speakers (*10* male and *10* female). M_{GJ} = Gujarati male speakers, F_{GJ} = Gujarati female speakers, M_{MR} = Marathi male speakers and F_{MR} = Marathi female speakers. After [194].

4.2.1 Prior Studies in VTLN

In previous work, VTLN was addressed by normalizing the vocal tract shape considering the stationary (steady-state) part of American English vowels [54]. VTLN warping factor estimation was mostly performed by considering the estimation of a formant frequency, in particular, the 3^{rd} formant frequency [195, 196]. Data-driven methods, such as, elastic registration [197] and dynamic programming(DP)-based method, such as, Dynamic Frequency Warping (DFW) [198] were used for VTLN warping factor estimation. The study presented in [199] presents a scale transform for speaker normalization, whereas the study reported in [200] suggested the use of a conformal map (i.e., an allpass transformation) to perform VTLN in the cepstral-domain instead of frequency-domain. Linear transformations in the cepstral-domain were also used for VTLN in [201, 202] (called LT-VTLN) with less computational complexity and easy Jacobian matrix computation.

A Vector Quantization (VQ) codebook approach was suggested for VTLN for text-independent speaker normalization [203]. Frequency warping-based VTLN was suggested in [204]. The VTLN warping factor estimation problem was posed as frequency translation (shifting) factor estimation under the MLE framework in [205]. Spectral variations in speech among different speakers are phone-dependent and cannot be fully captured using a single warping factor for a single utterance. In order to capture the dynamics of the VTLN warping factor along utterances, frames are converted into a sequence of regions and over each region, VTLN warping factor is estimated [206].

Most of these approaches for VTLN warping factor estimation approaches are used for ASR tasks. This framework is possible when a reference phoneme-level transcription is given along with the speech signal. The phoneme-level transcription is not available, when considering unsupervised QbE-STD task. In the absence of transcription (i.e., in unsupervised scenarios, such as the study reported in [78]), speech sound units are clustered to form a reference transcription. Since these new sound units are generated from the acoustic observation after segmentation, they were called Acoustic Segment Model (ASM) [78]. This process generates a transcription and finds the VTLN warping factor, α, under an MLE framework. The scope of the present work is to exploit the capability of GMM to estimate VTLN warping factor and its application for the QbE-STD task. The approach is rooted from the fast VTLN approach and the two-pass approach presented in [207, 208]. However, we attempt to perform multiple passes, i.e., an iterative framework of VTLN warping factor estimations for the QbE-STD task. Later, we considered a reduced number of frames (belonging to vowels) for VTLN warping factor estimation.

The conventional method (such as, Lee-Rose method [193]) for VTLN warping factor estimation requires a phoneme-level transcription, whereas the GMM framework does not require a phoneme-level transcription. In this work, we refer to the Lee-Rose method of VTLN warping factor estimation as HMM-based VTLN warping factor estimation, which is supervised as it requires transcription. In addition, this approach uses GMM that can also be *exploited* for Gaussian posteriorgram design. Hence, the presented work exploits trained GMM for VTLN warping factor estimation and then use VTL-warped features to re-train the GMM Gaussian posteriorgrams, used for QbE-STD tasks. The major contributions in this work are as follows:

- A GMM-based VTLN warping factor estimation is presented, which does not require manual transcription.
- The correlation between VTLN warping factor estimation using GMM (unsupervised) and HMM (supervised) framework is analyzed.
- Three different feature extraction schemes to incorporate linear VTLN, namely, MFCC, PLP and MFCC-TMP are used.
- The iterative approach for VTLN warping factor estimation under GMM framework and the likelihood values at the different stage of estimation procedure is presented.
- The application of GMM-based VTLN warping factor estimation to phoneme recognition task is presented.
- The QbE-STD tasks are performed with various experimental conditions, such as, the number of iterations of VTLN warping factor estimation, multiple examples of spoken query, the score-level fusion of various search sys-

tems and reduced number of feature vectors for VTLN warping factor estimation.

4.2.2 GMM-based VTLN

A GMM-based framework differs from an HMM-based framework in terms of the objective function used to estimate the VTLN warping factor. In HMM model, *61* phoneme units are modeled as three hidden states and each state of HMM is modeled with eight mixture component GMM. A GMM consists *64*-mixture components. The current work is focused on linear warping factor estimation, which is implemented in the frequency-domain. To that effect, we used Gaussian Mixture Model (GMM) likelihood scores to obtain the VTLN warping factor. The belief behind the GMM based VTLN is GMM trained over large number of speakers may contain speaker generalized behavior. The spectral variation w.r.t. the generalized model can be useful to adjust filterbank for each speaker. Furthermore, we kept on doing iteratively to obtain further normalization in spectral variation w.r.t. to general speaker model. The process of VTLN warping factor estimation and modified posteriorgram feature extraction is as follows.

1. *Feature Extraction*: Compute warped features, i.e., $\mathbf{x}_t^{\alpha}$ that carry information from different warping factors, namely, $\alpha = 0.88, 0.90, \cdots, 1.12$. Note that the number of distinct values of α is user-defined and can be empirically decided.
2. *Initial Training*: Train the GMM without warped features, i.e., $\mathbf{x}_t^{\alpha}$, where $\alpha = 1$, i.e., no VTLN warping. Let the initial GMM model be
$\lambda_{init} \sim (\mu_{init}, \Sigma_{init}, w_{init})$.
3. *VTLN warping factor estimation*: The likelihood is computed for all the different warped feature vectors $\mathbf{x}_t^{\alpha}$ against the initial model, λ_{init}, i.e.,

$$\hat{\alpha} = \underset{0.88 \leq \alpha \leq 1.12}{\text{argmax}} \; P(X^{\alpha}|\lambda_{init}). \tag{4.3}$$

   VTLN warping factor is chosen by performing grid search within 0.88 to 1.12.
4. *Retraining GMM*: GMM is re-trained on this optimal warped features, i.e., $\mathbf{x}^{\hat{\alpha}}$. This new model $\lambda_r \sim (\mu_r, \Sigma_r, w_r)$ is different from the earlier GMM model λ_{init}.
5. *Posteriorgram Computation*: Now, the VTLN warping factors of test and query features are estimated against the new GMM model λ_r. Based on the estimated warping factors, Gaussian posteriorgrams are computed.

VTL-warped Gaussian posteriorgram obtained are used for QbE-STD task. Figure 4.3 shows the overall block diagram for VTLN-based Gaussian posteriorgram feature extraction. In the next sub-Section, we will investigate the phone recognition performance of VTLN warping factor estimation. The idea of GMM-based VTLN warping factor estimation can be explained as follows. Let VTL-warped features be $X^{\alpha}, 0.88 \leq \alpha \leq 1.12$. Initially, the GMM is trained on unwarped features, i.e., $\alpha = 1$ and hence, ($X^{\alpha} \equiv X^{1}$), i.e.,

$$\lambda_{init} = \underset{\lambda}{\text{argmax}}\, P(X^{1}|\lambda). \tag{4.4}$$

Now, VTLN warping factors are estimated based on MLE, i.e.,

$$\hat{\alpha} = \underset{0.88 \leq \alpha \leq 1.12}{\text{argmax}}\, P(X^{\alpha}|\lambda_{init}). \tag{4.5}$$

In the next iteration, we consider VTL-warped features to train GMM,

$$\lambda^{(1)} = \underset{\lambda}{\text{argmax}}\, P(X^{\hat{\alpha}}|\lambda). \tag{4.6}$$

This implies $P(X^{\hat{\alpha}}|\lambda^{(1)}) \geq P(X^{1}|\lambda_{init})$. Thus, maximization in likelihood results into better Gaussian posteriorgram representation in the following iterations.

Next, we will investigate the relation between two VTLN warping factor estimates (using both GMM and HMM approach) on MFCC feature sets. The VTLN warping factor is estimated using the GMM and HMM-based approaches using the train set of TIMIT, consisting *3696* utterances. We employed linear frequency scaling to implement VTLN, i.e., α = *0.88, 0.90, $\cdots$,1.12*. Fig. 4.4 displays the mapping between these two VTLN warping factor estimates using a supervised Lee-Rose method [191] and the unsupervised method. The relatively diagonal darker band in Figure 4.4 indicates that most of the warping factors obtained through these two techniques are nicely correlated with each other. Moreover, it was observed that around *35* % utterances have the same VTLN warping factors (falling on the line $y = x$ of Figure 4.4) for both HMM-based estimation and GMM-based estimation. This analysis shows the potential of GMM-based VTLN warping factor estimation under the absence of transcription. TIMIT corpus contains more number of male speakers (for which $\alpha > 1$) than the female speakers. Now, due to the differences between the GMM and HMM-based VTLN warping factor estimates, VTLN warping factors that estimated from GMM framework are distributed more to the left of $y = x$ line. This results in an upward bending/tilting line (as shown in Figure 4.4).

To understand the effect of multiple utterances in VTLN warping factor estimation, we estimate VTLN warping factor per speaker for TIMIT train set. The

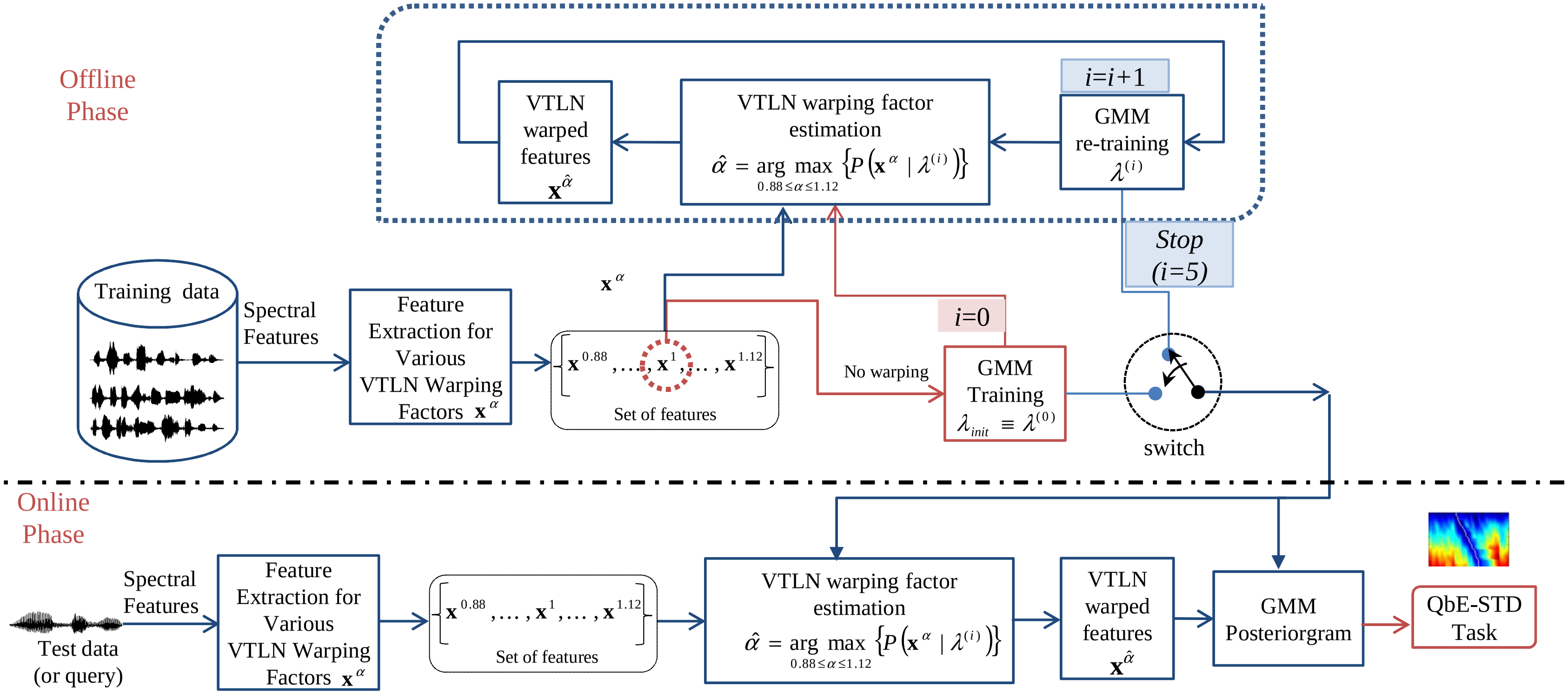

Offline
Phase
Training data
Spectral
Features
Feature
Extraction for
Various
VTLN Warping
Factors $\mathbf{x}^{\alpha}$
$\mathbf{x}^{\alpha}$
$\mathbf{x}^{0.88}, \ldots, \mathbf{x}^{1}, \ldots, \mathbf{x}^{1.12}$
Set of features
No warping
i=0
GMM
Training
$\lambda_{init} \equiv \lambda^{(0)}$
VTLN
warped
features
$\mathbf{x}^{\hat{\alpha}}$
VTLN warping factor
estimation
$\hat{\alpha} = \arg\max_{0.88 \le \alpha \le 1.12} \{P(\mathbf{x}^{\alpha} \mid \lambda^{(i)})\}$
i=i+1
GMM
re-training
$\lambda^{(i)}$
Stop
(i=5)
switch
Online
Phase
Spectral
Features
Test data
(or query)
Feature
Extraction for
Various
VTLN Warping
Factors $\mathbf{x}^{\alpha}$
$\mathbf{x}^{0.88}, \ldots, \mathbf{x}^{1}, \ldots, \mathbf{x}^{1.12}$
Set of features
VTLN warping factor
estimation
$\hat{\alpha} = \arg\max_{0.88 \le \alpha \le 1.12} \{P(\mathbf{x}^{\alpha} \mid \lambda^{(i)})\}$
VTLN
warped
features
$\mathbf{x}^{\hat{\alpha}}$
GMM
Posteriorgram
QbE-STD
Task

Figure 4.3: A schematic block diagram of proposed unsupervised GMM-based VTLN warping factor estimation and Gaussian posteriorgram feature extraction. The dotted box in figure 4.3 shows an iterative approach for VTLN warping factor estimation.

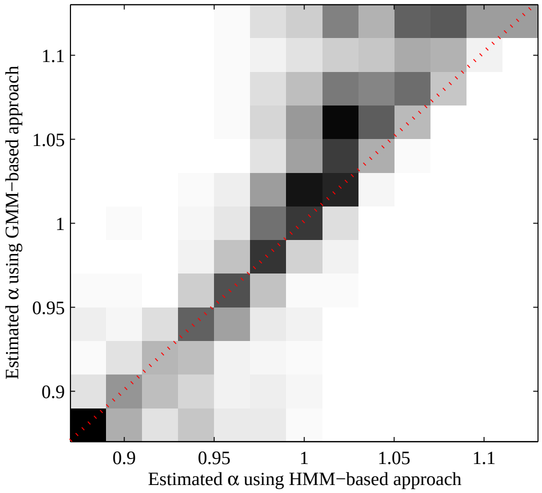


Figure 4.4: Estimated values of VTLN warping factor using two different methods, namely, HMM (supervised) and GMM (unsupervised). The red dashed line indicates the line $y = x$. After [1].

effect of a number of utterances in VTLN warping factor estimation is shown in terms of the difference in α between utterance wise and speaker wise. It can be observed from Figure 4.6 that the estimated VTLN warping factor per speaker and an utterance is too close and the difference is between 0.02 to 0.03. If the number of utterances or the duration increases, the difference between utterance-specific and speaker-specific VTLN warping factor gets reduced. That means, we get the almost same warping factor for multiple utterances.

4.2.3 Iterative Approach for VTLN

As discussed earlier in sub-Section 4.2.2, the iterative scheme of VTLN warping factor estimation leads to an increase in the likelihood of the training data. Algorithm 6 presents the details of the GMM-based approach used for VTLN warping factor estimation and corresponding Gaussian posteriorgrams extraction. The plot of the values of log-likelihood w.r.t. the iteration index is shown in Figure 4.5. This plot is for MFCC cepstral features and *128* mixtures of components. To compute log-likelihood, we start with no VTLN warping (i.e., $\alpha^{(0)} = 1$) and estimate initial GMM model $\lambda^{(1)} = \lambda_{init}$ (please refer eq. (4.4), the log-likelihood $\log(P(X^{\alpha^{(0)}}|\lambda^{(1)}))$ is computed. Now, new VTLN warping factors are estimated as per eq. (4.5) and the log-likelihood $\log(P(X^{\alpha^{(1)}}|\lambda^{(1)}))$ is computed.

In the GMM-based approach for VTLN warping factor estimation, we initially build a GMM on unwarped (i.e., $\alpha = 1$, no VTLN) features and estimate the appropriate VTLN warping factor using MLE.

Initially, we don't have speaker labels or phonetic transcription and hence,

Algorithm 6 An algorithm for proposed iterative approach for unsupervised GMM-based VTLN warping factor estimation and Gaussian posteriorgram extraction.

```
Input: VTL-warped features: Training $X^{\alpha}_{tr}$, Testing $X^{\alpha}_{te}$, Number of iterations $N_{iter}$.
Output: Optimal VTLN warping factor and corresponding Gaussian posteriorgram.
Initialization :
Initial GMM training on $\alpha = 1$. (No VTLN)
Build GMM $\lambda_{init}$ on $X^{\alpha}_{tr}$, ($\alpha = 1$), i.e.,
  $\lambda_{init} \sim (\mu_{init}, \Sigma_{init}, w_{init})$.
VTLN warping factor estimation and re-training :
$k = 0$                # set iteration index counter to 0
while ($k \geq N_{iter}$) do
    for each training utterance $i$ do
        Estimate VTLN warping factor using
        $\hat{\alpha}_i = \underset{\alpha}{\text{argmax}}\ P(X^{\alpha}_{tr,i}|\lambda_{init})$.
    end for
    Build GMM on optimal VTL-warped features ($X^{\hat{\alpha}}_{tr}$), i.e., $\lambda_r \sim (\mu_r, \Sigma_r, w_r)$.
    $k = k + 1$        # increment iteration index counter.
    $\lambda_{init} \leftarrow \lambda_r$        # store new model as old model.
end while
VTLN warping factor estimation and Gaussian posteriorgram computation for testing database :
for each testing utterance $i$ do
    Estimate VTLN warping factor using
    $\hat{\alpha}_i = \underset{\alpha}{\text{argmax}}\ P(X^{\alpha}_{te,i}|\lambda_r)$.
    Compute Gaussian posteriorgram using
    $GP(X^{\hat{\alpha}_i}_{te,i}) = [P(C_1|X^{\hat{\alpha}_i}_{te,i}), \cdots, P(C_{NG}|X^{\hat{\alpha}_i}_{te,i})]^T$,
    where $P(C_p|X^{\hat{\alpha}_i}_{te,i}) = \frac{w_r^p \mathcal{N}(X^{\hat{\alpha}_i}_{te,i};\mu_r^p,\Sigma_r^p)}{\sum_j^{NG} w_r^j \mathcal{N}(X^{\hat{\alpha}_i}_{te,i};\mu_r^j,\Sigma_r^j)}$.
end for
```

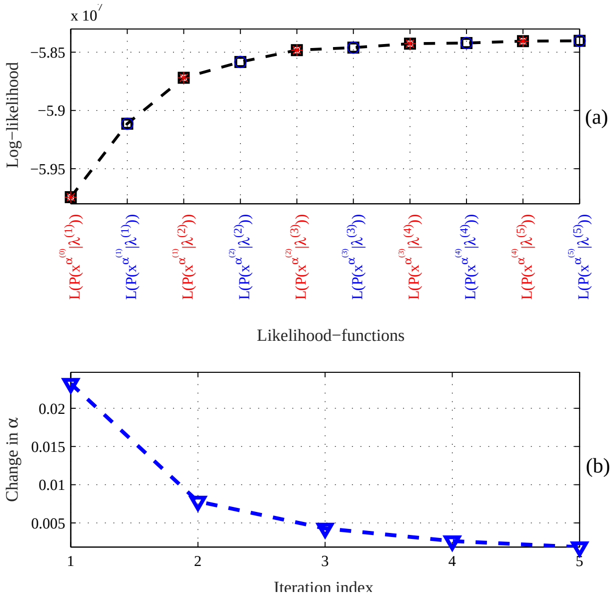


Figure 4.5: (a) Log-likelihood values w.r.t.the iteration index, and (b) the change in VTLN warping factor estimation from previous estimates. The red points (in Fig. 4.5 (a)) indicate log-likelihood after GMM training, whereas the blue points (in Fig. 4.5 (a)) indicate log-likelihood after VTLN warping factor estimation.

we started with unwarped features. We assumption that GMM trained with unwarped features captures the information from different speakers and it captures speaker-independent information. The proposed approach of VTLN warping factor estimation is rooted from the study presented in [207]. As an initial seed, VTLN warping factors from different speakers are taken as $\alpha = 1$, then the model is trained and GMM is retrained on warped features. Again new VTLN warping factors are estimated with retrained GMM. This process is continued for 5 iterations. As iteration increases, the difference between VTLN warping factors estimated with current iteration and previous iteration gets reduced. Figure 4.5 (b) also indicates the change in VTLN warping factor estimates w.r.t. previous estimates. It can be observed that as number of iterations increase,the difference between VTLN warping factors estimated with current iteration and previous iteration gets reduced.

In the next sub-Section, we observe the effectiveness of the VTLN warping factor estimation for phoneme recognition task.

4.2.4 Results for Phoneme Recognition

We considered the TIMIT database without /sa/ sentences that are common across all the speakers and these sentences may bias the results. We used the training set from TIMIT for HMM monophone training for a phoneme recog-

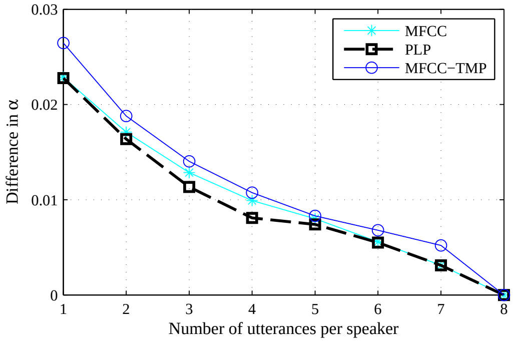


Figure 4.6: Effect of number of utterances in GMM-based VTLN warping factor estimation.

nition task. The HTK was used to perform ASR (phoneme recognition) experiments [180]. The phoneme recognition is evaluated in terms of % phoneme accuracy [180]. We used MFCC and PLP as acoustic features that are extracted using HTK. The details of feature extractions were presented in sub-Section 3.3.1.

Table 4.1: Effect of VTLN warping factor estimation on phoneme recognition performance (% Phoneme Accuracy)

VTLN	×	L (TRAN)	L (DEC)	P (1)	P (2)	P (3)	P (4)	P (5)
MFCC	67.44	**70.62**	68.90	69.02	69.67	69.73	69.80	69.81
PLP	67.25	**70.89**	68.70	69.45	70.16	70.19	70.21	70.26

(× = No VTLN, L = Lee-Rose VTLN and P=Proposed iterative approach for GMM-based VTLN warping factor estimation, TRAN= with actual phonetic transcription, DEC=with decoded phonetic transcription. The numbers in brackets indicate the iteration index in proposed iterative GMM-based VTLN warping factor estimation.)

Table 4.1 shows the performance of the VTLN warping factor estimation on phoneme recognition task in terms of % Phoneme Accuracy. We consider context-independent monophone models trained over *61* TIMIT phoneset, then merged them into *39* phoneset as suggested in [209]. A bi-gram phoneme-based language model is trained under HTK framework. In HMM-based approach, we considered the two cases for VTL-warping factor estimation. The *Transcription (TRAN)* case , in which the exact phonetic transcription along with the test utterance is given to estimate VTLN warping factor. The *Decode (DEC)* case, where only test utterance is given without the transcription and the transcription is decoded with the help of trained HMM model to estimate the VTLN warping factor. Better phoneme recognition accuracy with the *TRAN* case than the *DEC* case indicates the dependency of transcription because the errors introduced while decoding might result into incorrect VTLN warping factor estimation. In GMM-based approach, we retrain the GMM for new VTL-warped features and use VTL-warped test audio to decode. It can be observed that the GMM-based VTLN warping factor estimation improves the performance due to this VTLN-based speaker normalization. It can be observed that as a number of iterations increases, the performance of VTL-warped features also increases. The improvements in phoneme accuracy saturates as iteration increases to 4 or 5. GMM-based approach gives better phoneme accuracy than the DEC case. However, poor phoneme accuracy (across all the iterations considered here) than the TRAN case. The better phoneme recognition performance with the GMM-based approach for VTL-warping factor estimation over no VTLN case, motivated the authors to exploit unsupervised GMM-based framework for a QbE-STD task.

4.2.5 Experimental Results

The performance obtained using GMM-based approach is better than no VTLN is applied. This is analyzed in Figure 4.7. It shows accumulated distance matrix, when the query is present in test utterance. In this example, we consider two queries, namely, *government* and *meeting* from TIMIT corpus. The corresponding detections for Gaussian posteriorgram and VTL-warped Gaussian posteriorgram are shown by blue and red arrows, respectively. The green arrow corresponds to the actual endpoint (i.e., the ground truth) of the query within test utterance. In this example, it can be observed that Gaussian posteriorgram shows the wrong detection, whereas VTL-warped Gaussian posteriorgram detection falls very close to the ground truth. It can be observed that VTLN Gaussian posteriorgram exhibits more similarity towards the actual location of the query within test utterance and hence, can be useful for the QbE-STD task. More results on TIMIT dataset for various experimental conditions are given in Appendix A. In the next sub-Section, we will discuss the experimental results for SWS 2013 dataset.

This sub-Section discusses experimental results on MediaEval SWS 2013. With 128 mixture components, we investigate the effect of VTL-warped GP. Figure 4.8 shows the performance for various numbers of iterations used in VTLN warping factor estimation. It can be observed that VTLN improves the QbE-STD performance. As discussed earlier in sub-Section 4.2.3, we used 5 iterations in iterative VTLN warping factor estimation framework. It can be observed from Figure 4.8 that performance of QbE-STD system improves as the number of iterations increases. As discussed in sub-Section 4.2.3, the stopping criteria can be set at 2^{nd} or 3^{rd} iteration. The performance does not vary significantly after that. The stopping criteria in iterative approach can be set by examining MTWV in Dev set. It can

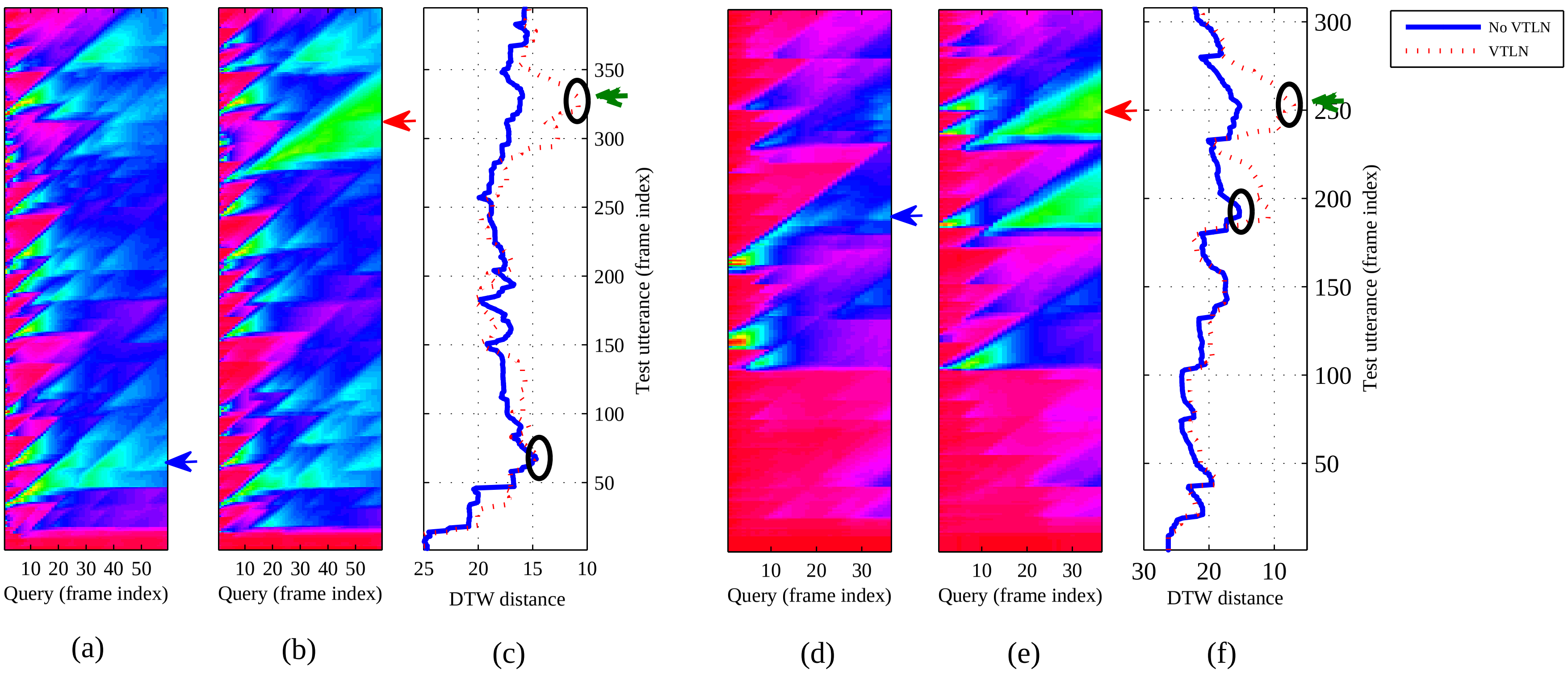
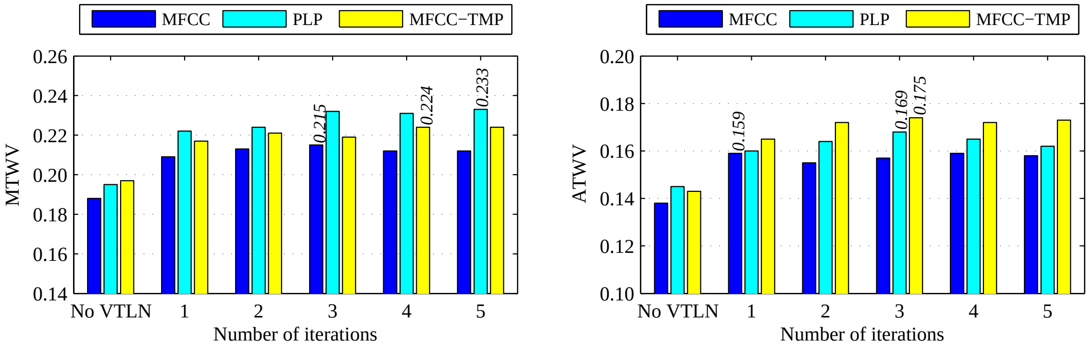


Figure 4.7: Dissimilarity analysis with VTL-warped Gaussian posteriorgram. Accumulated distance matrix for query /*government*/ with (a) no-VTLN Gaussian posteriorgram (b) VTLN Gaussian posteriorgram, and (c) DTW distance obtained in each case. (d), (e), and (f) show a similar analysis for a different query, /*meeting*/. The arrows indicate the minimum detection and the circles indicate corresponding DTW distance values. The accumulated distance matrix S is normalized by the matrix counting T for better visualization.



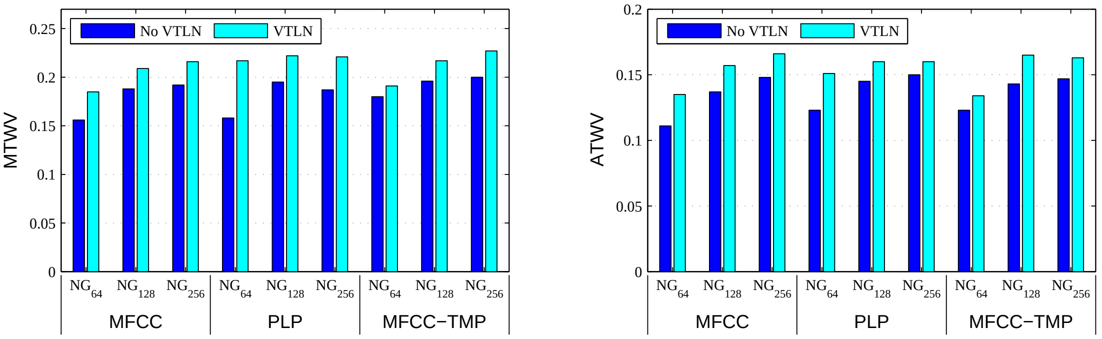
(a) (b)

Figure 4.8: Effect of the number of iterations for VTLN warping factor estimation on SWS 2013 QbE-STD performance. (a) MTWV for Dev set, (b) ATWV for Eval set. The number on the top of bars indicate MTWV.

be observed that from Figure 4.8 that MTWV is relatively higher at 3^{rd} iteration. Hence, *3* iterations would be a reasonable choice, and we may stop after 3 iterations. Interestingly, for Eval set MTWV at 3^{rd} iteration is higher than the no VTLN case. It can also be noted that the different feature sets achieved maximum MTWV at different iterations. This is because of varying acoustical property captured by different feature sets used in this thesis.

4.2.5.1 Effect of Number of Gaussians

The number of Gaussian components in Gaussian posteriorgram plays an important role in QbE-STD task [53, 77, 98]. In this Section, we investigate the effect of the number of mixture components used in VTLN warping factor estimation on QbE-STD task. In particular, we considered *64*, *128*, and *256* mixture components in GMM for both training and VTLN warping factor estimation. It can be observed from Figure 4.9, that performance using the GMM-based VTL-warped posteriorgram is better than the Gaussian posteriorgram. In addition, it can be observed from Figure 4.9, that an increasing number of mixture components improves the performance of a QbE-STD system. This finding matches a previous study reported in [53]. This might be because of the increasing number of clusters (in GMM) that better represents the speech signal at the frame-level. However, increasing number of Gaussians demands additional processing and storage cost and hence, we restrict our experiments till *128* number of clusters. The lower number of Gaussian components might be insufficient to capture the distribution of feature vectors and hence, the Gaussian posteriorgram of 64 components does not give better performance as compared to the Gaussian posteriorgram of 128



(a) (b)

Figure 4.9: Effect of the number of Gaussians on SWS 2013 QbE-STD systems on performance. Results on (a) Dev set, and (b) Eval set.

components.

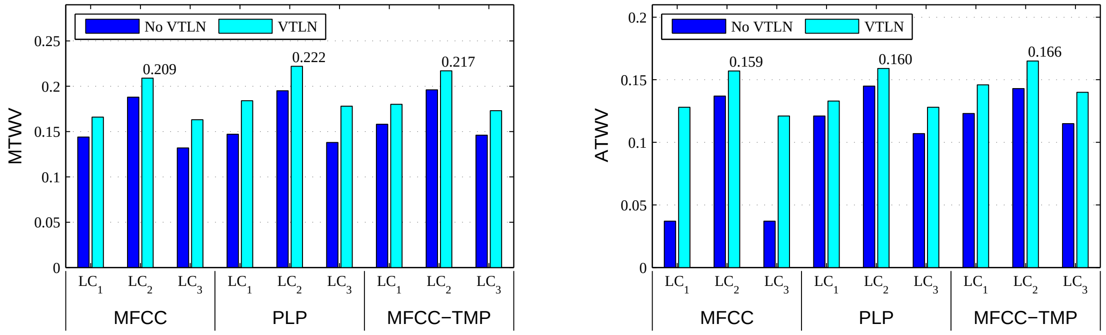
4.2.5.2 Effect of Local Constraints

Figure 3.6 shows three different local constraints for DTW-based searching in QbE-STD. The relative local temporal mismatch between a query and utterance due to different speaking rates by various speakers may require additional treatments in the search algorithm. In particular, the locality constraints considered during DTW distance accumulation has to be adjusted. The feature alignment is performed by similarity matching of consecutive features by considering different local constraints.

Figure 4.10 shows the performance of QbE-STD systems for different local constraints. It can be observed from Figure 4.10 that LC_2 performs relatively better than the other local constraints, probably due to its property of mapping more features along test utterances than the query. Thus, it allows to map more feature vectors from the test utterance than the query, which might be suitable in QbE-STD due to the nature of problem [53], where a test utterance is having longer duration than the query. In the experimental results presented earlier in this thesis, we used local constraint, LC_2 unless not specified. For every local constraint, it can be also observed that VTL-warped Gaussian posteriorgram improves QbE-STD performance over Gaussian posteriorgrams.

4.2.5.3 Multiple Examples per Query

As discussed earlier Section 3.2, SWS 2013 data consists of two sets of queries, namely, normal or basic (which involve only one example per query), and extended (which includes multiple examples per query). It was recommended in



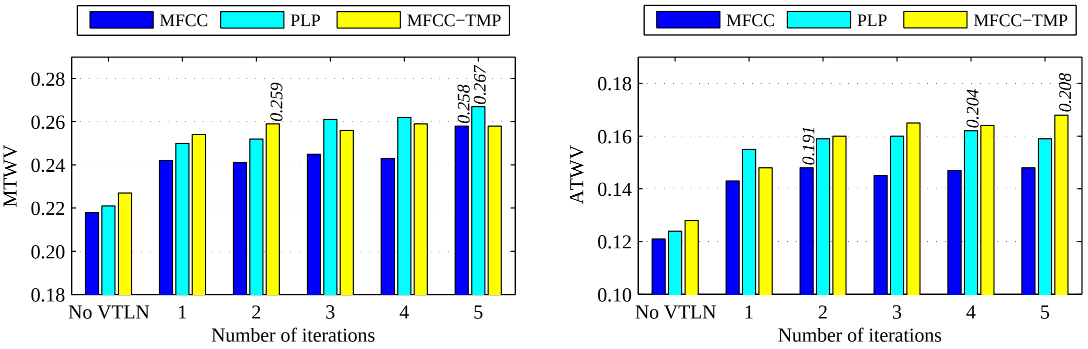
(a) (b)

Figure 4.10: Effect of local constraints (LC) on SWS 2013 QbE-STD systems (in MTWV). Results on (a) Dev set, and (b) Eval set. The number on the top of bars indicate MTWV.

MediaEval's SWS 2013 to use basic query sets. However, multiple examples of a single query capture multiple realizations of the spoken content and might be effective as suggested in [78]. In this Section, experimental results for multiple examples per query are presented. For instance, two languages, namely, Basque and Czech, have *3* and *10* examples, respectively, in Dev and Eval sets of the SWS 2013 dataset. In order to exploit the multiple examples, we prefer to use a simplistic yet effective approach to combine multiple query examples into a single query example as suggested in [74]. In this method, a single average query example is generated by DTW alignments of the multiple queries onto the longest duration query. After this operation, the number of queries in the extended set gets reduced to the number of queries in the normal type (since only one average query represents all the multiple queries). Hence, this process of exploiting the multiple query examples is quite computationally cheaper than considering each example individually. Figure 4.11 shows the performance of using multiple examples per query. It can be observed from Figure 4.11 that after fusing multiple examples, the performance improved for all the feature sets. The MTWV is improved after using VTL-warped Gaussian posteriorgrams (i.e., 4 % absolute and 15.5 % relative improvement for MFCC feature sets).

4.2.5.4 Score-level Fusion of VTL-warped Gaussian Posteriorgrams

The details for score-level fusion for different QbE-STD systems were discussed in sub-Section 3.5.1. Here, we are fusing three different systems (i.e., $NS = 3$), corresponding to three cepstral feature sets, namely, MFCC, PLP and MFCC-TMP (as per the eq.(3.13)). Table 4.2 shows the performance of score-level fusion after (indicated by ✓) and before (indicated by $\times$) VTLN for normal and extended



(a) (b)

Figure 4.11: Effect of the number of iterations for VTLN warping factor estimation on SWS 2013 QbE-STD performance with multiple examples of query: (a) Dev set, (b) Eval set. The number on the top of bars indicate MTWV.

Table 4.2: Performance of score-level fusion for SWS2013 QbE-STD.

Query Type	Features	VTLN	Dev set	Eval set
Normal	ALL	×	0.236	0.178
	ALL_V	✓	**0.256**	**0.194**
Extended	ALL	×	0.267	0.207
	ALL_V	✓	**0.296**	**0.231**
Normal	CZ	P	0.375	0.342
	ALL_V+CZ	P✓	0.445	0.398
	HU	P	0.374	0.341
	ALL_V+HU	P✓	0.445	0.394
	RU	P	0.386	0.359
	ALL_V+RU	P✓	**0.456**	**0.412**

(× = No VTLN, ✓ = VTLN, ALL = All three cepstral representations, ALL_V = All three cepstral representations with VTLN warping, P = phonetic posteriorgram, and P✓ = the score-level fusion of ALL_V with phonetic posteriorgram, + = score-level fusion)

query sets. From Table 4.2, it can be observed that MTWV increases and C_{nxe}^{min} reduces in the case of VTL-warped Gaussian posteriorgrams. VTLN Gaussian posteriorgram improves the Maximum Term Weighted Value (MTWV) by *0.02* (i.e., *2* %) and *0.016* (i.e., *1.6* %), for the Dev and Eval sets, respectively.

The performance of VTL-warped Gaussian posteriorgram can be further improved if the score-level fusion with phonetic posteriorgram is performed. We explore three BUT phoneme recognizers, namely, CZ, HU, and RU for this task. The performance of the score-level fusion of phonetic posteriorgrams (i.e., CZ, HU, and RU) and VTL-warped Gaussian posteriorgram with all the three cepstral representations, namely, MFCC, PLP, and MFCC-TMP, i.e., ALL_V is shown in Table 4.2. It can be seen that score-level fusion gave an improvement in MTWV than the VTL-warped Gaussian posteriorgram. The performance is statistically significant as it is consistently better for all the three phonetic posteriorgrams than the VTL-warped Gaussian posteriorgram. After score-level fusion for phonetic posteriorgram, MTWV of Eval query set is comparable with several SWS 2013 benchmark systems [31] (In particular, as shown in Table 2.3, GTTS: 0.399, L2F: 0.342, CUHK: 0.306, BUT: 0.297, CMTECH: 0.257, IIITH: 0.224, ELIRF: 0.159, TID: 0.093, GT: 0.084, SPEED: 0.059, Proposed (CZ + ALL$_V$): *0.398*, Proposed (HU + ALL$_V$): *0.394*, Proposed (RU + ALL$_V$): *0.412*, Proposed (ALL$_V$) : 0.194).

4.2.5.5 VTLN on Reduced/Expanded Number of Features

In this Section, we investigate the effect of GMM-based VTLN warping factor estimation on reduced number of feature vectors on QbE-STD task. The broad phoneme class of *vowel* are relatively stable and longer as compared to other speech sound units, such as, fricatives, nasals, etc. With this consideration, we reduced the number of features from a query by only considering the frames associated with a vowel. We then perform GMM-based VTLN warping factor for each query. The objective of this experiment is to reduce computational complexity during VTLN warping factor estimation. To detect frames associated to broad phoneme class of vowel, we took the posterior probabilities from broad vowel class. These posterior probabilities are computed from BUT phoneme recognizer. This approach of vowel frame selection from SWS 2013 queries, select about *52* % frames per query. The performance of SWS 2013 QbE-STD task is reported in Table 4.3. It can be observed from Table 4.3 that VTL-warped Gaussian posteriorgram performs better than the Gaussian posteriorgram.

DET curves for VTL-warped Gaussian posteriorgrams obtained from reduced features are shown in Figure 4.12. DET curve indicates that VTLN warping factor estimates obtained through less number of features can also perform better than the Gaussian posteriorgrams. It can also be seen from Figure 4.12 that performance on DET is very much similar for both VTLN warping estimations, i.e., for all the frames and the frames that corresponds to a vowel. This is an important observation that after considering only vowels to estimate VTLN warping factor, the performance of QbE-STD remains *almost the same*. Thus, VTLN warping factor estimation can be executed rapidly as compared to considering entire frames and hence, possibly this approach is computationally less intensive.

In contrast to above experimental condition, we conducted the experiments by considering expanding more training data for VTLN warping factor estimation. To that effect, we pooled the data from QUESST 2014 dataset [29] and used the

Table 4.3: Performance (MTWV) of QbE-STD with reduced number of frames. S= Single spoken example per query, M=Multiple spoken example per query, × = No VTLN, and ✓ = proposed approach. The numbers in the bracket indicate ATWV for Eval set.

Feature sets	# query example	VTLN	Dev set	Eval set
MFCC	S	×	0.188	0.138 (0.137)
		✓	**0.211**	0.155 (**0.154**)
	M	×	0.218	0.161 (0.161)
		✓	**0.240**	0.184 (**0.181**)
PLP	S	×	0.195	0.145 (0.145)
		✓	**0.237**	0.169 (**0.168**)
	M	×	0.221	0.169 (0.164)
		✓	**0.270**	0.207 (**0.204**)
MFCC-TMP	S	×	0.197	0.147 (0.143)
		✓	**0.230**	0.166 (**0.166**)
	M	×	0.227	0.169 (0.168)
		✓	**0.263**	0.199 (**0.196**)

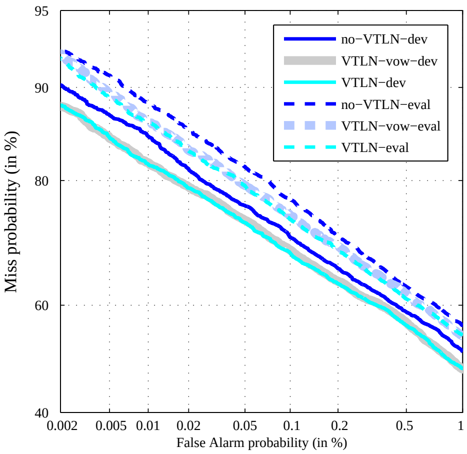


Figure 4.12: Effect of reduced features in VTLN warping factor estimate at iteration index=5 on DET curve.

features to train GMM and estimate VTLN warping factor. The performance in MTWV for different feature sets is shown in Table 4.4. It can be seen from the Table 4.4 that proposed GMM-based framework improves the QbE-STD performance with pooled data as well.

4.2.5.6 Deterministic Annealing Expectation Maximization (DAEM)

EM algorithm iteratively estimates the Maximum Likelihood (ML) of model parameters in the presence of incomplete or hidden data. Though EM algorithm has several issues, it suffers from local optimal value. To address this, problem

Table 4.4: Performance of SWS 2013 QbE-STD with pooled data from QUESST 2014 (in MTWV). The numbers in the brackets indicate ATWV for Eval set.

	Dev Set			Eval Set		
VTLN	MFCC	PLP	MFCC-TMP	MFCC	PLP	MFCC-TMP
×	0.188	0.198	0.181	0.143 (0.137)	0.159 (0.157)	0.144 (0.141)
✓	**0.203**	**0.231**	**0.207**	0.154 (**0.153**)	0.173 (**0.172**)	0.150 (**0.146**)

(× = No VTLN, ✓ = VTLN)

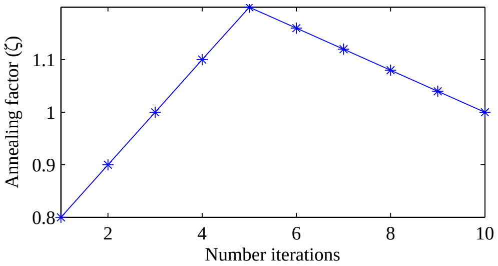


Figure 4.13: Values of annealing factor (ζ) at every iterations.

Deterministic Annealing EM algorithm was proposed in 1998 [210]. DAEM algorithm uses the principle of maximum entropy and statistical mechanism analogy. DAEM is an alternative to Expectation Maximization problem, where maximization of likelihood problem is posed as minimizing free energy [210–212]. This results into modified posterior probability that takes into account annealing factor ζ that is inversely proportional to the temperature.

The parameters of GMMs in EM framework, i.e., $\theta := \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ can be estimated using EM algorithm. The class assignments for each observation vector $\mathbf{o}_t$ can be made based on the responsibility terms, which is given by [212]:

$$\gamma_t^k = E_{\theta_0}[Z_t^k] = \frac{\tilde{\pi}_k \mathcal{N}(\mathbf{o}_t; \tilde{\mu}_k \tilde{\Sigma}_k)}{\sum_{k=1}^{K} \tilde{\pi}_k \mathcal{N}(\mathbf{o}_t; \tilde{\mu}_k \tilde{\Sigma}_k)}, \tag{4.7}$$

where $\theta_0 := \{\tilde{\pi}_k, \tilde{\mu}_k, \tilde{\Sigma}_k\}_{k=1}^{K}$ is old parameter values. For DAEM, eq. (4.7) is modified by annealing parameter ζ as [212]:

$$\gamma_t^k = E_{\theta_0}[Z_t^k] = \frac{(\tilde{\pi}_k \mathcal{N}(\mathbf{o}_t; \tilde{\mu}_k \tilde{\Sigma}_k))^{\zeta}}{\sum_{k=1}^{K} (\tilde{\pi}_k \mathcal{N}(\mathbf{o}_t; \tilde{\mu}_k \tilde{\Sigma}_k))^{\zeta}}. \tag{4.8}$$

The values of ζ changes as shown in Figure 4.13. At equilibrium, a thermodynamic system approaches to the state that has minimum free energy. Similarly, as number of iteration increases, the parameters of GMM attains maximum likelihood. The annealing factor ζ (in eq. (4.8)) that is analogues to the temperature in thermodynamics and annealing factor is inversely related to the temperature.

Here, we perform anti-annealing and annealing in DAEM algorithm. We experimented with DAEM-based parameter estimation approach for SWS 2013 database (as discussed in sub-Section 4.2.5.6). SWS 2013 QbE-STD task, the performance of EM and DAEM is shown in Table 4.5. It can be observed that VTL-warped Gaussian posteriorgram gave better performance than the Gaussian posteriorgram. In the next Section, we present novel mixture of GMMs approach for QbE-STD task.

Table 4.5: Performance of DAEM on SWS 2013 QbE-STD (in MTWV). The numbers in the brackets indicate ATWV. After [1].

VTLN	Dev Set				Eval Set			
	EM		DAEM		EM		DAEM	
	MFCC	PLP	MFCC	PLP	MFCC	PLP	MFCC	PLP
×	0.188	0.195	0.188	0.200	0.138 (0.137)	0.145 (0.145)	0.139 (0.137)	0.146 (0.145)
✓	0.209	**0.222**	0.211	**0.222**	0.159 (0.154)	0.169 (**0.168**)	0.159 (0.158)	0.160 (**0.159**)

4.3 Mixture of GMMs

In this Section, we introduce a modification in Gaussian posteriorgram by imposing the constraints from broad phoneme recognition system. In particular, we used broad phoneme classes (such as, vowels, semi-vowels, fricatives, nasals, plosives) to provide constraints in Gaussian Mixture Model (GMM) clustering. The earlier studies used prior constraints from labeled data to provide better initialization during GMM training [101]. In addition, the mixture of Auto-associative Neural Network (AANNs) was introduced to improve the performance obtained by using single AANN for speaker recognition task [213]. The mixtures are tied using broad phoneme class probabilities derived from the Multilayer Perceptron (MLP). With these two motivations, in this thesis, we present a novel mixture of GMMs for QbE-STD task. The GMM is trained with complete speech data, where no phonetic constraints are imposed during training. GMM parameters can be controlled by using prior information supplied by phonetic inferences. The novelty of proposed approach lies in prior probability assignment as weights of mixture of GMMs. In the next sub-Section, the mathematical formulation of proposed approach is presented.

4.3.1 Mixture of GMM Posteriorgram

The speech data governs acoustically similar broad phonetic structures. The mixture of GMMs, comprises of a group of GMMs, where each group corre-

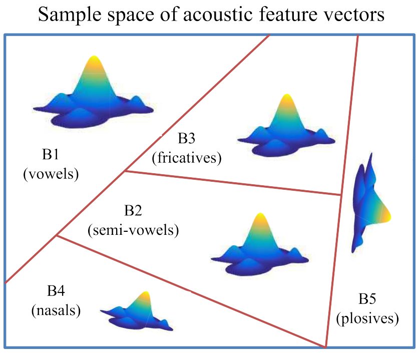


Figure 4.14: An illustration of mixture of GMMs as set of broad phoneme classes (that are mutually exclusive and exhaustive events) each representing a GMM. After [214].

sponds to broad phoneme categories. Consider a set of K broad classes, namely, $B := \{B_1, B_2, \cdots, B_K\}$ and number of Gaussians in each k^{th} broad class is M_k. As illustrated in Figure 4.14, acoustic feature vectors (except for silence regions) are classified into five broad phoneme classes. All five broad classes of a probabilistic sample space are $B_1, B_2, \cdots, B_5$ are assumed to be mutually exclusive and exhaustive events.

The probability of data under mixture of GMMs is given by [215]:

$$P(\mathbf{o}_t|\theta) = \sum_{k=1}^{K} P(B_k|\mathbf{o}_t) \left(\sum_{i=1}^{M_k} \pi_i^k \mathcal{N}(\mathbf{o}_t; \mu_i^k, \Sigma_i^k) \right), \tag{4.9}$$

where parameters, $\theta = \{\pi_i^k, \mu_i^k, \Sigma_i^k\}_{k=1}^{K}{}_{i=1}^{M_k}$ containing set of M_k GMM parameters for each k^{th} broad phoneme classes. Each broad class consists of M_k Gaussian components. The log-likelihood of observation feature sequence, $\mathbf{o} := \{\mathbf{o}_t\}_{t=1}^{T}$ having T length, can be expressed by considering observations from independent identical distribution (*i.i.d.*) and taking logarithm on both sides of eq. (4.9), we get,

$$\begin{aligned} L_\theta(\mathbf{o}) &= \sum_{t=1}^{T} \log P(\mathbf{o}_t|\theta), & (4.10) \\ &= \sum_{t=1}^{T} \log \left(\sum_{k=1}^{K} P(B_k|\mathbf{o}_t) \sum_{i=1}^{M_k} \pi_i^k \mathcal{N}(\mathbf{o}_t; \mu_i^k, \Sigma_i^k) \right). & (4.11) \end{aligned}$$

Now, applying Jensen's inequality [216] for *logarithm* function (which is a *concave* function), we have,
$\log(\lambda x_1 + (1-\lambda)x_2) \geq \lambda \log x_1 + (1-\lambda)\log x_2$.

$$\therefore \log P(\mathbf{o}|\theta) \geq \sum_{t=1}^{T} \sum_{k=1}^{K} P(B_k|\mathbf{o}_t) \log \left(\sum_{i=1}^{M_k} \pi_i^k \mathcal{N}(\mathbf{o}_t; \mu_i^k, \Sigma_i^k) \right).$$

Here, the role of multipliers, i.e., λ and $1-\lambda$, is played by the $P(B_k|\mathbf{o}_t)$ as $\sum_{k=1}^{K} P(B_k|\mathbf{o}_t) = 1$. Hence, we can pull them outside the logarithm and thus the inequality comes into the picture. The parameters of mixture of GMMs, i.e., $\theta := \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$ can be estimated using Expectation-Maximization (EM) algorithm. In order to invoke EM algorithm, a *latent variable* Z_{ti}^{k} is introduced, which represents the assignment of observation vector, $\mathbf{o}_t$, being in i^{th} Gaussian component of k^{th} broad phoneme class. The latent variable Z_{ti}^{k} governs the Bernoulli distribution. Hence, the objective function in **E-step** for initial model, $\theta_0 = \{\tilde{\pi}_i^k, \tilde{\mu}_i^k, \tilde{\Sigma}_i^k\}_{k=1}^{K}{}_{i=1}^{M_k}$ is given by [215]:

$$Q(\theta|\theta_0) = \sum_{t=1}^{T}\sum_{k=1}^{K} P(B_k|\mathbf{o}_t) \sum_{i=1}^{M_k} E_{\theta_0}[Z_{ti}^{k}] \log\left(\pi_i^k \mathcal{N}(\mathbf{o}_t; \mu_i^k, \Sigma_i^k)\right).$$

The class assignments for each observation vector, $\mathbf{o}_t$, can be made based on the responsibility term, which is given by [215]:

$$\gamma_{ti}^{k} = E_{\theta_0}[Z_{ti}^{k}] = \frac{\tilde{\pi}_i^k \mathcal{N}(\mathbf{o}_t; \tilde{\mu}_i^k \tilde{\Sigma}_i^k)}{\sum_{j=1}^{M_k} \tilde{\pi}_j^k \mathcal{N}(\mathbf{o}_t; \tilde{\mu}_j^k \tilde{\Sigma}_j^k)}. \tag{4.12}$$

During maximization, **M-step**, parameters are updated as follows [215]:

$$\pi_i^k = \frac{\sum_{t=1}^{T} P(B_k|\mathbf{o}_t)\gamma_{ti}^{k}}{\sum_{t=1}^{T} P(B_k|\mathbf{o}_t)}, \tag{4.13}$$

$$\mu_i^k = \frac{\sum_{t=1}^{T} \mathbf{o}_t P(B_k|\mathbf{o}_t)\gamma_{ti}^{k}}{\sum_{t=1}^{T} P(B_k|\mathbf{o}_t)\gamma_{ti}^{k}}, \tag{4.14}$$

$$\Sigma_i^k = \frac{\sum_{t=1}^{T} (\mathbf{o}_t \mathbf{o}_t^T) P(B_k|\mathbf{o}_t)\gamma_{ti}^{k}}{\sum_{t=1}^{T} P(B_k|\mathbf{o}_t)\gamma_{ti}^{k}} - \mu_i^k \mu_i^{kT}. \tag{4.15}$$

More details of the mixture of GMMs are given in [215].

4.3.2 Practical Implementation

4.3.2.1 Broad Phoneme Posterior Probability

To compute the posterior probability associated with broad phonetic classes, $P(B_k|\mathbf{o}_t)$, we have used MLP posterior values. In particular, we have used open source *BUT phoneme recognizer* trained for multiple languages, namely, Czech (CZ), Hungarian (HU) and Russian (RU) [61]. Later on, we combined the total probabilities *w.r.t.* each phoneme associated with the same broad phoneme classes by summing them up for each recognizer. Thereafter, we normalized each broad phoneme class posterior values by 3, which is a number of phoneme recognizer. We treated affricate and plosive as a common broad phoneme category as a plosive broad phoneme class. Another alternatives could be use of ergodic HMM for broad phoneme computation that requires labels. We can also generate initial labels from foreign recognizer such as, BUT phoneme recognizer and train ergodic HMM.

4.3.2.2 Relative Significance of Each Broad Class

In order to evaluate the discrimination capability of relative significance different broad phoneme classes followed by possible assignments of different number of Gaussians, we conducted the following experiments. The spoken queries in Czech language in Dev set of SWS2013 are taken and the discrimination between two different queries having different contents are time-aligned using DTW. Then, the distance corresponding to each broad class is pooled. For instance, if two spoken queries $\mathcal{X}^i$ and $\mathcal{Y}^i$ are the i^{th} pair of queries. DTW between them gives time-aligned queries $\mathcal{X}_a^i$ and $\mathcal{Y}_a^i$ having length L_i. The claim is to investigate the relative importance of the broad phoneme class that maximizes the discrimination by more distance value. In this context, we propose the discrimination capability of each broad class, $d_B(k)$, associated in each broad phoneme class as:

$$d_B(k) = \frac{1}{IL} \sum_{i=1}^{I} \sum_{l=1}^{L_i} P(B_k | \mathcal{X}_a^i(l)) d_a^i(l), \tag{4.16}$$

where $d_a^i(l) = dist(\mathcal{X}_a^i(l), \mathcal{Y}_a^i(l))$ is the distance between feature vectors and k corresponds to five broad phoneme class ($1 \leq k \leq 5$). Here, $L = \sum_{i=1}^{I} L_i$ is a total number of aligned frames, and I is the total number of frame pairs. The values of $d_B(k)$ obtained using MFCC and PLP Gaussian posteriorgram features, i.e., MFCC-GP and PLP-GP, are shown in Table 4.6. The higher the distance corresponds to better discrimination and as shown in Table 4.6, broad vowel class gives higher discrimination capability across different word pairs. This higher discrimination for broad vowel class motivated authors for using more number of Gaussians in vowel broad class. In addition, the most of the phonemes are belonging from vowel category. Hence, we set more GMMs to vowel category than other broad phoneme categories. In addition, it was studied that the vowel sounds can be much compressed than the consonant preserving the same intelligibility [190]. In this thesis, we considered two cases, where total number of Gaussian components are *64* (= 32, 8, 8, 8, and 8) and *128* (= 64, 16, 16, 16, and 16) as assignments for broad phoneme classes, namely, vowel, plosive, semivowel, nasal and fricative, respectively.

Table 4.6: Contribution in each broad phoneme class (i.e., $d_B(k)$) for discriminating different spoken words. After [214]

Broad Class Features	Vowel	Plosive	Semivowel	Nasal	Fricative
MFCC-GP	**7.26**	4.19	2.25	2.52	1.17
PLP-GP	**7.62**	4.39	2.36	2.62	1.23

4.3.2.3 Training Procedure and Posteriorgram Computation

The initialization of mixture GMMs is done with the $P(B_k|\mathbf{o}_t)$. Initially, all the features are split into $K = 5$ broad classes based on the

$$\mathbf{o}_t^k := \{\mathbf{o}_t | \underset{1 \leq j \leq K}{\operatorname{argmax}} P(B_j|\mathbf{o}_t) = k\}. \tag{4.17}$$

Now to capture the spread within each broad class, Vector Quantization (VQ) is performed with Linde-Buzo-Gray (LBG) algorithm with splitting parameter, $\epsilon = 0.2$ [184]. More detail about VQ algorithm used in the thesis is discussed in Figure D.1 of Appendix D. After initial model parameter estimation, the parameters of mixture of GMMs are estimated with 10 iterations. The convergence in terms of log-likelihood, $L_\theta(\mathbf{o})$, is shown for MFCC and PLP feature sets in Figure 4.15. It can be observed that as number of iterations increases, likelihood, $L_\theta(\mathbf{o})$, converges.

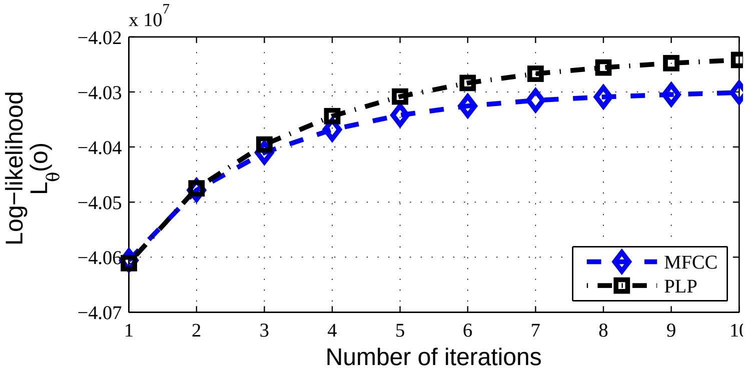
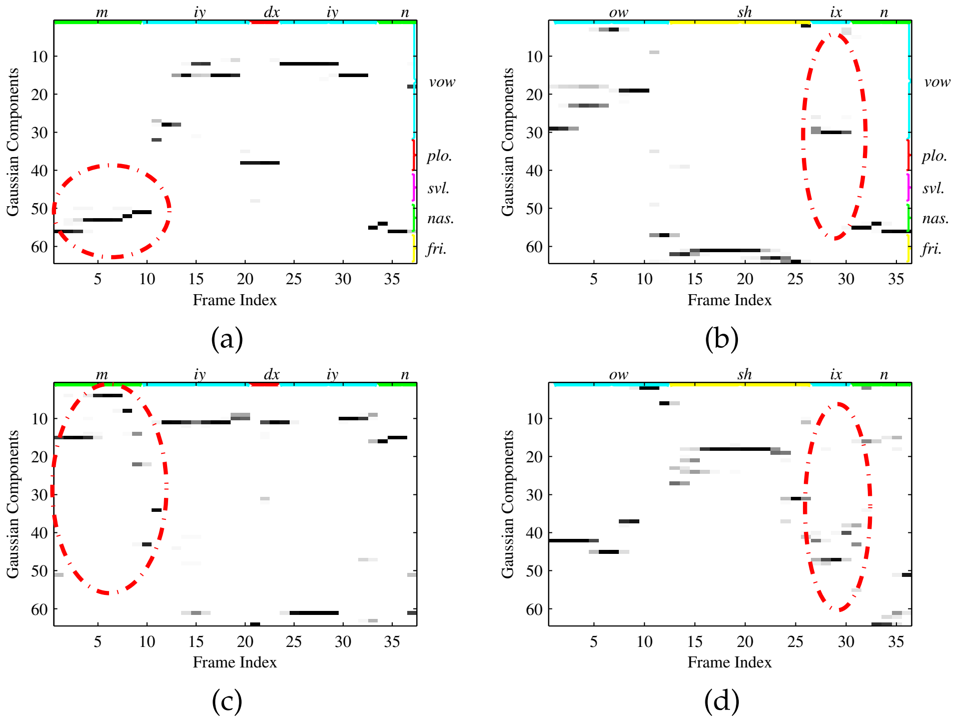


Figure 4.15: Plot of log-likelihood (i.e., $L_\theta(\mathbf{o})$) *w.r.t.* number of iterations. After [214].

Now, for computing the posterior probability under the mixture of GMMs framework, consider G_i^k be the i^{th} Gaussian component in the mixture of GMMs. We represent G_i^k as a joint event of k^{th} broad class, B_k, and C_k^i as the i^{th} Gaussian component in k^{th} broad class. Both of these events are conditionally-independent.



(a) (b) (c) (d)

Figure 4.16: Comparison between Gaussian posteriorgram and mixture of GMM posteriorgram for two different words taken from TIMIT database, namely, *meeting* and *ocean*. Mixture of GMMs posteriorgram for (a) *meeting*, and (b) *ocean*. Gaussian posteriorgram for (c) *meeting*, and (d) *ocean*. The red circles in (b) and (d) indicates better representation via mixture of GMMs than its GMM counterpart. After [214].

Hence, the posterior probability is given by:

$$P(G_i^k|\mathbf{o}_t) = P(B_k C_i^k|\mathbf{o}_t),$$

$$P(G_i^k|\mathbf{o}_t) = \frac{P(B_k|\mathbf{o}_t)P(C_i^k)P(\mathbf{o}_t|C_i^k)}{\sum_{k=1}^{K}\sum_{j=1}^{M_k} P(B_k|\mathbf{o}_t)P(C_j^k)P(\mathbf{o}_t|C_j^k)}, \tag{4.18}$$

$$P(G_i^k|\mathbf{o}_t) = \frac{P(B_k|\mathbf{o}_t)\pi_k^i\mathcal{N}(\mathbf{o}_t;\mu_k^i,\Sigma_k^i)}{\sum_{k=1}^{K}\sum_{j=1}^{M_k} P(B_k|\mathbf{o}_t)\pi_k^j\mathcal{N}(\mathbf{o}_t;\mu_k^j,\Sigma_k^j)}. \tag{4.19}$$

An algorithm for training of mixture of GMMs is shown in Algorithm 7. Figure 4.16 shows the posteriorgram of the mixture of GMMs for the word /*meeting*/ and /*ocean*/ and corresponding Gaussian posteriorgrams (GP). AS shown in Figure 4.16 (a) and (b), posterior probabilities in GP is scattered along multiple clusters. This indicates that the mapping between phonetic classes and Gaussian components is one-to-many (which is also discussed in [97]). It can be observed that the highest probability of posterior falls onto the broad phoneme categories.

Algorithm 7 Proposed algorithm for training of mixture of GMMs

```
Input: Feature vector, $\mathbf{o}_t$, and broad phoneme posterior probability $P(B_k|o_t)$, $1 \leq k \leq K$, and $M_k$ number of Gaussian components in each $k^{th}$ broad phoneme class.
Output: The parameters of mixture of GMMs, i.e., $\theta := \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{K}$.
    Initialization :
    for $k = 1$ to $K$ do
        Let $\mathbf{o}_t^k := \{\mathbf{o}_t | \underset{1 \leq j \leq K}{\text{argmax}}\, P(B_j|\mathbf{o}_t) = k\}$.
        Perform vector quantization (VQ) on feature vectors, $\mathbf{o}_t^k$ and let the codebook centroids be $c_i^k$, $1 \leq i \leq M_k$
        Initial mean: $\tilde{\mu}_i^k := c_i^k$
        Codebook data assignment:
        $\mathbf{o}_{ti}^k = \{\mathbf{o}_t^k | \underset{1 \leq\ l \leq M_k}{\text{argmin}}\, d_{Euc}(\mathbf{o}_t^k, c_l^k) = i\}$
                                # Set of the closest feature vectors to the centroid
        Initial weights: $\tilde{\pi}_i^k := \frac{|\mathbf{o}_{ti}^k|}{|\mathbf{o}_t^k|}$
                                # Ratio of number of features, $\mathbf{o}_{ti}^k$ to number of features, $\mathbf{o}_t^k$
        Initial covariance: $\tilde{\Sigma}_i^k := \mathbb{E}\left[(\mathbf{o}_{ti}^k - c_k^i)(\mathbf{o}_{ti}^k - c_k^i)^T\right]$
    end for
    Run EM algorithm for $N_{iter}$ times
    for $n_{iter} = 1$ to $N_{iter}$ do
        Expectation (E-Step)
        $\gamma_{ti}^k = \frac{\tilde{\pi}_i^k \mathcal{N}(\mathbf{o}_t; \tilde{\mu}_i^k \tilde{\Sigma}_i^k)}{\sum_{j=1}^{M_k} \tilde{\pi}_j^k \mathcal{N}(\mathbf{o}_t; \tilde{\mu}_j^k \tilde{\Sigma}_j^k)}$.
        Maximization (M-Step)
        Updated weight: $\pi_i^k = \frac{\sum_{t=1}^{T} P(B_k|\mathbf{o}_t)\gamma_{ti}^k}{\sum_{t=1}^{T} P(B_k|\mathbf{o}_t)}$,
        Updated mean: $\mu_i^k = \frac{\sum_{t=1}^{T} \mathbf{o}_t P(B_k|\mathbf{o}_t)\gamma_{ti}^k}{\sum_{t=1}^{T} P(B_k|\mathbf{o}_t)\gamma_{ti}^k}$,
        Updated covariance: $\Sigma_i^k = \frac{\sum_{t=1}^{T} (\mathbf{o}_t \mathbf{o}_t^T) P(B_k|\mathbf{o}_t)\gamma_{ti}^k}{\sum_{t=1}^{T} P(B_k|\mathbf{o}_t)\gamma_{ti}^k} - \mu_i^k {\mu_i^k}^T$.
        Substitution:
        $\tilde{\pi}_i^k = \pi_i^k$ , $\tilde{\mu}_i^k = \mu_i^k$ and $\tilde{\Sigma}_i^k = \Sigma_i^k$.
    end for
```

For example, in Figure 4.16 (c) highest posterior values around phoneme $/m/$ and $/n/$ fall on nasal broad phoneme category.

Table 4.7: Performance of mixture of GMMs for SWS 2013 QbE-STD. The numbers in the brackets indicate ATWV for Eval set. After [214]

Feature Sets	NG	Query Type	GMM		Mixture of GMMs	
			Dev	Eval	Dev	Eval
MFCC	64	Normal	0.156	0.111 (0.111)	0.217	0.180 (0.180)
	128	Normal	0.188	0.138 (0.137)	0.240	0.195 (0.194)
	128	Extended	0.218	0.161 (0.161)	**0.281**	0.227 (**0.224**)
PLP	64	Normal	0.158	0.126 (0.123)	0.228	0.196 (0.192)
	128	Normal	0.195	0.145 (0.145)	0.246	0.208 (0.203)
	128	Extended	0.221	0.169 (0.163)	**0.283**	0.238 (**0.233**)

NG=Number of Gaussians.

Table 4.8: Performance of score-level fusion with mixture of GMMS for SWS 2013 QbE-STD. The numbers in the brackets indicate ATWV for Eval set. After [214]

Feature Sets	Normal Query		Extended Query	
	Dev	Eval	Dev	Eval
CZ	0.375	0.342 (0.340)	0.401	0.371 (0.367)
CZ + MFCCGP	0.427	0.378 (0.375)	0.463	0.418 (0.400)
CZ + MFCCmixGP	0.442	0.398 (0.396)	0.479	0.441 (0.423)
CZ + PLPGP	0.424	0.381 (0.377)	0.456	0.422 (0.403)
CZ + PLPmixGP	0.444	0.397 (0.394)	0.479	0.435 (0.420)
MFCCGP + PLPGP + MFCCmixGP + PLPmixGP	0.276	0.226 (0.211)	0.315	0.265 (0.254)
CZ + MFCCGP + PLPGP + MFCCmixGP + PLPmixGP	**0.455**	0.412 (**0.400**)	**0.494**	0.453 (**0.449**)

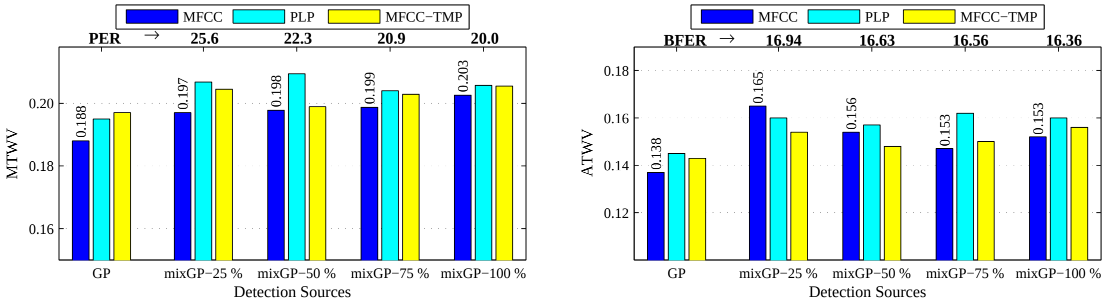
+ indicates the score-level fusion

4.3.3 Experimental Results

The search performance obtained on SWS 2013 database is reported in terms of MTWV and shown in Table 4.7. The MTWV for SWS 2013 dataset is improved by 0.057 and 0.063 w.r.t. Gaussian posteriorgram for MFCC and PLP, respectively.

4.3.3.1 Score-level Fusion

The details for score-level fusion for different QbE-STD systems were discussed in sub-Section 3.5.1. Here, all the four search systems, namely, Gaussian posteriorgrams and a mixture of Gaussian posteriorgram from MFCC and PLP, participate into the fusion. The performance in terms of MTWV is reported in Table 4.8. It can be observed that score-level fusion further improves the performance of QbE-STD system. To further improve the query detection performance, phonetic posteriorgram is used. We have used BUT phoneme recognizer's CZ phoneme decoder. After score-level fusion for phonetic posteriorgram, MTWV of Eval query



(a) (b)

Figure 4.17: Effect of amount of labeled data used for broad phoneme posterior probability computation on SWS2013 QbE-STD for (a) Dev set , and (b) Eval set. The numbers with percentage indicate amount of training data used in DNN. The numbers on the top of plots indicate the Phone Error Rate (PER) and Broad class Frame Error Rate (BFER). The numbers on the bar plots indicate MTWV.

set is comparable with several SWS 2013 benchmark systems [31] (In particular, as shown in Table 2.3, GTTS: 0.399, L2F: 0.342, CUHK: 0.306, BUT: 0.297, Proposed (CZ + MFCCmixGMM): *0.398*, Proposed (CZ + PLPmixGMM): *0.397*), where '+' indicates score-level fusion.

4.3.3.2 Effect of Amount of Labeled Data

In mixture of GMMs, we use the phoneme posterior probabilities to derive broad phoneme posterior probabilities, i.e., $P(B_k|\mathbf{o}_t)$. In order to investigate the importance this supervised posterior probability, we vary the amount of labeled data for posterior probability computation. We use TIMIT training database without $|\mathtt{sa}|$ sentences and we took different size of training data (i.e., 25 %, 50 %, 75 %, and 100 %). A deep neural network (DNN)-based phoneme recognizer was trained using the KALDI toolkit following Karel's DNN training implementation [217]. DNN contains 3 hidden layers with 1024 hidden units. These broad phoneme posterior probabilities are used to produce a mixture of GMMs. It can be observed from Figure 4.17 that the amount of training data does not change the performance significantly. The numbers on the top of plots show Phone Error Rate measured with TIMIT test set (excluding $|sa|$ sentences). It can be observed from Figure 4.17 that the PER decreases as amount of labeled data increases. However, Broad classwise Frame Error Rate (BFER) is not decreasing much. This is because by considering broad class, the classification error is reduced as the different phonemes of same broad class is treated as the same class. MTWV does not vary significantly with the different size of labeled data used in DNN training. However, MTWV is better than the Gaussian posteriorgram alone. MTWV obtained with TIMIT is not as comparable as the results presented in Table 4.7. This may be due to the language mismatch and sampling rate conversion. We downsampled TIMIT (16 kHz) data by 2 to match the sampling rate of SWS 2013 (8 kHz). However, the performance in MTWV is better for different amount of labeled data. This indicates only 25 % of TIMIT data can be used to train DNN and broad phoneme posterior gave better performance than the Gaussian posteriorgram alone. This observation is same for both Dev and Eval set.

4.4 Chapter Summary

This chapter presented novel frame-level acoustic representation, namely, VTL-warped Gaussian posteriorgram and mixGP for QbE-STD task. GMM-based approach VTLN warping factor reduces the speaker variability and does not require transcription to estimate VTLN warping factor α. We experimented QbE-STD with different evaluation conditions, such as, local constraints, number of Gaussian, EM *vs.* DAEM, etc. The experimental results show that iterative approach gave better performance at 3 or 4 iterations, and hence, further iteration may lead to overfitting. In the second part of this chapter, we discussed novel mixture of GMM for posteriorgram representation. The mixture of GMM utilizes broad phoneme constraints, which makes it better than the state-of-the-art GP. The experimental results shows that mixture of GMM gave better performance than the GP in all the experiments conducted in this chapter and thus, the proposed mixture of GMM posteriorgram shows promising results over the GP posteriorgram. In the next chapter, we will discuss the matching perspective for the design of QbE-STD system.

Chapter 5

Audio Matching Perspective

5.1 Introduction

Chapter 4 discussed several representations for improving performance of QbE-STD system. In this chapter, we will discuss the matching perspective of QbE-STD system. In particular, this chapter will present modified DTW search algorithm to deal with partial matching. The need for partial matching is essential when the instance of query present in test document has variations either at suffix, prefix or word order. The proposed modified DTW search algorithm combines the evidences from various partial matching strategies via different functions (namely, harmonic mean, arithmetic mean and minimum value). MediaEval QUESST 2014 database was used to investigate the effectiveness of partial matching of spoken query. We found that the combined distance pulled using harmonic mean gave better performance for a non-exact match (i.e., *T2* and *T3* queries), whereas it slightly degrades the performance for the exact match (*T1* query) because of the influence of other partial matching distances.

In the next part, we will discuss computational improvement during DTW-based searching using two approaches, namely, feature reduction approach and BoAW approach. DTW-based search requires huge computation as dataset size grows. The speed up of subDTW is necessary for scaling the QbE-STD task to significantly large dataset. In feature reduction approach, consecutive feature vectors within phonetic segment boundaries are merged and DTW is performed on the reduced number of feature vectors. Thus, a lesser number of feature vectors reduces the computational cost by reducing the number of comparison operations. BoAW is two-stage search approach for the QbE-STD task. In the first stage, BoAW models are built by computing the term-frequency (tf) and inverse-document frequency (idf). In posterior feature framework, term corresponds to the phoneme class.

The organization of this chapter is as follows: Section 5.2 discusses the partial matching for non-exact query matching task. Frame-merging approach for search space reduction is discussed in Section 5.3. Proposed segment-level BoAW is discussed in Section 5.4.

5.2 Partial Matching for Non-exact Query Matching

The proposed modified approach does not require to run DTW for multiple times for each query and test utterance pair. Following are the various partial matching strategies employed in modified DTW search algorithm [2]. Figure 5.1 demonstrates partial matching cases along with the exact-matching case.

- **Type-1 (T1) exact match:** We do not need any modification to detect *T1* query, since it is an *exact* match. Hence,

$$dist_1(t_x, q_y) = \min_i \left(\frac{S(i,N)}{T(i,N)} \right). \tag{5.1}$$

- **Type-2 (T2) forward partial match:** Here, the query is truncated at the end and we assumed that the truncated duration is no more than *250* ms. Hence,

$$dist_2(t_x, q_y) = \min_{i, N-24 \leq j \leq N} \left(\frac{S(i,j)}{T(i,j)} \right). \tag{5.2}$$

  Here, *250* ms corresponds to *25* frames (as discussed in sub-Section 3.3.1, the frame rate is *100* frames per second). The warping path of this partial matching aligns to the initial warping path of exact match and hence, this partial matching distance supports the alignment distance obtained using exact match.

- **Type-2 (T2) backward partial match:** The truncation is at the beginning and no more than *250* ms. Since DTW algorithm accumulates distance in the forward direction, this procedure needs to perform the backtracking. Here, our approach consider only single backtracking instead of multiple backtracking. The reason for considering single backtracking is to avoid false detection due to the partial matching. To perform that, we first detect the end frame index, e_{bt}, and corresponding start frame index, s_{bt}, of test utterance, i.e.,

$$e_{bt} = \operatorname*{argmin}_i \left(\frac{S(i,N)}{T(i,N)} \right), \tag{5.3}$$

$$s_{bt} = P(e_{bt}, N). \tag{5.4}$$

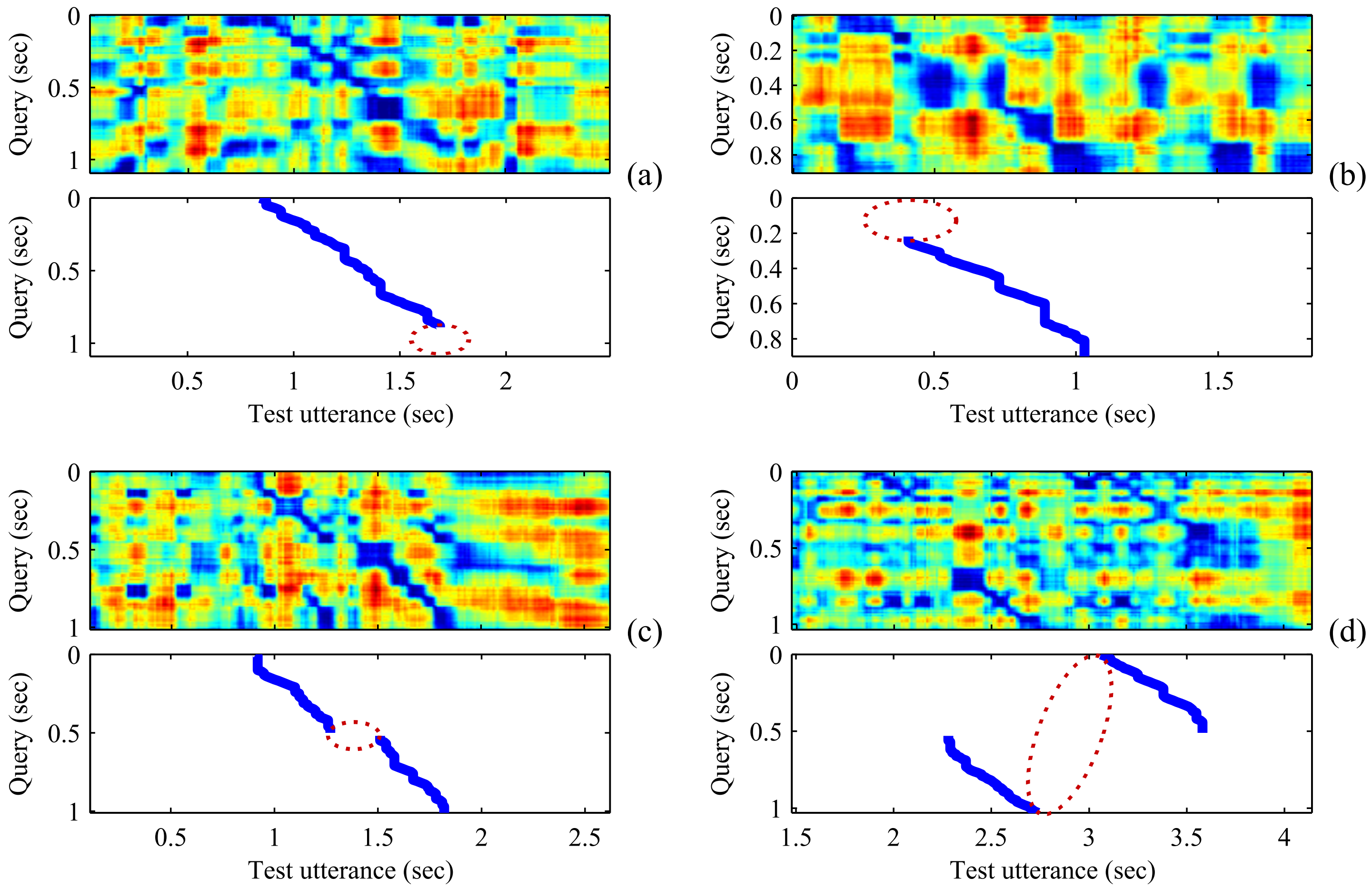


Figure 5.1: Modified DTW for non-exact matching, (a) *T2* query forward partial match case, (b) *T2* query reverse partial match case, (c) *T3* query with filler case, and (d) *T3* query with word reordering case (Local distance matrix on the top panel and optimal DTW warping path on the bottom panel). The dotted circle shows non-matching portion for approximate search. After [2].

After that, we compute backward partial match using the following eqs.,

$$dist_3(t_x, q_y) = \min_{\{(i,j)|1 \le j \le 25, P(i,j)=s_{bt}\}} \left(\frac{S(e_{bt}, N) - S(i,j)}{T(e_{bt}, N) - T(i,j)} \right). \tag{5.5}$$

Here, *250* ms corresponds to *25* frames (as discussed in sub-Section 3.3.1, the frame rate is *100* frames per second). The warping path of this partial matching aligns to the last portion of the warping path of exact match, and hence, this partial matching distance supports the alignment distance obtained using the exact match.

To investigate the effectiveness of proposed partial matching on the performance of *T2* query, we took the harmonic mean scores of $dist_2$ and $dist_3$ for each query. The results for the partial matching against the exact match is presented for *T2* query in Table 5.1. It can be seen from the Table 5.1 that the combination of partial matching $dist_2$ and $dist_3$, i.e., $dist_{23}$, gave better performance for *T2* query than no partial matching case. Hence, this contributes to the overall harmonic distance score d_h.

Table 5.1: Performance of *T2* query using no partial matching and $dist_{23}$ on QUESST 2014 Dev set (in MTWV). After [2].

Feature Vector	MTWV		C_{nxe}^{min}	
	No Partial Matching	$dist_{23}$	No Partial Matching	$dist_{23}$
CZ	0.313	**0.345**	0.697	**0.689**
HU	0.295	**0.304**	0.727	**0.718**
RU	0.313	**0.321**	**0.693**	0.694

- **Type-3 (T3) word re-ordering and filler:** *T3* query considers the cases of word reordering (i.e., jumbling of words) or filler contaminant (i.e., some different word/words in between words of query). To detect such modification of a query, we assume warping path is broken into two parts. To do that, we split the query into an equal number of frames with an assumption that a query contains two words and each word has equal duration. We execute the similar partial matching strategy, as we used for the *T2* forward match and the *T2* backward match. Then, we combine the accumulated distance and normalize with a total number of frames falling on a particular warping path.

  - **Forward half match:** We compute accumulated distance value $S_3^{'}(t_x, q_y) = S(e_{hbt}, \lceil \frac{N}{2} \rceil)$ and path counting value $T_3^{'}(t_x, q_y) = T(e_{hbt}, \lceil \frac{N}{2} \rceil)$ where e_{hbt} is the end frame index corresponds to forward half match and $e_{hbt} =$

$\underset{i}{\operatorname{argmin}}\left(\frac{S(i,\lceil\frac{N}{2}\rceil)}{T(i,\lceil\frac{N}{2}\rceil)}\right)$. $\lceil z \rceil$ is the ceiling function, which is equal to the smallest integer greater than or equal to z.

- **Reverse half match:** To obtain the reverse partial matching, first we compute the start frame index of reverse half matching s_{rhbt} and end frame index of reverse half matching e_{rhbt} as follows:

$$\begin{aligned} e_{rhbt} &= \underset{\{i|P(i,\lceil\frac{N}{2}\rceil)=s_{bt}\}}{\operatorname{argmin}} \left(\frac{S(e_{bt},N)-S(i,\lceil\frac{N}{2}\rceil)}{T(e_{bt},N)-T(i,\lceil\frac{N}{2}\rceil)}\right), \\ s_{rhbt} &= P(e_{rhbt},\lceil\frac{N}{2}\rceil). \end{aligned}$$

Now, reverse half accumulated distance value $S_3^{''}(t_x,q_y)$ and reverse half path counting value $T_3^{''}(t_x,q_y)$ are computed as follows:

$$\begin{aligned} S_3^{''}(t_x,q_y) &= S(e_{bt},N)-S(s_{rhbt},\lceil\frac{N}{2}\rceil), \\ T_3^{''}(t_x,q_y) &= T(e_{bt},N)-T(s_{rhbt},\lceil\frac{N}{2}\rceil). \end{aligned}$$

Now, we compute forward and reverse half match DTW values as $dist_4(t_x,q_y)$, where

$$dist_4(t_x,q_y) = \frac{S_3^{'}(t_x,q_y)+S_3^{''}(t_x,q_y)}{T_3^{'}(t_x,q_y)+T_3^{''}(t_x,q_y)}. \tag{5.6}$$

For *T1* query, the warping paths of the first half query and the second half query overlap to the warping path obtained for the exact match. Thus, the distance obtained through this split query supports the distance obtained by an exact match.

To investigate the effectiveness of $dist_4$ on the performance of *T3* query detection, we compare the performance of no partial matching, i.e., exact matching with $dist_4$ score. The performance of QbE-STD for *T3* query is shown in Table 5.2. It can be seen from the Table 5.2, $dist_4$ gave higher MTWV and lower C_{nxe}^{min}, indicating that it improves the *T3* query detection as compared to the exact matching.

- These partial matching DTW distances, namely, $dist_1(t_x,q_y)$, $dist_2(t_x,q_y)$, $dist_3(t_x,q_y)$ and $dist_4(t_x,q_y)$ are combined using three different functions, namely, harmonic mean $d_h(t_x,q_y)$, minimum value $d_{mi}(t_x,q_y)$ and arithmetic

Table 5.2: Performance of *T3* query using no partial matching and $dist_4$ on QUESST 2014 Dev set (in MTWV). After [2].

Feature Vector	MTWV		C_{nxe}^{min}	
	No Partial Matching	$dist_4$	No Partial Matching	$dist_4$
CZ	0.131	**0.169**	0.742	**0.716**
HU	0.119	**0.129**	0.765	**0.759**
RU	0.144	**0.152**	0.735	**0.710**

mean $d_{mn}(t_x, q_y)$, which are computed as follows [2]:

$$d_h(t_x, q_y) = 4\left(\sum_{k=1}^{4}\frac{1}{dist_k(t_x, q_y)}\right)^{-1}, \tag{5.7}$$

$$d_{mi}(t_x, q_y) = \min_{1\leq k\leq 4}\{dist_k(t_x, q_y)\}, \tag{5.8}$$

$$d_{mn}(t_x, q_y) = \frac{1}{4}\sum_{k=1}^{4} dist_k(t_x, q_y). \tag{5.9}$$

The performance of proposed modified search algorithm for Dev and Eval sets are shown in Figure 5.2 and Figure 5.3, respectively. It can be seen that the overall performance of modified search algorithm is better than the conventional DTW search. However, for *T1* query, the performance of modified search algorithm is slightly poor than the conventional approach. This is because conventional DTW search algorithm is designed for *T1* query, i.e., an *exact match*, whereas modified approach considers all the kinds of non-exact variations in the query. Modified search approach with minimum value selection performs better on *T3* query. This might be because minimum value always selects the best match among different partial matching distances. However, it also detects a partial match to the test utterance, which does not contain the query and hence, this introduces false alarms. The harmonic mean of different partial matching distances is less affected, which avoids the higher valued outliers [218]. Hence, in this work, we prefer harmonic mean over arithmetic mean to combine the partial matching distances.

The truncation point for *T3* query can be anywhere in the middle. We also checked with case of equal phone segment split, where we split the query based on an equal number of phone segments. The results on QUESST 2014 Dev set are shown in Figure 5.4. It can be seen from Figure 5.4, that overall performance is almost similar for most of the query types. In majority of the cases, it can be seen that using equal segment split for $dist_4$ computation, the performance of *T3* query slightly improves. However, it slightly degrades the performance of *T2*

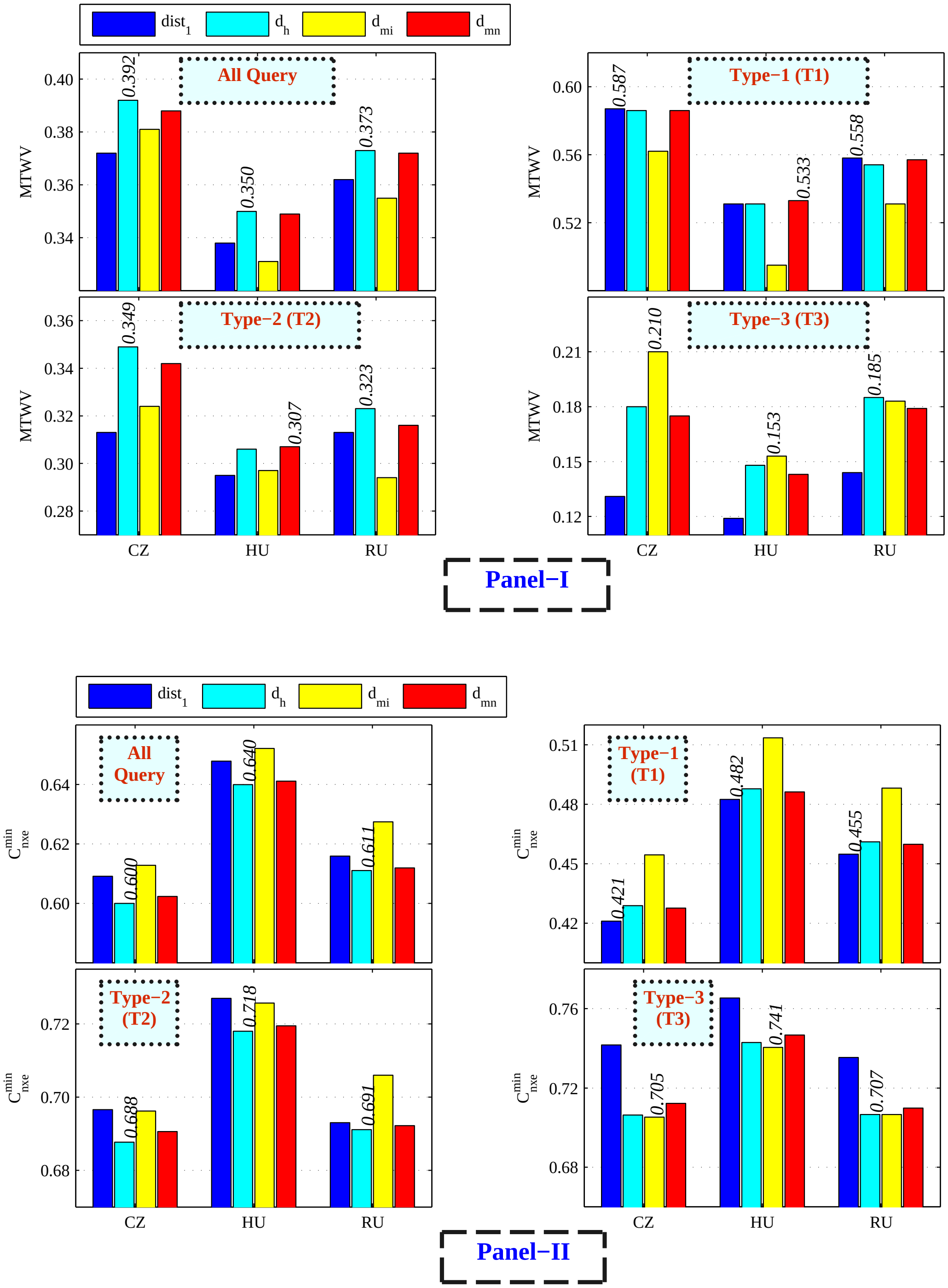


Figure 5.2: Performance of partial matching for QUESST 2014 Dev set. Panel (I) MTWV, and Panel (II) C^{min}_{nxe}. After [2].

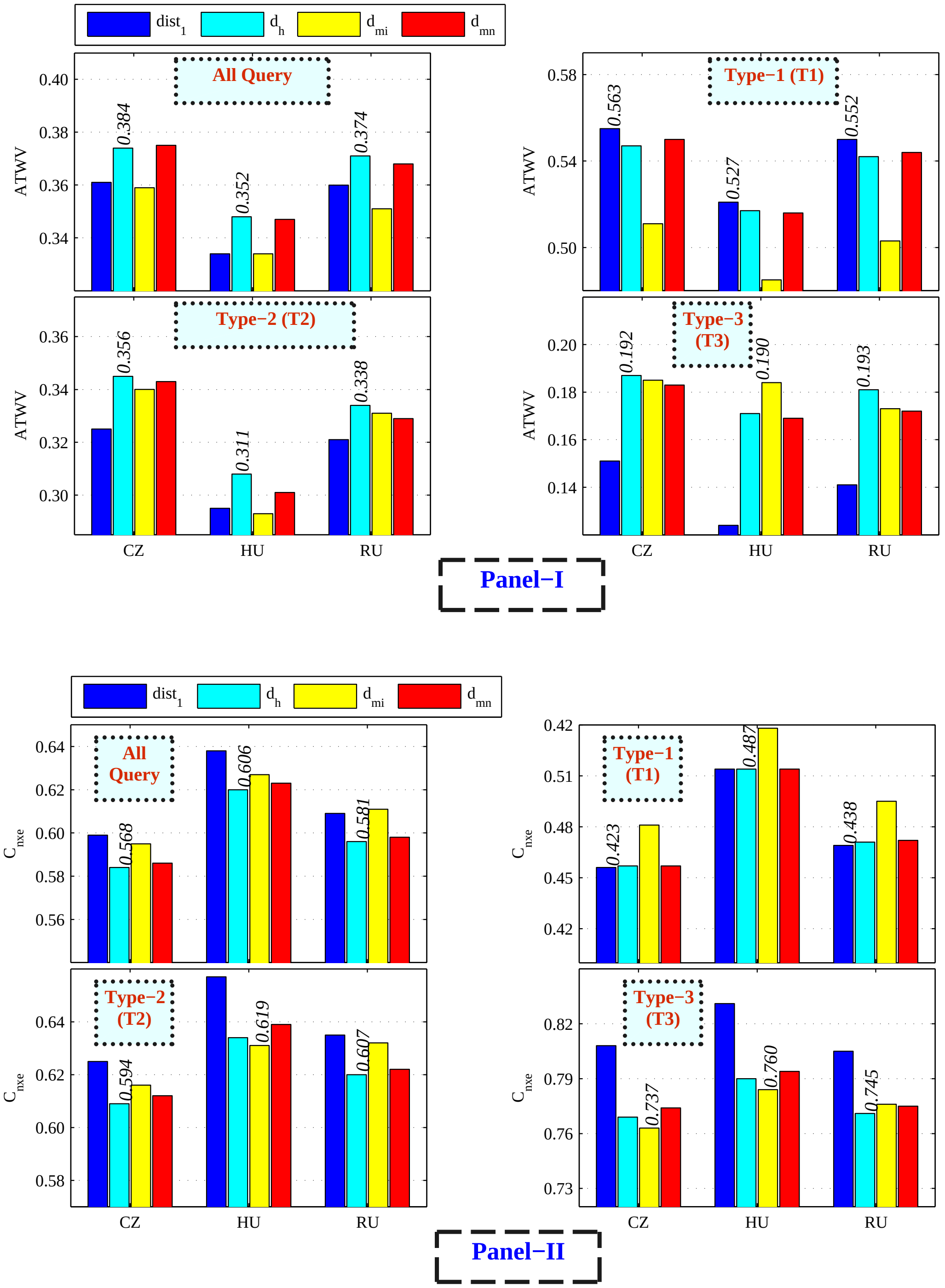


Figure 5.3: Performance of partial matching for QUESST 2014 Eval set. Panel (I) ATWV, and Panel (II) C_{nxe}. The number on the top of bars indicate the optimal values, i.e., MTWV and C_{nxe}^{min}. After [2].

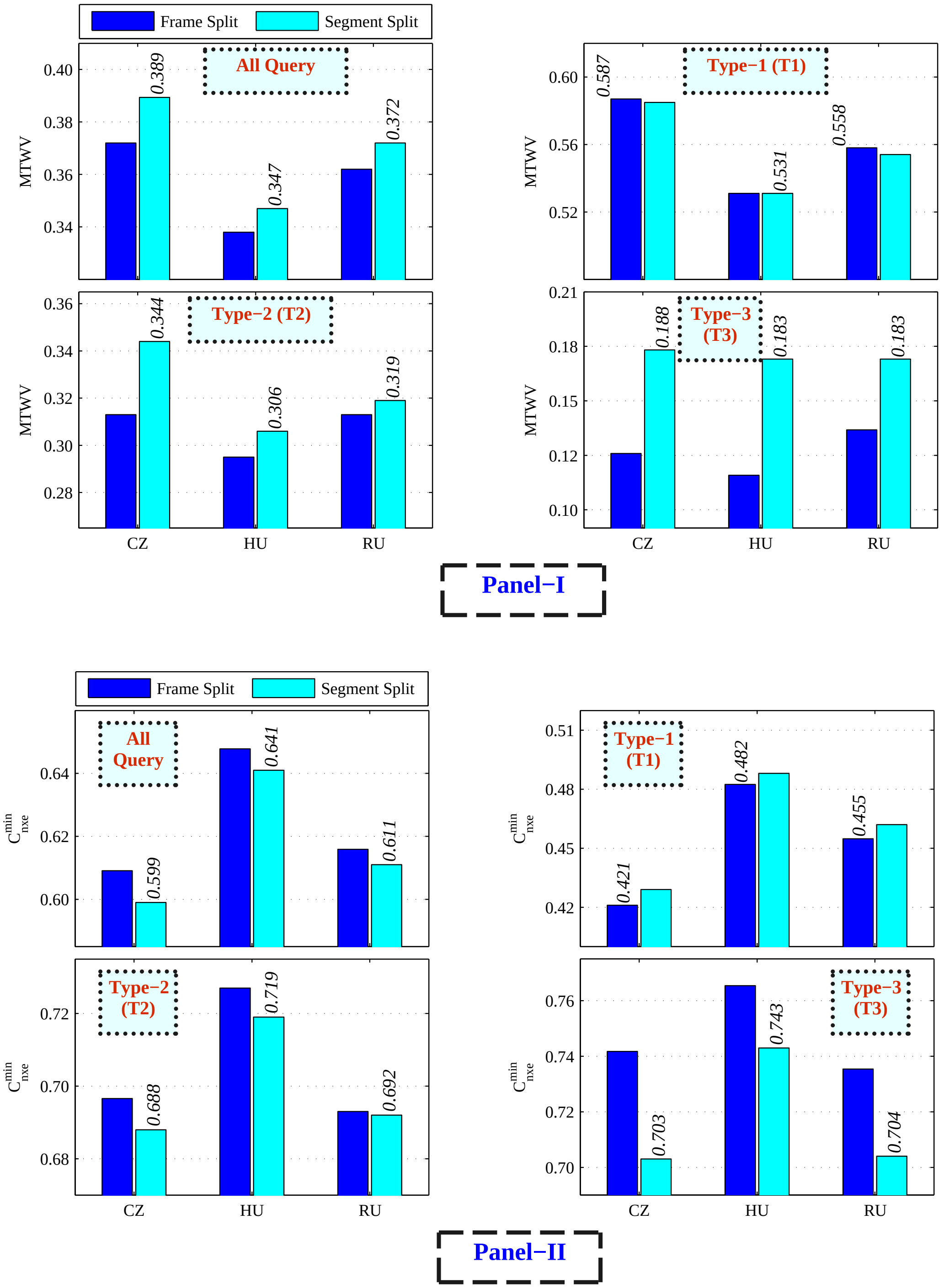


Figure 5.4: Results of QbE-STD for equal frame split and equal phone segment split QUESST 2014 Dev set. Panel (I) MTWV, and Panel (II) C_{nxe}^{min}. After [2].

query. Hence, we will split the query into an equal number of frames for $dist_4$ computation.

In this Section, we investigated the searching algorithm for partial matching and combine the evidences from various partial matching strategies. Next, two Sections present the issue of search space complexity and the approaches to reduce search space, namely, feature reduction and BoAW.

5.3 Feature Reduction Approach

Feature vectors are extracted and then the reduction is performed on feature vectors with different reduction factor β. In this thesis, we used three reduction factor, namely, $\beta = 2, 3, 4$, and we analyzed the search performance and the time required to perform the search. Further reduction in β value was found to improve the speed of search at the cost of degradation in search performance. As suggested in [53], computation time is reduced from $\mathcal{O}(MN)$ to $\mathcal{O}(\frac{MN}{\beta^2})$, where M and N are the number of features in test utterance and query, respectively. For the analysis, we refer the feature reduction suggested in [53] as original feature reduction strategy. The code for sequential DTW is scripted in C language and compiled under MATLAB EXecutable (MEX) environment. All the experiments are conducted on a general purpose CPU having hardware specifications: *64*-bit Intel *i5 @ 2.80* GHz, *4* GB installed RAM. We applied phone segmentation on test utterance and query and then perform feature reduction. To illustrate this, consider Figure 5.5, which shows the feature reduction factor, $\beta = 3$. The phonetic segmentation is performed using well known Spectral Transition Measure (STM) [219], which is discussed in the following sub-Section. The performance of phone segmentation is discussed in [84]. Feature reduction approach reduces the number of frames used in DTW comparison.

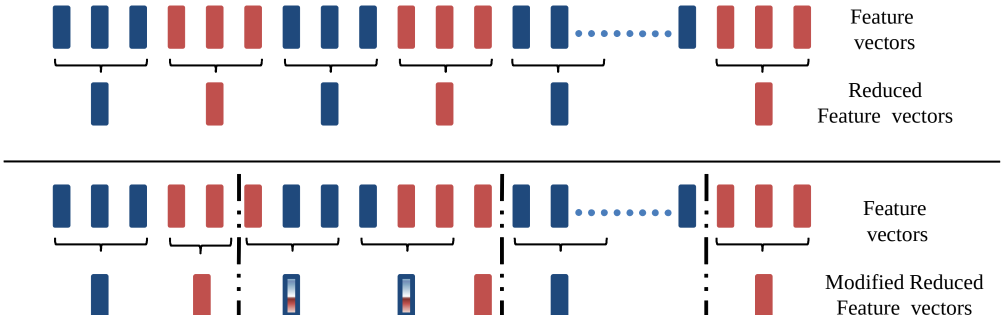


Figure 5.5: An illustrative diagram of feature reduction (reduction factor, $\beta = 3$). Dashed line represents the phonetic boundaries. After [84].

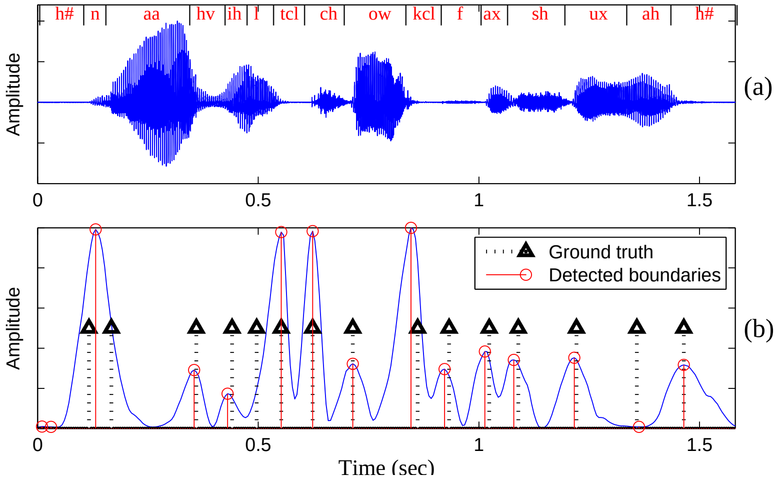


Figure 5.6: Phonetic boundary segmentation using STM: (a) speech signal taken from TIMIT database, '*Now he'll choke for sure*', and (b) STM contour, detected phonetic boundaries (continuous) and manually marked phonetic reference (dotted) boundaries. After [2, 84].

Recently, for spoken word discovery task, uniform and non-uniform downsampling approach was suggested in [220]. In both the approaches, acoustic feature vectors series are segmented into k acoustic regions and the corresponding mean feature vectors are concatenated to form a single feature vector. However, in our present work, consecutive features are averaged out, which does not increase the dimension of the posterior feature vector.

5.3.1 Phone Segmentation

During speech production mechanism, there is a change in the vocal tract apparatus (system) as well as (a vocal) source of a speaker. The excitation source and system characteristics might be relatively steady as well as transitional. In addition, human perception for hearing system responds better to the transitional stimuli than the steady-state stimuli [219, 221]. Generally, the steady characteristics are often observed around the *middle* region of vowels, nasals, fricatives, etc. Transitions occur between the adjacent phonemes due to the transitional movements of the articulators (such as, lips, tongue, jaw muscles, velum, etc.). This might be reflected in the *spectral* and *temporal* structure of the speech signal. In order to capture such variations, spectral transition measure (STM) is used [219, 222]. To estimate STM contour, cepstral information is extracted from the speech signal.

$$STM(t) = \frac{1}{K}\sum_{i=1}^{K} a_{it}, \tag{5.10}$$

where $a_{it} = \frac{\sum_{k=-k_0}^{k_0} kC_i(t+k)}{\sum_{k=-k_0}^{k_0} k^2}$, and $C_i(k)$ is the i^{th} cepstral coefficients. Figure 5.6 shows an utterance taken from TIMIT database and corresponding STM contour. Figure 5.6 also shows the detection obtained as an output of STM algorithm and manually marked phonetic boundaries are almost aligned. Though there are various missing and extra boundaries, which should be incorporated while evaluating the performance of phonetic boundary detection task.

There are several advantages of STM over phoneme decoders for phone segmentation. In particular, STM does not require training rather it uses spectral transitions over the consecutive frames and hence, it is language-independent approach. However, the boundaries detected by the phoneme recognizer (i.e., phoneme decoders) are language-dependent because phoneme recognizers are trained on a particular language. To understand the effectiveness of phoneme segmentation using STM, we conducted the phone segmentation task on TIMIT dataset using STM and BUT phoneme decoders (namely, CZ, HU and RU). The performance of phone segmentation task is evaluated at different agreement interval (AgInt). x % agreement interval for i^{th} segment, i.e., ϵ_i is defined as:

$$\zeta_i - \frac{x}{100}\left(\zeta_i - \zeta_{i-1}\right) \leq \epsilon_i \leq \zeta_i + \frac{x}{100}\left(\zeta_{i+1} - \zeta_i\right), \tag{5.11}$$

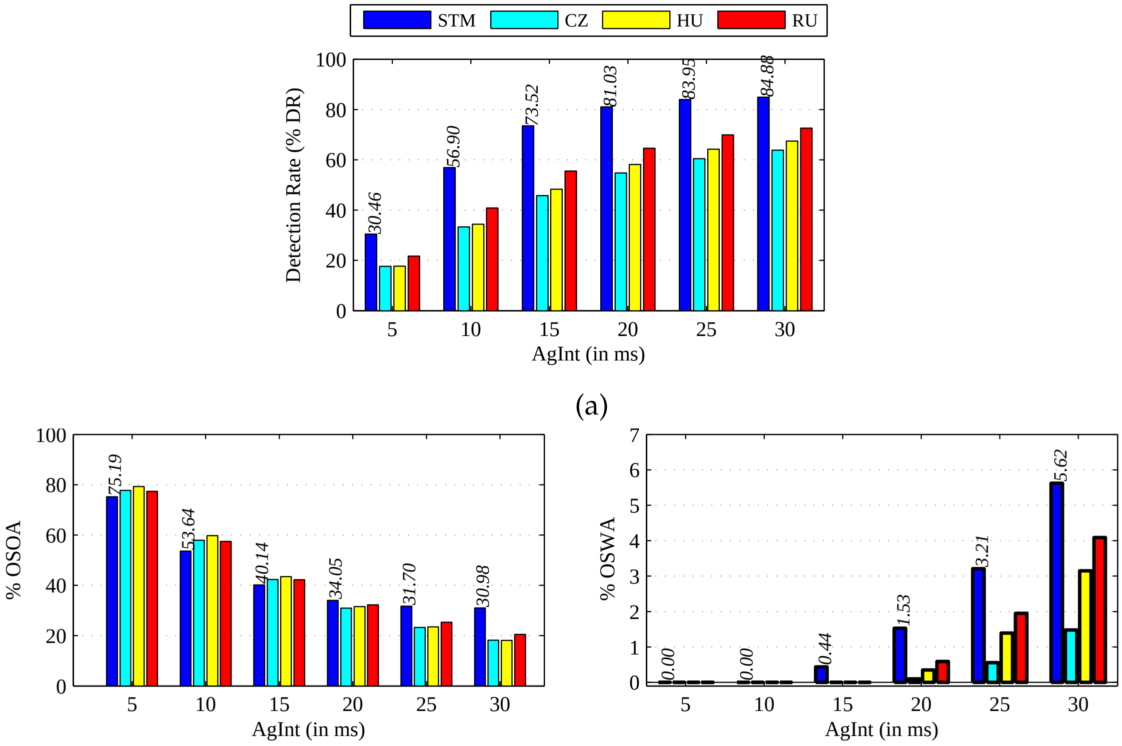
where ζ_i's are the ground truth boundaries. The evaluation metrics are % detection rate (% DR) within the agreement duration and % over segmentation within agreement (% OSWA) and % over segmentation outside agreement window (% OSOA). Formally, performance evaluation metrics are defined based on the position of hypothetical boundary (*HyB*) and agreement interval (*AgInt*), i.e.,

$$\%\ \text{DR} = \frac{\#\ \text{Times}\ \mathit{HyB}\ \text{fall within}\ \mathit{AgInt}}{\#\ \text{Total reference boundaries}} \times 100\ \%, \tag{5.12}$$

$$\%\ \text{OSOA} = \frac{\#\text{Times}\ \mathit{HyB}\ \text{fall outside}\ \mathit{AgInt}}{\#\text{Total}\ \mathit{HyB}} \times 100\ \% \tag{5.13}$$

$$\%\ \text{OSWA} = \frac{\#\ \text{Times}\ \mathit{HyB}\ \text{fall inside}\ \mathit{AgInt}}{\#\ \text{Total}\ \mathit{HyB}} \times 100\ \%, \tag{5.14}$$

The % DR should be high (ideally *100* %), and over segmentations rates % OSWA and % OSOA should be low (ideally *0* %). As shown in Figure 5.7, as agreement duration (% AgInt) increases, detection rate increases and over segmentation (false detection) outside decreases. It can be seen from Figure 5.7 that STM gave relatively higher % DR, lower % OSOA and lower % OSWA than the phoneme decoders. Hence, we have used STM for phone segmentation over the



(b) (c)

Figure 5.7: Performance of phonetic segmentation using STM and BUT's phonetic decoders (CZ, HU and RU): (a) % DR, (b) % OSOA, and (c) % OSWA.

phoneme decoders. Authors have used STM for various tasks, namely, for the obstruent sound detection from the speech signal [223] and the phoneme-level segmentation for Gujarati Text-to-Speech (TTS) synthesis development [224].

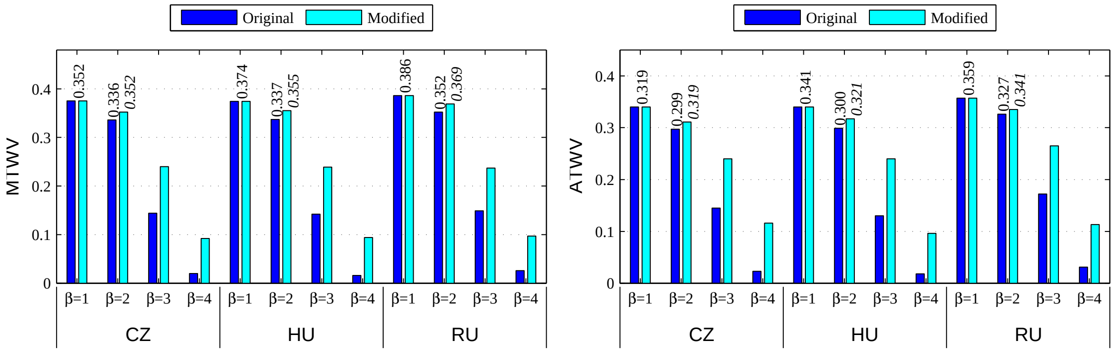
5.3.2 Results for SWS 2013

Table 5.3 shows the average time required to complete the search. The number in the bracket shows the time reduction w.r.t. *no* feature reduction case ($\beta = 1$). As the feature reduction factor β increases, the searching time is found to reduce drastically. It is also observed that as the dimension of feature vector increases, the search time also increases (which is only due to local distance computation). It can be observed from the Figure 5.8, as β increases, search performance degrades. This might be because of transition between two adjacent phonemes is averaged out. In addition, as suggested in [53], $\beta = 2$ is an appropriate choice, which adjusts the trade-off between search performance and searching speed. For all the values of feature reduction factor β (i.e., $\beta = 1$, $\beta = 2$ and so on), feature vectors are loaded before subDTW search. Hence, the CPU measured time does not indicate the only time for search execution. Thus, the time consumed in loading the

Table 5.3: The average time (minutes) required to execute search using feature reduction. The number in the bracket shows the (%) time reduction w.r.t. $\beta = 1$

Reduction Factor	Dev Set		Eval Set	
	Original	Modified	Original	Modified
$\beta = 1$	485.55		491.94	
$\beta = 2$	192.29 (60.40)	261.45 (**46.15**)	197.5 (59.85)	250.08 (**49.16**)
$\beta = 3$	131.35 (72.95)	182.95 (62.32)	136.71 (72.21)	175.95 (64.23)
$\beta = 4$	107.88 (77.78)	160.63 (66.92)	109.57 (77.73)	146.07 (70.31)

(original= feature reduction without phone boundaries into consideration, modified=feature reduction with phone boundaries into consideration (Pleaser refer to Figure 5.5))



(a)

(b)

Figure 5.8: Effect of feature reduction approach on performance of SWS 2013 (in terms of MTWV): (a) Dev set, and (b) Eval set. After [2].

feature vectors could be the possible reason behind obtaining less computational time reduction as expected from theoretical, i.e., $\frac{1}{\beta^2}$ for feature reduction factor β.

5.3.3 Results for QUESST 2014

As discussed, we found better performance with harmonic mean and feature reduction factor $\beta = 2$. Hence, the next set of experiments in this thesis use $\beta = 2$ and harmonic mean among partial DTW distances d_h. Figure 5.9 and Figure 5.10 show the performance of search space reduction using feature reduction on the Dev set and Eval set of QUESST 2014, respectively. It can be observed that performance after feature reduction gets reduced slightly and the performance of modified search is better than the performance of subDTW search for both $\beta = 1$ and $\beta = 2$. This supports our earlier results, which were discussed in Section 5.2. It is also shown in Figure 5.9 that performance with reduced number of feature vectors $\beta = 2$ using modified search is better than the simple search and no feature reduction (i.e., for $\beta = 1$). This indicates that we can obtain better performance with reduced number of features for non-exact match (*T2* query and *T3* query). However, the performance for an exact match is slightly worse because of the influence of other partial matching evidences (*T1* query). In the next Section, we will discuss the BoAW model to reduce the search space at the first-level and then DTW is performed on the selected test utterances to detect the query.

5.4 Segment-level Bag-of-Acoustic Words (BoAW)

In Bag of Acoustic Words (BoAW) approach, we perform search space reduction using two-stage operations. In the first stage, we select few audio documents (test utterances) by computing the score (modified cosine similarity) between BoAW (tf-idf) vector of audio document and query. In the second stage, subDTW is performed for QbE-STD on the selected test utterances. The number of selected test utterances are less as compared to the total test utterances and hence, the DTW takes less time for QbE-STD as compared to the time required to perform QbE-STD on the entire set of test utterances. Thus, BoAW helps in search space reduction. Figure 5.11 shows the schematic block diagram of the two-stage approach. In another words, segment-level BoAW use reasonably well defined short stretches of the speech signal to execute quick matching. These short stretches of speech signal span a predefined number of acoustic segments, where each acoustic segment more or less corresponds to a phone or a Gaussian components.

5.4.1 BoAW Model

BoAW term is derived from the "*Bag-of-words*" which is originally motivated from text-document retrieval [43]. To retrieve the similar word image pattern from the large word image datasets, the Bag of Visual Words (BoVW) was used for word image retrieval task [134]. The 'word' in BoAW is not directly related to the actual spoken word. In bag of acoustic word, the acoustic word is referred to as the class (i.e., phoneme label for phoneme posteriorgram and Gaussian component for GMM posteriorgram). The phonetic content of speech is not uniformly spread along the time and hence non-uniform segments should be created in order to preserve the similar speech production characteristics. The spectral transition measure is used to produce such segments and concatenation of these consecutive segments is regarded as a bag in BoAW framework. Thus, the bag corresponds to the the concatenation of the consecutive segments. In this thesis, we characterize

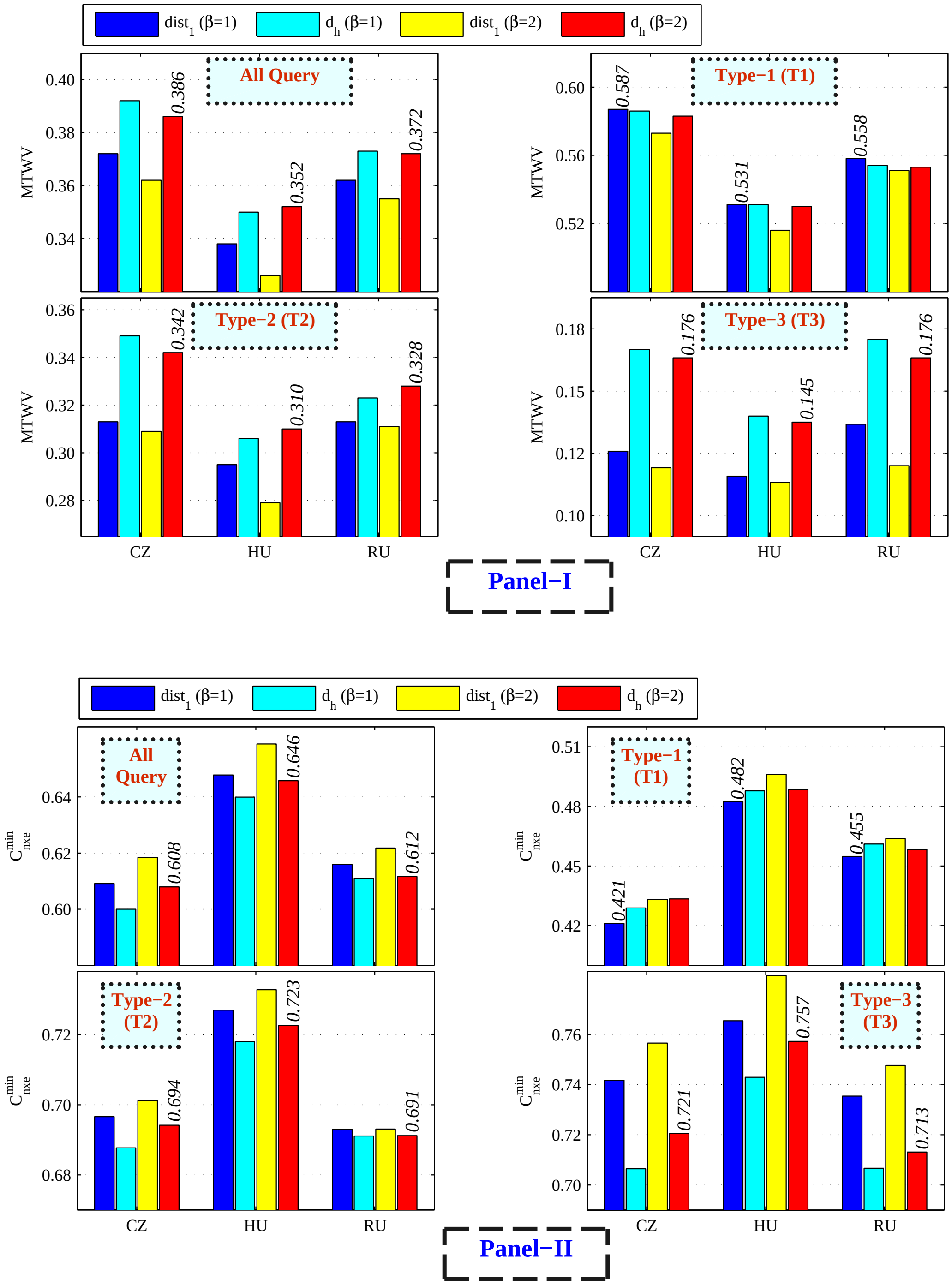


Figure 5.9: Performance of QUESST 2014 on Dev set for feature reduction. Panel (I) MTWV, and Panel (II) C_{nxe}^{min}. The numbers on the top of bar indicates the relative better performance between $dist_1$ with $\beta = 1$ and d_h with $\beta = 2$. After [2].

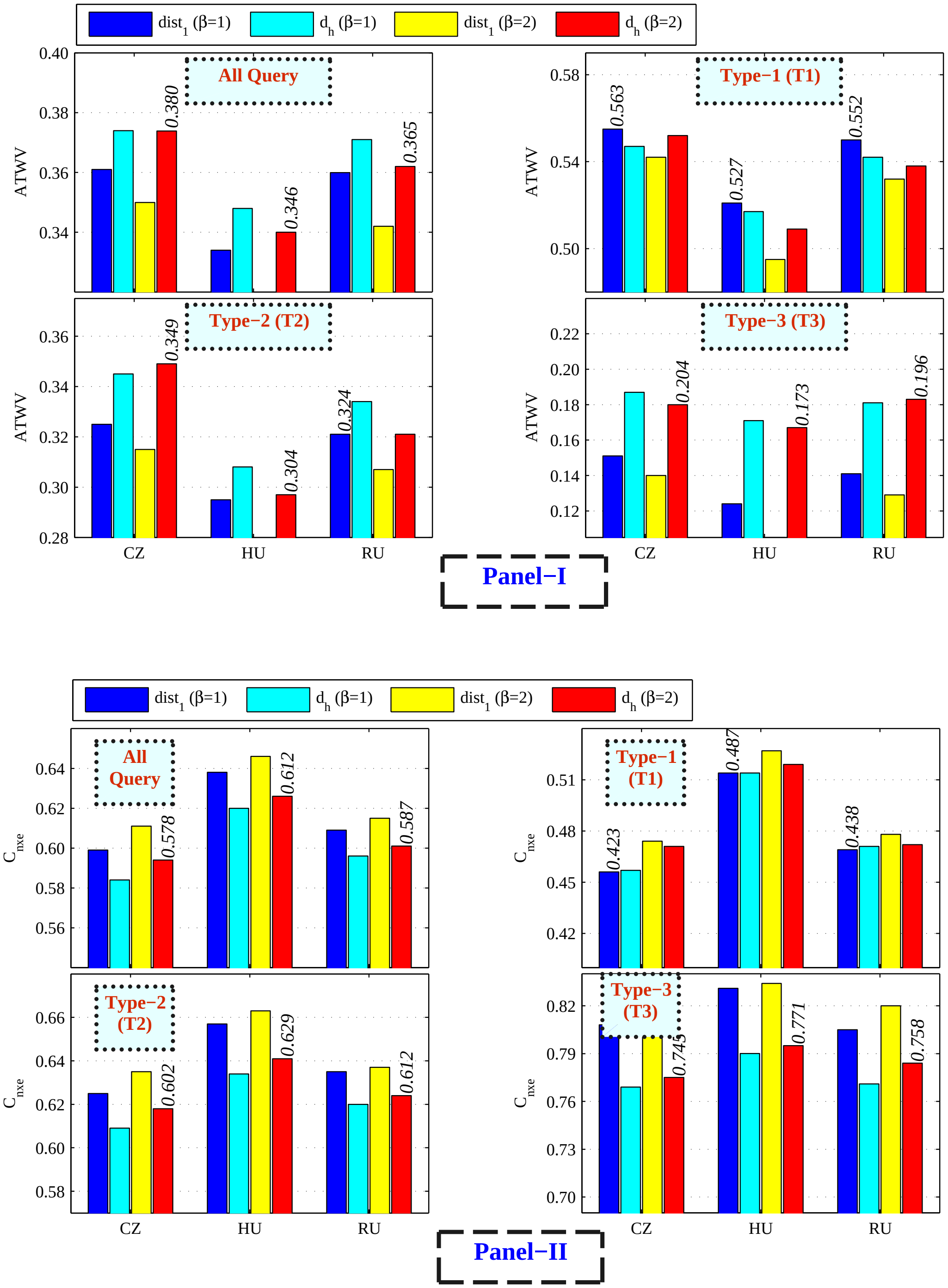


Figure 5.10: Performance of QUESST 2014 on Eval set for feature reduction. Panel (I) ATWV, and Panel (II) C_{nxe}. The numbers on the top of bar indicates the relative better performance between $dist_1$ with $\beta = 1$ and d_h with $\beta = 2$ in terms of MTWV and C_{nxe}^{min}. After [2].

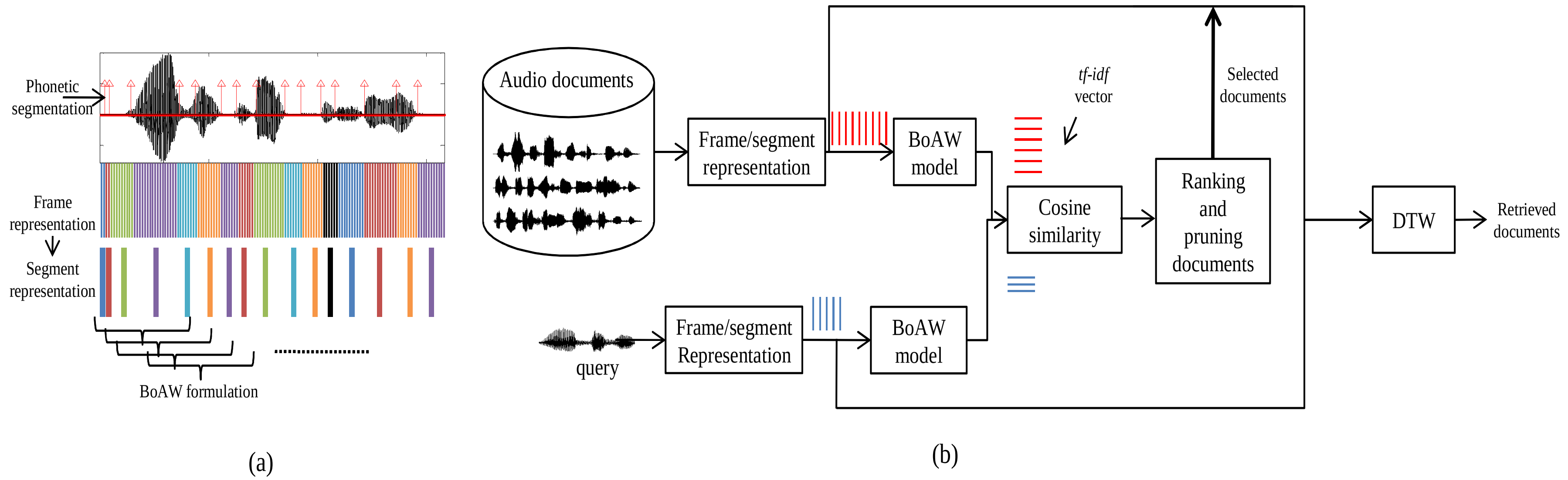


Figure 5.11: A schematic block diagram of two-stage search space reduction using segment-level BoAW: (a) Phonetic segmentation using STM (Ref. Figure 5.6) and BoAW formulation (phonetic segmentation is performed by computing STM) and (b) the schematic block diagram for two-stage search space reduction using segment-level BoAW. After [2].

BoAW by term frequency (*tf*) and inverse-document frequency (*idf*) at a segment-level. The term frequency, $tf(t,d)$ represents the total number times the term t is present in the document d. The inverse frequency document frequency for each term $idf(t) = \log\left(\frac{Nd}{df_t}\right)$, where df_t indicates number of documents that contains term t and Nd indicates total number of documents.

To understand this, consider schematic shown in Figure 5.11, which shows how term-frequency is computed at segment-level. We perform automatic phonetic segmentation using STM to group the feature vectors into acoustically homogeneous segments. The *segment* is referred to as phonetic segments obtained using STM. Then, we accumulate the consecutive segments to compute term frequency vector. This process remains the same for the spoken query as well. The phones and word boundaries are not directly available. Thus, we estimate the phone boundaries using spectral transition measure (STM). We experimented with different values of $Nseg$ (that corresponds to number of phones) for BoAW and we select the best value of $Nseg$ from the Dev set. The proposed approach accumulates local histogram properties and we may have multiple BoAW vectors for each query and test utterance. We denote d_i as i^{th} sub-document and $|D|$ as total number of sub-documents obtained after segment accumulation. We call d_i as sub-document because the test utterances in QbE-STD literature are also referred to as audio documents. In addition, we considered only the part of test utterance by combining the consecutive segments. Hence, the *sub-document* notation is used in this thesis.

In BoAW framework, the term t corresponds to t^{th} phonetic class of posterior feature representation, respectively. f_{t,d_i} indicates the sum of posteriors at t^{th} class (where, $t \in [1, N_p]$) in sub-document d_i. Formally, the definitions of term-frequency $tf(t,d_i)$ and inverse document frequency $idf(t)$ are as follows [43]:

$$tf(t,d_i) = f_{t,d_i}, \tag{5.15}$$

$$idf(t) = \log\left(\frac{|D|}{\sum_i^{|D|} f_{t,d_i}}\right). \tag{5.16}$$

5.4.1.1 Score Computation

The score between test utterance u_k and query q_j is defined as the maximum modified cosine similarity between the BoAW of test utterance u_k and query q_j, i.e.,

$$score(u_k, q_j) = \max_{l,m} < b_m^{u_k}, b_l^{q_j} >, \tag{5.17}$$

where $b_m^{u_k}$ and $b_l^{q_j}$ represent normalized *tf-idf* vector (BoAW) for m^{th} sub-document of k^{th} test utterance, u_k, and l^{th} sub-document of j^{th} query, q_j, respectively. The group of $Nseg$ consecutive segments features are combined to form the sub-document and characterized by *tf* vectors (in Figure 5.11 (a), $Nseg = 5$). The sub-document BoAW is computed by computing normalized *tf-idf* vector. Formally, $b_m^{u_k}$ and $b_l^{q_j}$ are defined as follows [2]:

$$b_m^{u_k} = \frac{tf(t, d_m^{u_k}) \circ idf(t)}{\|tf(t, d_m^{u_k}) \circ idf(t)\|}, \qquad \text{and} \tag{5.18}$$

$$b_l^{q_j} = \frac{tf(t, d_l^{q_j}) \circ idf(t)}{\|tf(t, d_l^{q_j}) \circ idf(t)\|}, \tag{5.19}$$

where $\|\cdot\|$ represents the l^2 norm and $\circ$ represents the Hadamard product (element-wise multiplication) between tf and idf vectors. Eq. (5.18) Eq. (5.19) give N_p-dimensional vector (i.e., number of Gaussians in GP and number of classes in phonetic posteriorgram). We considered the matching score between test utterance and query according to eq. (5.17). I and L are the total number of BoAW (total number of sub-documents) in test utterance u_k, and query q_j, respectively, which gives IL number of cosine similarity values. Hence, the $score(u_k, q_j)$ is set equal to the maximum value out of IL cosine similarity values. The spoken query can have more than one BoAW and to incorporate more than one modified cosine similarity score across BoAW, we have used the number of context (NC). NC is used to accumulate NC consecutive cosine similarity scores from the consecutive BoAW vectors. Hence, the eq. (5.20) is modified as [2]:

$$score(u_k, q_j) = \max_{1 \le l \le L, 1 \le i \le I - NC + 1} \sum_{m=i}^{i+NC-1} < b_m^{u_k}, b_l^{q_j} > . \tag{5.20}$$

Now, the score value, i.e., $score(u_k, q_j)$ takes the maximum value out of $L(I - NC + 1)$ number of accumulated cosine similarity scores.

5.4.2 Results for SWS 2013 using Phonetic Posteriorgram

The experimental results are discussed for four different test utterance selection (or pruning) procedures.

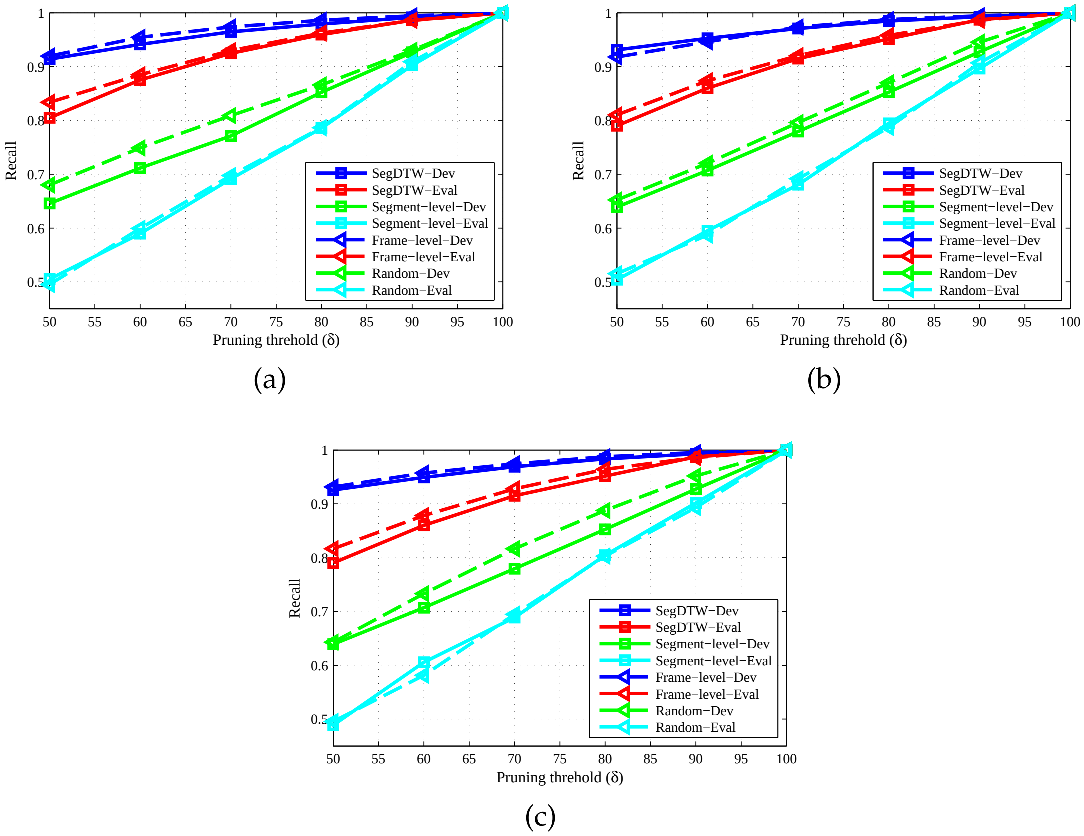
- **Random selection**: Here, test utterances are randomly selected from an entire set of test utterances. This selection does not incorporate the feature vector (posteriorgram) information into account.

- **Frame-merging**: A single *tf-idf* vector is computed for each test utterance and query. This is performed by averaging of the feature vectors (posteriorgram). The local *tf-idf* vector information is blunted and a single BoAW represents a query or a test utterance. The modified cosine similarity between the test utterance and query is used to rank the documents.
- **Proposed:** The group of *tf-idf* vectors, which corresponds to *Nseg* consecutive segments are combined to form a segment-level BoAW. As compared to the frame-merging BoAW, this approach accounts to local feature vectors in terms of *tf-idf* BoAW vectors. After taking the modified cosine similarity, a ranking is performed and test utterances are selected.
- **Segment-based DTW (segDTW):** In this approach, subDTW is performed onto the segment (inspired from [121]). We have used STM to segment the speech into phone-level, as discussed in sub-Section 5.3.1. The segDTW approach provides more stronger first stage for query detection as compared to the random selection approach.

5.4.2.1 Performance of the First Stage

We refer pruning threshold δ, which is the percentage of test utterances selected after ranking and the remaining ($100 - \delta$ %) utterances are pruned (or discarded). In a particular case, $\delta = 100$ indicates no pruning, i.e., an entire set of test utterances are selected. To investigate the effectiveness of proposed pruning on recall value, we vary δ from *50* % to *100* %. Figure 5.12 shows the recall values at different pruning threshold δ. It can be observed that recall values approach to *1* as δ increases from *50* to *100*. For a random selection, recall values follow a straight line, which intuitively makes sense because the presence of query is *uniformly distributed* among the test utterances. For a given pruning threshold δ, proposed segment-based approach gives a high value of recall than the frame merging. This might be because we hypothesize query detection by considering local BoAW vector. In addition, it can be observed that for a given pruning threshold δ, segDTW gave more recall values than our proposed segment-based BoAW. This is because of the fact that DTW algorithm provides temporal information in consecutive segments, whereas segment-based BoAW does not.

The computation of score for a given utterance and query requires more number of comparison operations in segDTW, whereas single comparison across all the possible modified cosine similarity is performed in segment-based BoAW approach. No comparison operation is performed in frame-merging BoAW because a single BoAW represent an utterance or a query. If M_s and N_s are the



(a)

(b)

(c)

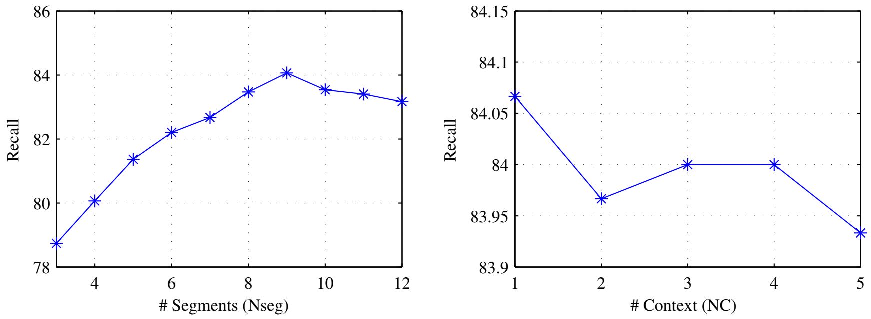
Figure 5.12: Recall values at various levels of pruning for different posteriorgrams: (a) CZ, (b) HU, and (c) RU posteriorgram. After [2].

total number of segments in test utterance and query, respectively, then order of comparison is $\mathcal{O}(M_s N_s)$ in segment-based DTW across three elements (selection of optimal warping path as discussed in Section 3.4, but for the segments rather than the frames), whereas a single comparison is performed for $NC = 1$ across $(M_s - Nseg + 1)(N_s - Nseg + 1)$ elements, indicating that proposed segment-level BoAW is computationally less complex than the segDTW. Time and storage computation is discussed in the next sub-Section.

Here, in this experiments, we varied the number of segments ($Nseg$) for BoAW formulation between *3* to *12*. The retrieval efficiency in terms of recall is shown in Figure 5.13. It was found that the combination $Nseg = 9$ and $NC = 1$ gave overall better performance for recall. However, the recall values does not vary significantly w.r.t. NC and hence, we will take $NC = 1$ in this thesis.

5.4.2.2 Performance of the Second Stage

The second stage performs DTW between a query and the selected test utterances. The performance in terms of MTWV is shown in Table 5.4 and Table 5.5. MTWV at δ = *100*, no pruning, considered entire test database for QbE-STD. Again, it



(a)

(b)

Figure 5.13: Average recall values of SWS QbE-STD for (a) different $Nseg$ and $NC = 1$, and (b) different NC and $Nseg = 9$. After [2].

can be observed that random selection follows linear increment in MTWV for most of the cases. The proposed selection approaches to MTWV at $\delta = 100$. In all the cases, proposed BoAW can give better MTWV over frame merging-based approach. The execution time and footprint size of BoAW are reported in Table 5.6. For the case of CZ-phonetic posteriorgram, the execution time of scoring using BoAW is about *41.05* sec, *829.24* sec and *4777* sec ($\approx$ 79 *minutes*) for frame-merging BoAW, segment-level BoAW and segDTW, respectively. In comparison, a segment-level BoAW approach takes high time for scoring because of multiple sub-document formations. However, the second stage of subDTW takes, even more time (about *445 minutes*), which is significant compared to the time required in first stage pruning. However, the segDTW takes 79 minutes, that is high as compared to the BoAW approaches. This might be because of computationally intensive DTW operation in segDTW. After pruning the test utterances in phonetic posteriorgram, MTWV is still comparable with several SWS 2013 baseline systems [31] (In particular as shown in Table 2.3, GTTS: 0.399, L2F: 0.342, CUHK: 0.306, BUT: 0.297, Proposed CZ at 50 % Pruning: *0.315*).

Table 5.4: Effect of BoAW pruning on search performance for SWS 2013 Dev set (in MTWV). The bold numbers indicate the highest MTWV performance for proposed approach at various pruning threshold δ. After [2]

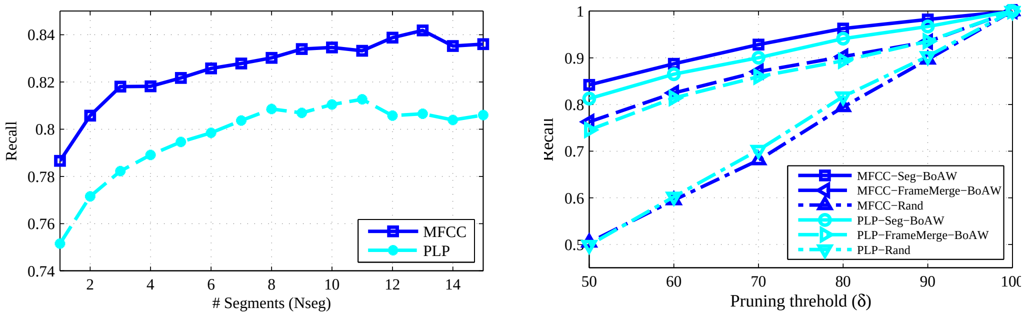
Feature Vector	Pruning Threshold (δ)	Proposed	Frame	segDTW	Random
CZ	50	0.357	0.260	0.375	0.201
	60	0.366	0.277	0.375	0.229
	70	0.373	0.296	0.375	0.270
	80	0.374	0.317	0.375	0.302
	90	0.374	0.345	0.375	0.339
	100	**0.375**			
HU	50	0.346	0.245	0.374	0.203
	60	0.361	0.268	0.374	0.231
	70	0.372	0.294	0.374	0.258
	80	0.373	0.321	0.374	0.292
	90	**0.374**	0.353	0.374	0.344
	100	0.374			
RU	50	0.360	0.243	0.385	0.189
	60	0.373	0.278	0.385	0.216
	70	0.377	0.311	0.385	0.270
	80	0.382	0.341	0.385	0.313
	90	0.385	0.372	0.386	0.348
	100	**0.386**			

Table 5.5: Effect of BoAW pruning on search performance SWS 2013 Eval set (in MTWV). The bold numbers indicate the highest MTWV performance for proposed approach at various pruning threshold δ. The numbers in the brackets indicate ATWV. After [2]

Feature Vector	Pruning Threshold (δ)	Proposed	Frame	segDTW	Random
CZ	50	0.315 (0.308)	0.235 (0.232)	0.341 (0.339)	0.176 (0.176)
	60	0.331 (0.318)	0.247 (0.243)	0.341 (0.340)	0.205 (0.199)
	70	0.335 (0.332)	0.261 (0.259)	0.342 (0.340)	0.233 (0.231)
	80	0.341 (0.332)	0.292 (0.288)	0.342 (0/340)	0.271 (0.268)
	90	**0.342** (0.339)	0.315 (0.314)	0.342 (0.340)	0.308 (0.303)
	100	0.342 (0.340)			
HU	50	0.312 (0.309)	0.226 (0.224)	0.340 (0.340)	0.165 (0.164)
	60	0.327 (0.327)	0.242 (0.241)	0.340 (0.340)	0.203 (0.201)
	70	0.333 (0.330)	0.261 (0.260)	0.341 (0.340)	0.232 (0.230)
	80	0.337 (0.334)	0.284 (0.283)	0.341 (0.341)	0.261 (0.259)
	90	0.340 (0.339)	0.315 (0.315)	0.341 (0.341)	0.304 (0.302)
	100	**0.341** (0.340)			
RU	50	0.324 (0.321)	0.248 (0.246)	0.358 (0.356)	0.174 (0.173)
	60	0.340 (0.338)	0.269 (0.267)	0.358 (0.355)	0.211 (0/211)
	70	0.350 (0.349)	0.290 (0.287)	0.359 (0.355)	0.251 (0.251)
	80	0.356 (0.352)	0.314 (0.309)	0.359 (0.355)	0.284 (0.284)
	90	**0.359** (0.357)	0.337 (0.337)	0.359 (0.355)	0.324 (0.323)
	100	0.359 (0.357)			

Table 5.6: Time and space requirement for BoAW stage on Dev set of SWS 2013. The bold fonts indicate the least space and time complexity using phonetic posteriorgram. (CPU hardware specifications: *64*-bit Intel *i5* @ *2.80* GHz, 4 GB RAM). After [2].

Feature Vector	Frame-merging BoAW		Segment-based BoAW		Segment-based DTW	
	Time (sec.)	Footprint (MB)	Time (sec.)	Footprint (MB)	Time (sec.)	Footprint (MB)
CZ	**41.05**	**1.77**	829.24	155.98	4777.00	169.02
HU	**41.50**	**2.42**	875.88	213.56	4984.40	231.87
RU	**41.09**	**2.05**	843.56	181.19	4938.40	196.52



(a) (b)

Figure 5.14: Recall performance of Gaussian posteriorgram representation: (a) for different *Nseg* and (b) at various levels of pruning threshold.

5.4.3 Results for SWS 2013 using Gaussian Posteriorgram

The experimental results are discussed for three different test utterance selection (or pruning) procedures.

5.4.3.1 Performance of the First Stage

At first, we ran the the experiments by varying the number of segments (*Nseg*) for BoAW formulation between 3 to 15 for Dev set query. The retrieval efficiency for different *Nseg* value, in terms of recall is shown in Figure 5.14 (a). It was found that the MFCC-GP, and PLP-GP posteriorgram gave relatively better performance for $Nseg = 13$, and $Nseg = 11$, respectively. We used these optimal *Nseg* values in BoAW computation for respective posteriorgram. Figure 5.14 (b) shows the recall values at different pruning threshold δ. It can be observed that recall values approach to *1* as δ increases. For a random selection, recall values follow a straight line, which intuitively makes sense because the presence of query is *uniformly distributed* among all the test utterances.

Table 5.7: Performance of SWS 2013 (in terms of MTWV) for feature reduction on Dev set.

Features Vector	Pruning Threshold (δ)	Segment-level BoAW	Frame-merging BoAW	Random Approach
MFCC-GP	50	0.186	0.163	0.101
	60	**0.188**	0.170	0.117
	70	0.188	0.175	0.134
	80	0.188	0.183	0.152
	90	0.188	0.186	0.171
	100	0.188		
PLP-GP	50	0.189	0.160	0.100
	60	0.194	0.167	0.114
	70	**0.195**	0.175	0.135
	80	0.195	0.18	0.16
	90	0.195	0.187	0.174
	100	0.195		

For a given pruning threshold δ, proposed segment-level approach gives a high value of recall than the frame-merging. This might be because we hypothesized query detection by considering local BoAW vectors. It can be observed that for both the posteriorgram cases, recall value of proposed segment-level BoAW at given pruning threshold (δ) is much higher than the other pruning approaches. In addition, for all the posteriorgrams considered here, the recall values of segment-level BoAW converges faster to *1*, than the frame-merging approach, indicating that the segment-level BoAW is relatively better pruning method.

5.4.3.2 Performance of the Second Stage

The second stage performs DTW between a query and the selected test utterances. The performance in terms of MTWV is shown in Table 5.7 and Table 5.8 for Dev and Eval sets, respectively. Again, it can be observed that random selection approach follows linear increment in MTWV for most of the cases. The proposed selection approaches to MTWV at $\delta = 100$. In all the cases, proposed BoAW can give better MTWV over frame-merging. The MTWV performance after pruning 70 % test utterances, does not deteriorate much and this holds consistent across all the three posteriorgrams. On the contrary, at $\delta = 70$, proposed test utterance selection procedure gave the same performance over respective $\delta = 100$, which might be due to the lower false alarm.

The execution time and footprint size of BoAW using different posteriorgram

Table 5.8: Performance of SWS 2013 (in terms of MTWV) for feature reduction on Eval set. The numbers in the brackets indicate ATWV.

Features Vector	Pruning Threshold (δ)	Segment-level BoAW	Frame-merging BoAW	Random Approach
MFCC-GP	50	0.137 (0.132)	0.128 (0.127)	0.072 (0.067)
	60	**0.138** (0.134)	0.131 (0.128)	0.082 (0.078)
	70	0.138 (0.135)	0.134 (0.131)	0.103 (0.102)
	80	0.138 (0.135)	0.137 (0.133)	0.110 (0.109)
	90	0.138 (0.135)	0.138 (0.135)	0.125 (0.121)
	100	0.138 (0.137)		
PLP-GP	50	0.143 (0.143)	0.136 (0.136)	0.066 (0.064)
	60	**0.145** (0.143)	0.140 (0.139)	0.091 (0.088)
	70	0.145 (0.145)	0.142 (0.141)	0.105 (0.103)
	80	0.145 (0.145)	0.143 (0.143)	0.118 (0.117)
	90	0.145 (0.145)	0.143 (0.142)	0.131 (0.131)
	100	0.145 (0.145)		

Table 5.9: Time and space requirements for BoAW stage on Dev set of SWS 2013 using Gaussian posteriorgram (CPU hardware specifications: *64*-bit Intel *i5 @ 2.80* GHz, 16 GB RAM)

Feature Vector	Frame-merging BoAW		Segment-level BoAW		sub. DTW
	Time (sec.)	Footprint (MB)	Time (sec.)	Footprint (MB)	Time (minutes)
MFCC-GP	**201.76**	**5.25**	1104.11	440.26	681
PLP-GP	**199.12**	**5.25**	994.43	450.54	669

representations are reported in Table 5.9. The footprint size of segment-level BoAW is larger than the frame-merging due to multiple BoAW for each utterance. This translates into more number of comparison for similarity computation and hence, the computational cost is more for segment-level BoAW. However, in comparison with only subDTW, the computational cost (in terms of time required for execution) is very low. As shown in 5.9, for MFCC-GP the execution time of scoring using BoAW is about *201.76* sec and *1104.11* sec for frame-merging and segment-level approach, respectively. However, the second stage of subDTW takes, even more time about *681 minutes*, that is significant as compared to the time required in first pruning stage. The same footprint size in frame-merging BoAW for MFCC and PLP is due to the same number of BoAW vectors and the same dimension of Gaussian posteriorgram representation.

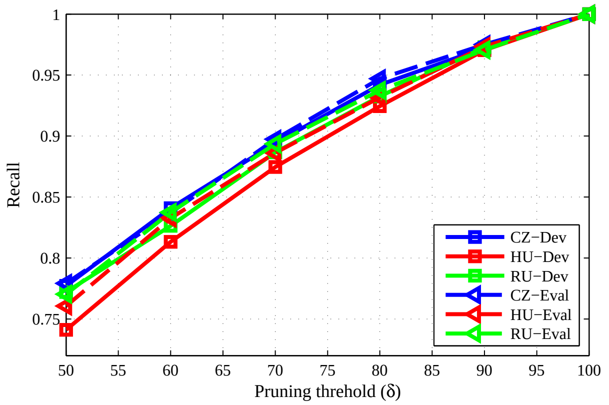


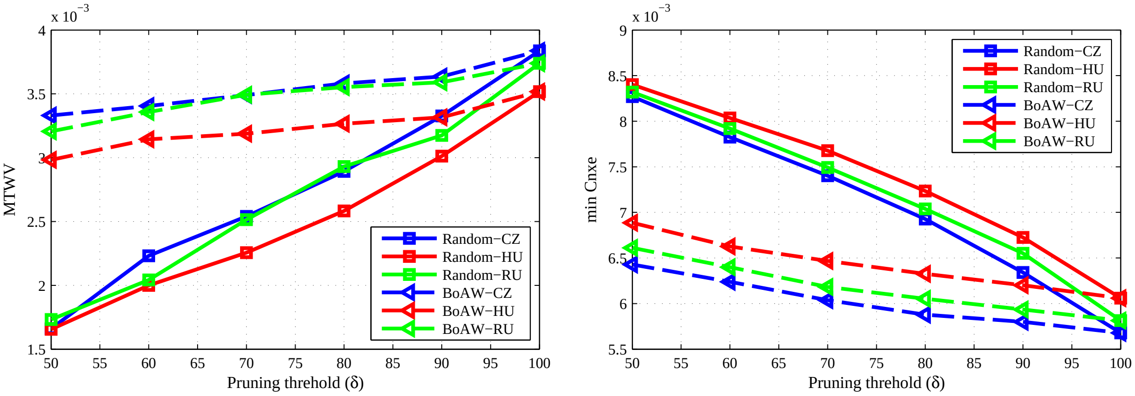
Figure 5.15: Recall values for at various levels of pruning QUESST 2014. After [2].

5.4.4 Results for QUESST 2014

As discussed in earlier sub-Section 5.4.2, proposed segment-level BoAW has a higher recall at every pruning threshold. Hence, we will perform segment-level BoAW pruning at first stage. The recall values for posteriorgrams on QUESST 2014 task are plotted in Figure 5.15. It can be seen that top *50* % test utterances, contains more than *70* % recall value, which approaches to *1* at the exponential growth. Again, it can be seen that curve is not exponential and hence, the growth is better than the random selection. As discussed in sub-Section 5.2, modified DTW search algorithm improves the search performance in the case of partial matching, which is part of QUESST 2014. We used modified DTW search with harmonic mean to combine the partial matching evidence for the selected test utterances from BoAW model. Figure 5.16 shows the MTWV and C_{nxe}^{min} for Eval set. It can be seen from Figure 5.16 that MTWV is more as compared to the random selection of test utterance. Hence, BoAW approach reduces the search space, which is useful for the QbE-STD task. In addition, it can be seen that MTWV and C_{nxe}^{min} gradually approaches to the performance without pruning. The value of $Nseg = 9$ is chosen empirically by tuning $Nseg$ from *3* to *12*.

5.5 Chapter Summary

This chapter presented the matching subsystems used in QbE-STD problem. We discussed partial matching strategies for non-exact query matching task. We found that this partial matching strategies improve the detection of non-exact query yet giving almost the equal performance for the exact query matching. In addition, we proposed two search space reduction approaches, namely, feature reduction approach and segment-level BoAW approach. The proposed feature



(a) (b)

Figure 5.16: Performance of BoAW model in terms of (a) MTWV and (b) C_{nxe}^{min}, for QbE-STD on QUESST 2014 Eval set. After [2].

reduction approach, which considers the phone boundaries into consideration, gave relatively better performance than the conventional feature reduction approach. Proposed segment-level BoAW gave relatively better performance than the frame-merging and random selection approach. In the next chapter, we will discuss multiple acoustic features and detection sources for re-scoring the detection of QbE-STD.

CHAPTER 6

Exploring Multiple Resources

6.1 Introduction

Most of the research studies in QbE-STD mainly focus on the representation of speech signal to improve the detection as a stand-alone system. QbE-STD problem aims to detect all possible presence of the spoken query within the audio documents. There could be many reasons, such as, variabilities across the speakers, recording channels, and the context. The earlier studies that combined the evidences (along with the relevance scores across multiple search systems) were found to be successful for QbE-STD task [62]. However, this approach requires the development of multiple QbE-STD systems, which is not feasible because as size of audio data grows, DTW takes huge time for searching. The another solution is to exploit multiple examples of spoken query, which can be performed by using either all the examples [39] or selectively combined the examples [59, 74, 225]. In QbE-STD, the retrieval output is heavily dependent on the single example of audio, and hence, the detected candidates may not be robust. Though the posteriorgram representation helps to eliminate the non-linguistic variation in the spoken query, we still lack the performance of QbE-STD. Previous studies used detection from the first retrieval to re-score the detection. Fewer studies discussed the information retrieval-based approach, i.e., relevance feedback to hypothesize the detection with additional query [121, 226].

In this context, this chapter presents a two-stage approach for re-scoring the detection hypothesis with the help of another acoustic features and detection sources (or detection cues). In other words, the objective is to use multiple acoustic features and detection sources to mainly combine the subDTW scores obtained with the posteriorgrams. The organization of this chapter is as follows: Section 6.2 discusses the several acoustic features used in *stage-2* for re-scoring. Section 6.3 discusses the detection sources and exploiting them to improve the performance obtained using Gaussian posteriorgram as well as phonetic posteriorgram. Ex-

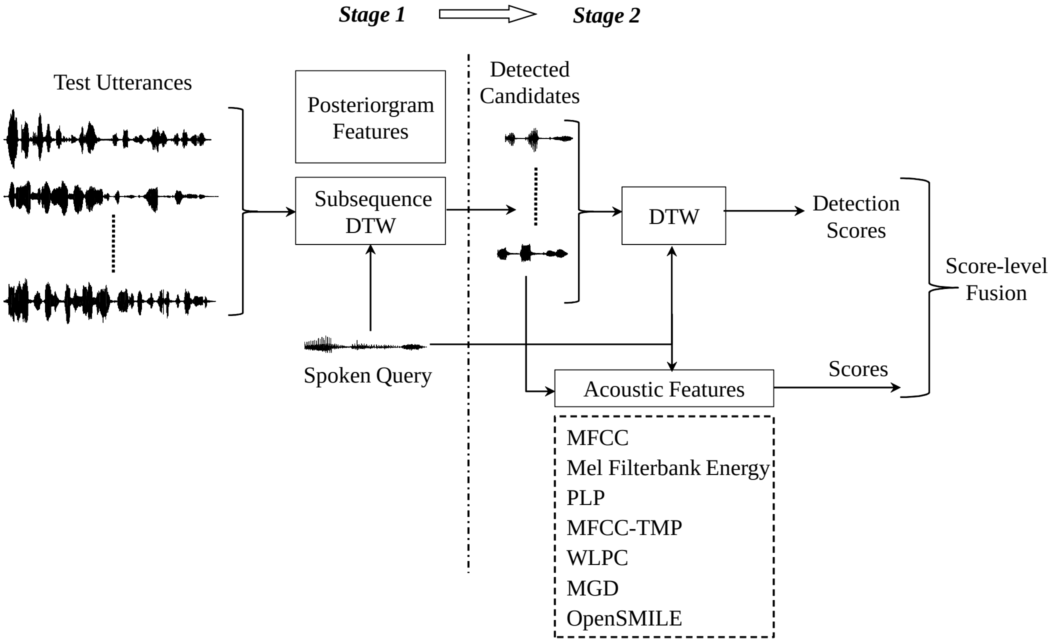


Figure 6.1: A schematic block diagram of proposed two-stage QbE-STD search system using acoustic features. The block arrow indicates the transition from *Stage-1* to *Stage-2*. Dotted box indicates acoustic features used in the framework. After [36].

perimental results for acoustic features, detection sources, and their score-level fusion are discussed in Section 6.3.6. The posteriorgrams used in *stage-1* are Gaussian posteriorgram (GP), VTL-warped GP, mixture of GMMs posteriorgram, and phonetic posteriorgram. We have used 128 number of clusters to compute the posteriorgrams and BUT phoneme recognizer to compute phonetic posteriorgram.

6.2 Acoustic Features

The schematic block diagram of the two-stage QbE-STD system with acoustic features is shown in Figure 6.1. In *stage-1,*posteriorgram features are used to perform subDTW to obtain fewer detected segments. In the *stage-2*, we used several acoustic features to improve the detection scores. The score-level fusion of all the detection scores gave the improved performance on SWS 2013 database.

We refer acoustic features as the first-level parameterization conventionally at short-time or segment-level (i.e., duration of 25 ms). To characterize production and perception properties, linear prediction (LP) [50], mel cepstrum [51], and LP-based features, such as, perceptual linear prediction (PLP) have been used as an acoustic representation. DTW is performed with detected segments and query. The global mean and variance are computed from all the features from test utterances, and we normalize the features (along with delta coefficients) with this mean and variance.

Pearson correlation distance between the features is used during DTW computation. The reason behind using the Pearson correlation distance is that it achieved superior performance. The Pearson correlation distance between two posterior vectors t_i and q_j is given by [46]:

$$D(t_i, q_j) = 1 - \frac{\langle t_i - \mu_{t_i}, q_j - \mu_{q_j} \rangle}{||t_i - \mu_{t_i}|| |q_j - \mu_{q_j}||}, \tag{6.1}$$

where $|| \cdot ||$ represents the l^2 norm, $<,>$ represents the dot or inner product, μ_{t_i} and μ_{q_j} are the mean of feature t_i and q_j, respectively. $\mu_{t_i} = \frac{1}{D} \sum_{k=1}^{D} t_i(k)$ and $\mu_{q_j} = \frac{1}{D} \sum_{k=1}^{D} q_j(k)$. The acoustic features used in this study are described in the next sub-Section:

6.2.1 Warped Linear Prediction (WLP)

This feature set provides Bark scale-based frequency warping via warped linear prediction (WLP). WLP is obtained by replacing unit delays of classical LP filter by first-order allpass filters with transfer function, which is given by [227]:

$$D(z) = \frac{z^{-1} - \lambda}{1 - \lambda z^{-1}}, \tag{6.2}$$

and phase response is given by [227]:

$$\Psi(\omega) = \omega + 2 \tan^{-1} \left(\frac{\lambda \sin \omega}{1 - \cos \omega} \right), \tag{6.3}$$

where $-1 < \lambda < 1$ is the warping factor. For $0 < \lambda < 1$, lower frequencies are compressed and higher frequencies are expanded. The reverse warping happens for $0 > \lambda > -1$. An analytical expression provides the value of λ for warping similar to Bark scale [227] depending on the sampling frequency (i.e., f_s) and is given by:

$$\lambda_{fs} \approx 1.0674 \left(\frac{2}{\pi} \arctan \left(0.6583 \frac{fs}{1000} \right) \right)^{\frac{1}{2}} - 0.1916. \tag{6.4}$$

In this thesis, 13 DCT coefficients of warped spectra are taken as the WLP coefficients.

6.2.2 Modified Group Delay Function

The negative derivative of the Fourier transform phase is called as group delay function. The group delay function has the better formant resolving capability than the magnitude spectrum of Fourier transform [228]. The modified group delay function is defined as follows:

$$\tau_x(k) = sign. \left| \frac{X_R(k)Y_R(k) + Y_I(k)X_I(k)}{S(k)^{2\gamma}} \right|^{\alpha}, \tag{6.5}$$

$sign$ is given by $\frac{X_R(k)Y_R(k)+Y_I(k)X_I(k)}{S(k)^2}$. Here, $X_R(k)$ and $X_I(k)$ are the real and imaginary parts of $X(k)$, respectively, (i.e., Fourier transform of $x(n)$), $Y_R(k)$ and $Y_I(k)$ are the real and imaginary parts of $Y(k)$, respectively, (i.e., Fourier transform of $y(n) = nx(n)$). $|S(k)|^2$ is the cepstral smoothed spectra, α and γ are the smoothing parameters, which are kept as 0.4 and 0.9, respectively.

6.2.3 OpenSMILE Library Features

The OpenSMILE library [229] is used to capture 36 acoustic features and their delta features. The list of features computed using the OpenSMILE library is given in Table 6.1. All these features were extracted at every 10 ms using 25 ms window, except for fundamental frequency (F_0), a probability of voicing, jitter, shimmer and Harmonic-to-Noise Ratio (HNR), where 60 ms window was used. The jitter and shimmer are used as to represent voice excitation source information, giving voicing related cues.

In addition to that, we also used MFCC, mel filterbank energy, PLP and MFCC-TMP acoustic features in *stage-2* for re-scoring. The details of MFCC, PLP and MFCC-MTP are given in sun-Section 3.3.1. We used 26-D mel filterbank energy features corresponds to 26 subband filters, where filters span 0-4000 Hz frequency regions.

6.2.4 Experimental Results

The DTW distance obtained with acoustic features and scores obtained using sub-DTW are fused at the score-level. Since the DTW is performed on the detected segments (as shown in Figure 6.1), all the detection candidates and corresponding detection scores are synchronized to the time stamps of the *stage-1*. The results are presented for four different types of posteriorgrams, namely, Gaussian posteriorgram, VTL-warped Gaussian posteriorgram, mixture of GMMs posteriorgram

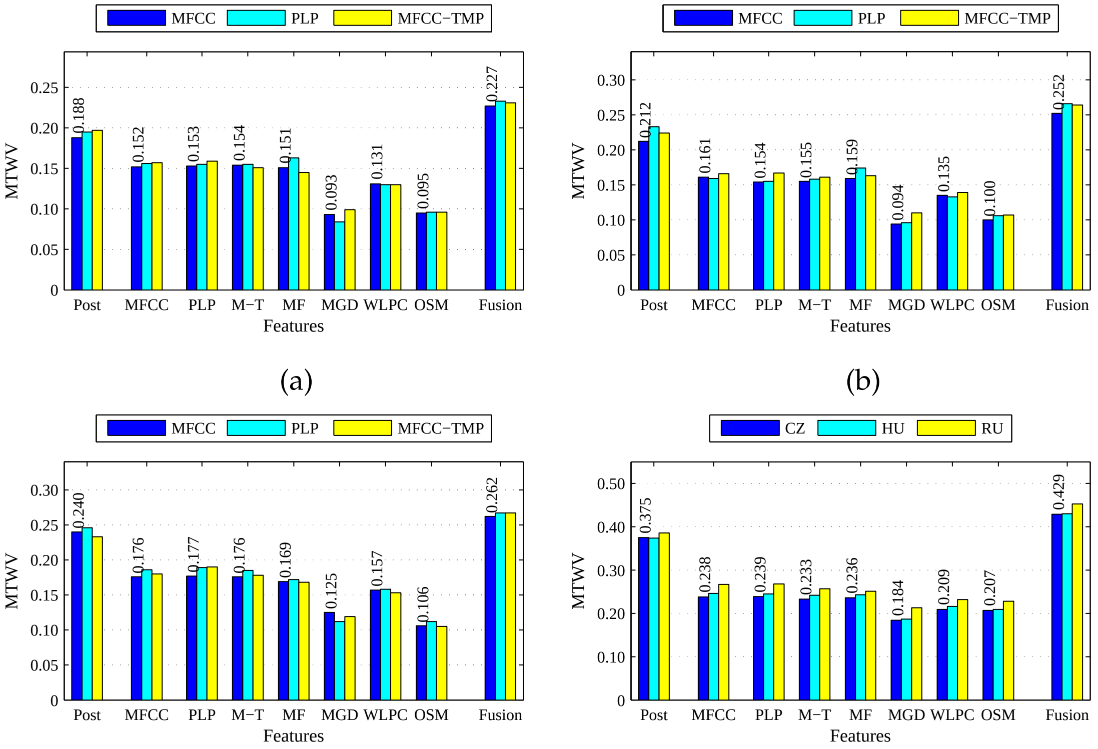
Table 6.1: Acoustic Features Extracted using OpenSMILE [229]

Description	# Features
Zero-Crossing Rate	1
Frame Intensity	1
Frame Loudness	1
Root Mean Square (RMS) Energy and log-energy	2
Energy in Frequency Bands 250-600 Hz and 1000-4000 Hz	2
Spectral Flux	1
Spectral Entropy	1
Spectral Variance	1
Spectral Skewness	1
Spectral Kurtosis	1
Spectral Sharpness	1
Spectral Harmonicity	1
Spectral Flatness	1
Line Spectral Pairs (LSP)	8
LPC Formant (Frequency and Bandwidth)	6
Fundamental Frequency (F_0)	1
Probability of Voicing	1
Voicing Quality	1
log Harmonic-to-Noise Ratio (logHNR)	1
Jitter (local and periodic variation)	2
Shimmer	1
Total	36

and phonetic posteriorgram. Scores obtained from different detection sources are normalized to have zero-mean and unit-variance and then combined using the discriminative fusion approach presented in [62] and discussed in Chapter 3.

The performance of QbE-STD systems is shown in Figure 6.2. The performance of raw acoustic features was measured by DTW similarity. It can be observed that the performance of each acoustic features is slightly lower than the GP. However, the performance due to cepstral features, such as, MFCC, PLP, MFCC-TMP and mel filterbank energy is much relatively higher than other features. This might be due to their capability of mimicking the human perception. The performance of WLPC and MGD are slightly lower. This might be because the coefficients are computed on the entire frequency range. In addition, MGD requires additional parameters to be tuned, i.e., γ and α. The acoustic features derived from OpenSMILE may give better performance if features are selected based on the dataset as suggested in [67].

Finally, the score-level fusion of all the systems gave better performance than the posteriorgram representation alone. Since the acoustic features used are the



(a)

(b)

(c)

(d)

Figure 6.2: Performance of two-stage QbE-STD systems using several acoustic features with (a) Gaussian posteriorgrams, (b) VTL-warped Gaussian posteriorgrams, (c) mixture of GMMs posteriorgram, and (d) phonetic posteriorgram system for SWS 2013 Dev set. Post = posteriorgram, M-T = MFCC-TMP, MF= mel filterbank energy, and OSM = OpenSMILE features.

same for different posteriorgram used in *stage-1* for subDTW, the performance is highly dependent on the time-stamps generated during *stage-1*. For instance, comparing the performance of OpenSMILE feature that is relatively lower than the other acoustic features. However, phonetic posteriorgram-based *stage-1* with OpenSMILE features gave *0.215* MTWV, which is almost double than all the other posteriorgrams used in *stage-1*. This observation is the same for all the acoustic features, which indicates the importance of miss detection. Similarly, VTL-warped GP and a mixture of GMMs posteriorgram in the *stage-1* gave relatively better performance than the GP. After applying the score-level fusion, GP, VTL-warped GP, a mixture of GMM posteriorgram and phonetic posteriorgram gave 0.235, 0.268, 0.265 and 0.437 MTWV scores, respectively. The use of different detection sources for re-scoring the detection is discussed in the next Section.

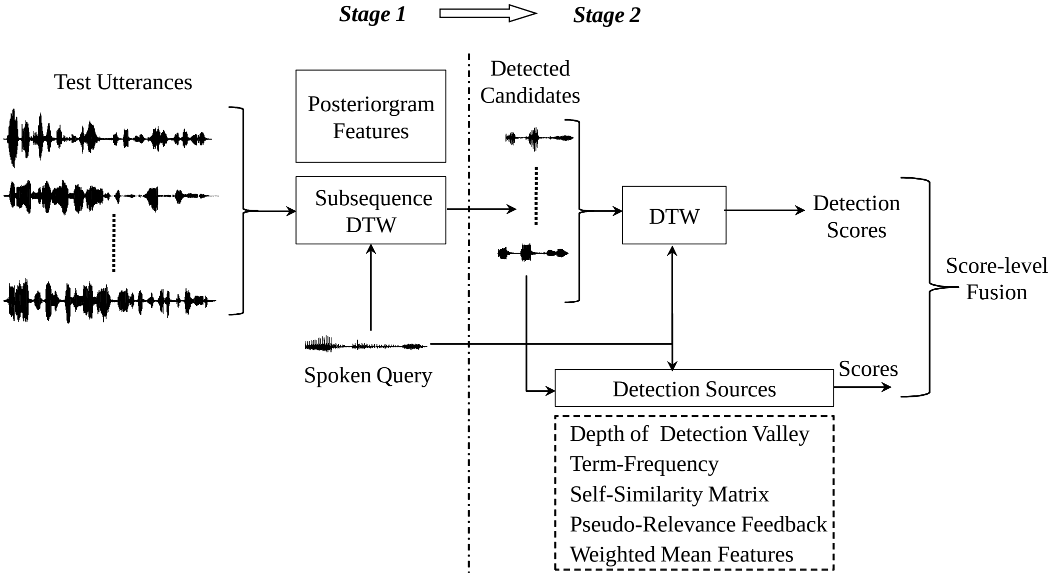


Figure 6.3: A schematic block diagram of proposed two-stage QbE-STD search system using detection sources. The block arrow indicates the transition from *Stage-1* to *Stage-2*. Dotted box indicates detection sources used in the framework. After [37].

6.3 Detection Sources

The schematic block diagram of the two-stage QbE-STD system with detection sources is shown in Figure 6.3. In *stage-1*, the posteriorgram features are used to perform subDTW to obtain fewer detected segments. In*stage-2*, we used several detection sources to improve the detection scores. The score-level fusion of all the detection scores gave the improved performance on QbE-STD task. These detection sources are generated from either single execution of DTW or no execution of DTW and hence, does not require additional computational overheads. In addition, we perform the re-scoring on the candidates obtained by using *stage-1*. The proposed detection sources are term frequency-Bag of Acoustic Word (BoAW), Self-Similarity Matrix (SSM), an average query using pseudo relevance feedback (PRF), depth of detection along warping path of DTW and weighted mean cepstral representation obtained using posteriorgram. The proposed two-stage block diagram is shown in Figure 6.3. Similar to the acoustic feature-based re-scoring, here also *stage-1* performs subDTW using posteriorgram representation and *stage-2*, performs re–scoring with the detection sources. All the detection candidates and corresponding detection scores are synchronized to the time stamps of the *stage-1*.

6.3.1 Depth of Detection Valley

The depth of the valley in the vicinity of warping path can be computed as follows [37]:

$$depth = \frac{1}{N}\sum_{i=1}^{N}\left(\max_{r\in\mathcal{S}_j}\left(\frac{S(i,r)}{T(i,r)}\right) - \min_{r\in\mathcal{S}_j}\left(\frac{S(i,r)}{T(i,r)}\right)\right), \tag{6.6}$$

where $\mathcal{S}_j$ is the set of test utterance frame index for which $P(i,j) = sp$, i.e., $\mathcal{S}_j \equiv \{j|P(i,j) = sp\}$, sp is the starting frame index for given warping path under consideration and N is the number of feature vectors in the query. The details of computation of matrices S, T, and P are given in Section 3.4. Earlier, depth of the valley was used to select the features [66]. The similarity plot and the computation of valley depth are illustrated in Figure 6.4.

6.3.2 Term Frequency Similarity

The Term-Frequency (TF) similarity score indicates the total number of times the term is present in the document. We define the Term-Frequency (TF) similarity score between the query q and the detection candidate sg as follows [37]:

$$score(q,sg) = \langle \frac{tf_q}{\|tf_q\|}, \frac{tf_{sg}}{\|tf_{sg}\|} \rangle, \tag{6.7}$$

where $\|\cdot\|$ represents the l^2 norm and $<,>$ represents the dot or inner product. The TF vector tf_q and tf_{sg} are normalized by its l^2 norm before computing the similarity score. The term corresponds to the Gaussian component and phonetic class for Gaussian posteriorgram and phonetic posteriorgram, respectively [133]. Hence, TF is computed simply by summing the posterior probabilities for each component.

6.3.3 Self-Similarity Matrix (SSM)

The SSM represents the (dis)similarity between a pair of feature vectors within a segment [146, 230]. The SSM of feature vectors $\mathbf{x}_t$ (where $0 \leq t \leq T$) is a $T \times T$ squared symmetric matrix such that $\Phi(i,j) = d(\mathbf{x}_i, \mathbf{x}_j)$, where $d(.,.)$ defines the distance between two feature vectors $\mathbf{x}_i$ and $\mathbf{x}_j$. The feature vectors used to carry linguistic information as well as non-linguistic information, such as, speaker information and channel information. Thus, the feature vector $\mathbf{x}_j$ can be modeled as an additive noise model, i.e., $\mathbf{x}_i = \mathbf{s}_i + N$, where $\mathbf{s}_i$ and N resemble the linguistic and non-linguistic characteristics from the speech signal, respectively. The SSM is

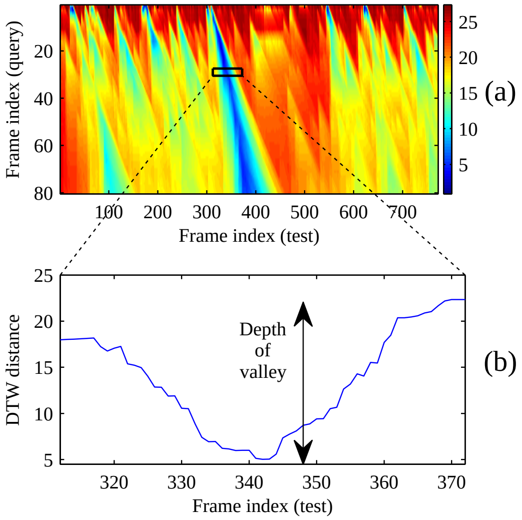


Figure 6.4: An illustration of depth of detection valley: (a) similarity normalized accumulated distance matrix between the query and test utterance, the patch along the warping path is shown in rectangle box and (b) the DTW distance value within selected rectangle box showing the depth of the valley surrounded by warping path. After [37].

computed for a small segment that is expected to be from the same speaker and the same channel. Hence, under this additive noise model, the SSM is expected to show linguistic information present in the speech segment. Many empirical visual observations suggest that different instances of the same word spoken by different speakers or undergoing different recording channels exhibit similar visual resemblance [146]. In addition, SSM brings out the similarity and dissimilarities between all the feature vectors of spoken query and detected candidates. Hence, the difference between these two can be an important detection source. The lesser the SSM value indicates better similarity between the query and detection candidates. The SSMs for the query and a segment of the test utterance are shown in Figure 6.5.

6.3.4 Pseudo Relevance Feedback (PRF)

The user relevance feedback has been widely used in text retrieval problem [43]. Inspired from the Information Retrieval (IR) literature, several studies in QbE-STD problem exploited the concept of relevance feedback. In relevance feedback scenario, the detection candidates (i.e., the part of spoken audio detected) at first few hits are assumed to be close to the query. Hence, these detected part of spoken audio may be treated as a query (which is referred to as *pseudo-query* or *pseudo-relevant example*) and the searching can be employed with this query. This approach has been called as pseudo-relevance feedback (PRF) [145, 226]. The scores

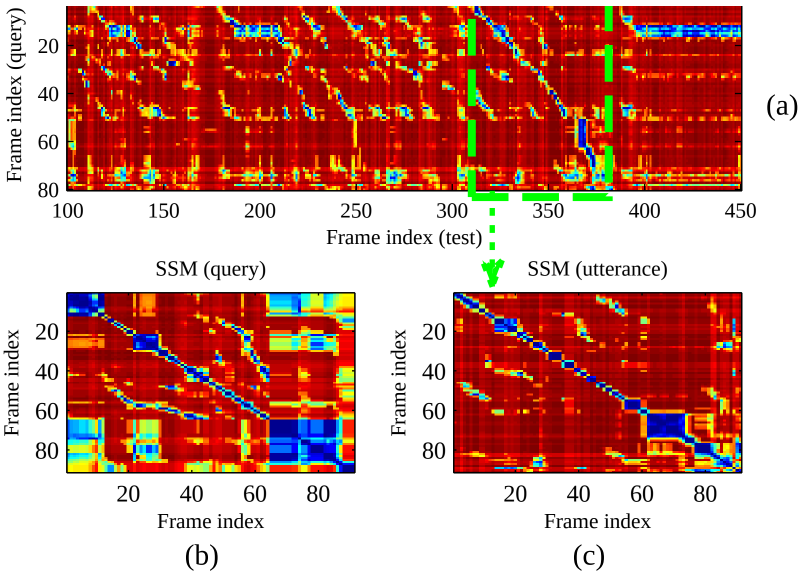


Figure 6.5: An illustration of Self-Similarity Matrix (SSM): (a) similarity local distance matrix between the query and test utterance, the dotted rectangle block shows the detected candidate within utterance and (b) SSM for query and (c) SSM for detected candidate. After [37].

obtained using pseudo queries are merged to the scores using actual query. To avoid the error for using pseudo-query directly, we generate the average query with the help of the original spoken query and the pseudo-query. We exploit the original query and align each feature of query onto pseudo-query so as to produce the average query. The average query X_{avg} for two different spoken queries $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{L_X}]$ and $Y = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{L_Y}]$ can be performed as follows [74]:

$$X_t^{avg} = \frac{1}{1 + |\mathcal{S}_t|} \left(X_t + \sum_{v \in \mathcal{S}_t} v \right), \tag{6.8}$$

where $\mathcal{S}_t$ is the set of all the features of Y that are aligned to t^{th} frame of X (i.e., $\mathbf{x}_t$) and $1 \leq t \leq L_X$. We have used three different average queries with top 3 pseudo-relevant queries. The average query obtained using pseudo-query is shown in Figure 6.6, which shows that the average query is very much similar to the actual query.

6.3.5 Weighted Mean Features

The correspondence between phonetic classes and Gaussian components is one-to-many, i.e., each phonetic class is defined in terms of a group of Gaussian components. In other words, multiple Gaussians are aligned to a single phonetic class [97]. The posterior probabilities for a given phonetic unit are split into mul-

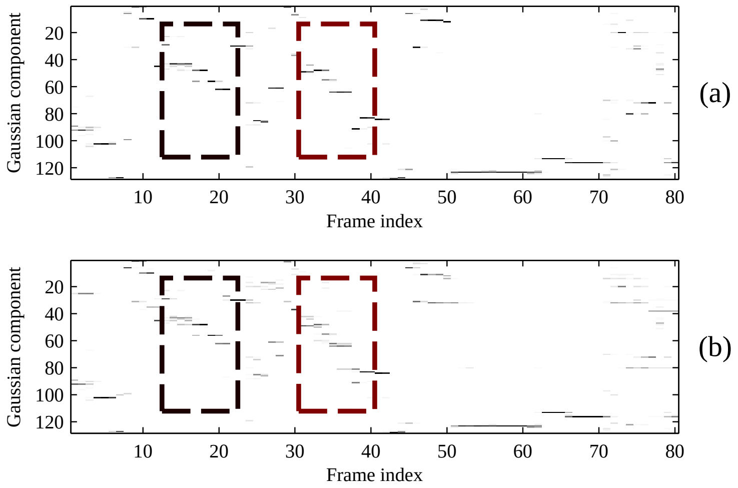


Figure 6.6: An illustration of average query with PRF: (a) Gaussian posteriorgram for query and (b) Gaussian posteriorgram for dynamically aligned time average query with first detection candidate. The dotted boxes show similarities in both representations. After [37].

tiple components, since the many of Gaussian components are close to each other in the feature space. With an assumption that the means of Gaussian components form the basis functions of cepstral features. A feature is represented in terms of the linear combination of means of Gaussian components. The weights in the linear combination are assigned by the Gaussian posterior probabilities. This representation assumes that means of Gaussian components constituted the basis functions of spectral feature space and known as weighed-mean feature [97]. For NG components, the posterior probability of data $\mathbf{o}_t$ being in k^{th} Gaussian component is given by [77]:

$$P(C_k|\mathbf{o}_t) = \frac{w_k \mathcal{N}(\mathbf{o}_t; \mu_k, \Sigma_k)}{\sum_{j=1}^{NG} w_j \mathcal{N}(\mathbf{o}_t; \mu_j, \Sigma_j)}, \tag{6.9}$$

where (w_k, μ_k, Σ_k) are model parameters of GMM. The weighted mean feature $\hat{\mathbf{x}}_t$ can be expressed as follows [97]:

$$\hat{\mathbf{o}}_t \approx \sum_{j=1}^{Np} \mu_j P(C_j|\mathbf{o}_t). \tag{6.10}$$

The rational behind using weighted mean features is that it minimizes the fluctuation around mean and eliminates the variabilities. Thus, the resulting weighted mean features are better than the original acoustic representation, i.e., MFCC, PLP or MFCC-TMP. We use this weighted mean features as an additional detection source to subDTW scores. To compare two weighted average cepstral features, Pearson correlation distance metric is used. The pictorial view of cepstral repre-

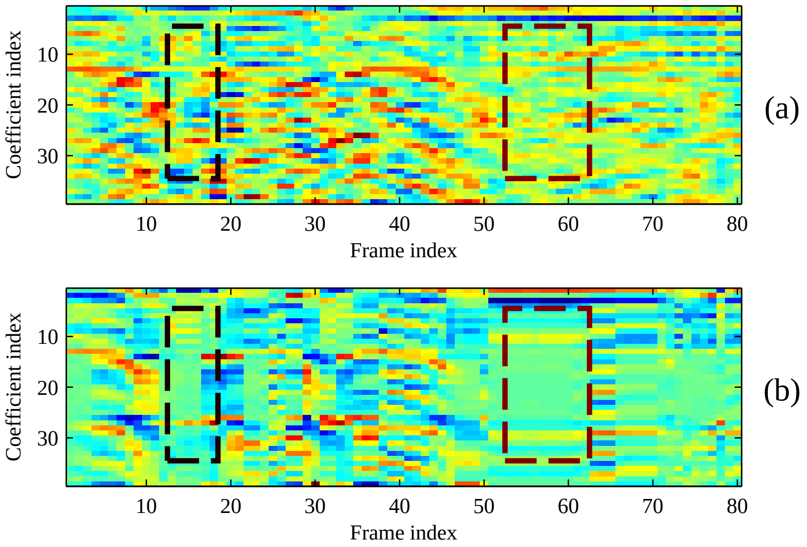


Figure 6.7: An illustration of weighted mean representation of a query: (a) MFCC representation of query, and (b) weighted mean feature of query. The dotted boxes on (b) shows the smoothed feature of corresponding dotted boxes on (a). After [37].

sentation and weighted mean cepstral features is shown in Figure 6.7. It can be seen that weighted mean cepstral representation is *smoothed* version of cepstral features (indicated by dotted boxes).

6.3.6 Experimental Results

The experimental results with posteriorgram representation are shown in Figure 6.8. The results are presented for four different types of posteriorgrams, namely, GP, VTL-warped GP, a mixture of GMMs posteriorgram and phonetic posteriorgram. Scores obtained from different detection sources are normalized to have zero-mean and unit-variance and then combined using the discriminative fusion approach presented in [62] and as discussed in Chapter 3. We have not used a weighted mean representation for a mixture of GMMs and phonetic posteriorgram. The average MTWV for a weighted mean feature for SWS 2013 Dev set is *0.148* and *0.155* for GP and VTL-warped GP, respectively. The relative better performance (in MTWV) with VTL-warped GP than the GP indicates VTL-warped GP as better representation than the GP. MTWV obtained using Term-Frequency (TF) is very low, since it does not exploit the temporal dynamic matching. The TF relatively gave better performance with phonetic posteriorgram, however, due to this missing temporal information, it possibly contributes very less in the score-level fusion.

The valley depth along warping path is another detection source that indi-

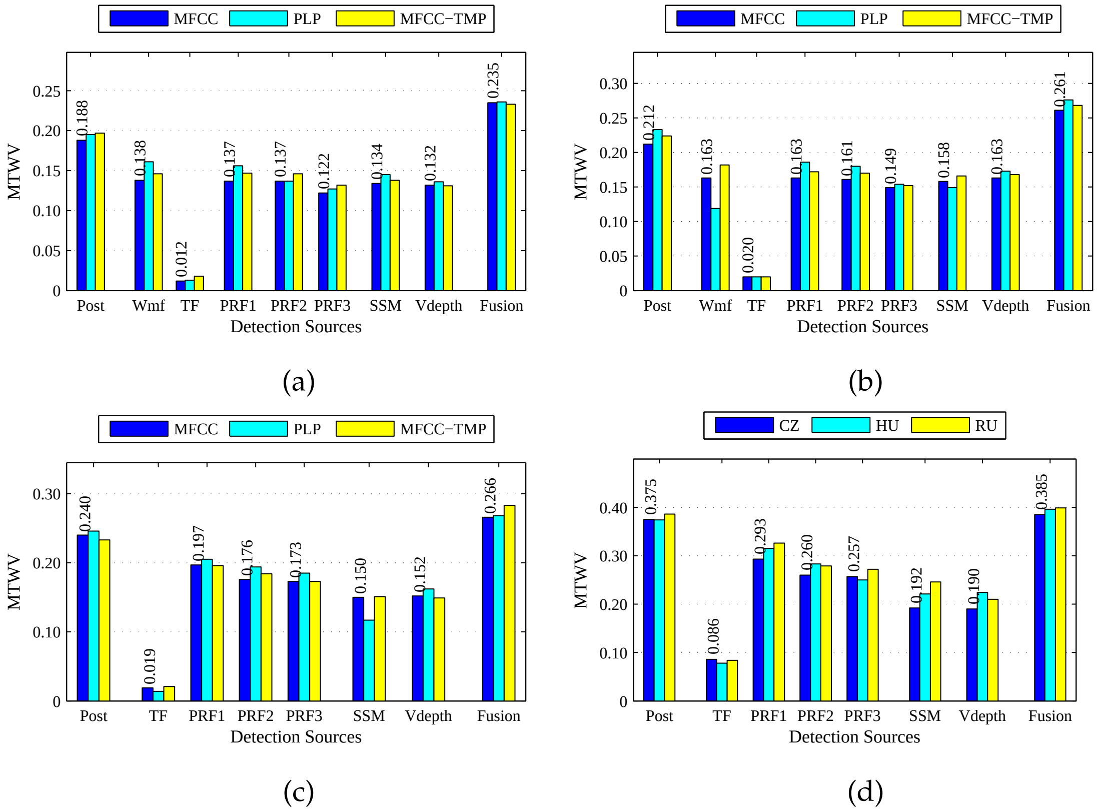


Figure 6.8: Performance of of two-stage QbE-STD systems using several detection source with (a) Gaussian posteriorgrams, (b) VTL-warped Gaussian posteriorgrams, (c) mixture of GMMs posteriorgram, and (d) phonetic posteriorgram system for SWS 2013 Dev set. Post = Posteriorgram, Wmf= Weighted mean features, PRF1=pseudo relevance feedback with 1st detected candidates, PRF2=pseudo relevance feedback with 2nd detected candidates, PRF3=pseudo relevance feedback with 3rd detected candidates, Vdepth=Depth of detection valley. After [37].

cates the presence of query in an utterance. The average MTWV for valley depth for SWS 2013 Dev set is *0.133*, *0.168*, *0.154* and *0.208* for GP and VTL-warped GP, respectively. VTL-warped GP and mixture of GMM posteriorgram gave relatively better performance than the GP. The performance of detection valley with phonetic posteriorgram is better than all the posteriorgrams. SSM and weighted mean feature detection sources gave relatively better performance as they resemble acoustic property well. PRF1, PRF2, and PRF3 indicate the performance due to 1st, 2nd and 3rd pseudo-relevant examples, respectively. As shown from Figure 6.8, the performance in MTWV slightly declines from PRF1 to PRF3. This is intuitively correct, since the 1st pseudo-query is more relevant to the original spoken query than the 3rd pseudo-query. With SSM and weighted mean features, we have a similar observation. From Figure 6.8, comparing the performance across different posteriorgrams, it can be observed that VTL-GP and mixture of GMMs gave better performance than the GP. Furthermore, to incorporate their contribution, the score-level fusion of Gaussian posteriorgram features is presented. The score-level fusion of all the detection sources gave on an average improvement of *4.1* %, *4.5* %, *3.3* % and *1.5* % in MTWV for GP, VTL-warped GM, mixture of GMMs posteriorgram and phonetic posteriorgram, respectively, than their posteriorgram alone. The relatively better performance improvement with the GP and VTL-warped GP suggest the importance of re-scoring with the help of detection sources for unsupervised posteriorgram.

In addition from Figure 6.8, we can observe that the relatively consistency in the results across all the different posteriorgrams used. The performance of valley depth is slightly improved. The Gaussian posteriorgram are computed from multivariate Gaussian and their posterior probabilities are split across many Gaussian components. Phonetic posteriorgrams are derived from trained phoneme recognizer supervisedly and they produce relatively distinct posterior probability than the corresponding Gaussian posteriorgram (Please refer Figure 2.4 for phonetic posteriorgram and Figure 2.5 for Gaussian posteriorgram). Hence, Gaussian posteriorgram may give a much false alarm in detection valley depth relatively than the phonetic posteriorgram. Again, the score-level fusion improves the performance than the phonetic posteriorgram alone. In particular, the score-level fusion of phonetic posteriorgram along with their various detection sources considered in this study gave on an average improvement of *1.5* % in MTWV indicating that detection sources carry complementary information than the phonetic posteriorgram alone. After score-level fusion for phonetic posteriorgram, MTWV is comparable with several SWS 2013 baseline systems [31] (In particular as shown in Table 2.3, GTTS: 0.399, L2F: 0.342, CUHK: 0.306, BUT: 0.297, Proposed (CZ): *0.355*).

6.3.6.1 Combining Acoustic Features and Detection Sources

Next, we combined the evidences from all the acoustic features and detection sources for four different types of posteriorgrams, namely, Gaussian posteriorgram, VTL-warped Gaussian posteriorgram, a mixture of GMMs posteriorgram and phonetic posteriorgram. The performance with a fusion of these multiple evidence, is shown in Table 6.2. As discussed in sub-Section 6.2.4, the acoustic features gave relatively better improvement for phonetic posteriorgram due to better representation. In addition, as discussed earlier in this sub-Section that detection sources improve the performance for unsupervised posteriorgram, such as, GP and VTL-warped GP. Thus, for Dev set on average, MTWV scores after combining all the evidences are improvement of *6.4* %, *6.8* %, *4.9* % and *6.8* % for

Table 6.2: Score-level fusion of several acoustic features and detection sources

	Dev Set			Eval Set		
Stage1 Post. Fusion	MFCC	PLP	MFCC-TMP	MFCC	PLP	MFCC-TMP
Posteriorgram	0.188	0.195	0.197	0.138	0.145	0.147
Detection sources	0.235	0.236	0.233	0.166	0.167	0.173
Acoustic Features	0.227	0.233	0.231	0.166	0.173	0.173
All	**0.257**	**0.261**	**0.253**	**0.187**	**0.189**	**0.193**

(a) Gaussian posteriorgram

	Dev Set			Eval Set		
Stage1 Post. Fusion	MFCC	PLP	MFCC-TMP	MFCC	PLP	MFCC-TMP
Posteriorgram	0.212	0.233	0.224	0.160	0.166	0.177
Detection sources	0.261	0.276	0.268	0.188	0.195	0.201
Acoustic features	0.252	0.266	0.264	0.193	0.204	0.199
All	**0.284**	**0.297**	**0.293**	**0.216**	**0.217**	**0.218**

(b) VTL-warped Gaussian posteriorgram

	Dev Set			Eval Set		
Stage1 Post. Fusion	MFCC	PLP	MFCC-TMP	MFCC	PLP	MFCC-TMP
Posteriorgram	0.240	0.246	0.233	0.195	0.208	0.201
Detection sources	0.266	0.268	0.283	0.217	0.229	0.245
Acoustic features	0.262	0.267	0.267	0.216	0.237	0.223
All	**0.287**	**0.286**	**0.292**	**0.239**	**0.257**	**0.256**

(c) Mixture of GMMs posteriorgram

	Dev Set			Eval Set		
Stage1 Post. Fusion	CZ	HU	RU	CZ	HU	RU
Posteriorgram	0.375	0.374	0.386	0.341	0.341	0.361
Detection sources	0.386	0.396	0.400	0.355	0.356	0.375
Acoustic features	0.429	0.430	0.453	0.386	0.392	0.428
All	**0.438**	**0.442**	**0.457**	**0.399**	**0.401**	**0.441**

(d) Phonetic posteriorgram

GP, VTL-GP, a mixture of GMMs and phonetic posteriorgram, respectively.

6.4 Chapter Summary

In this chapter, we explored various detection sources, namely, Term-Frequency, Self-Similarity Matrix (SSM), depth of detection along DTW warping path, pseudo-relevance feedback generated an average query and weighted mean feature, obtained using posteriorgram for QbE-STD. The several detection sources are exploited on the candidates obtained after subDTW. The results obtained using each detection source is encouraging as a stand-alone detector. Furthermore, the score-level fusion brings complementary information in score assignment resulting in better performance than the posteriorgram alone. However, the performance of unsupervised Gaussian posterior is still lacking, which demands further exploration. In the next chapter, the overall summary of the thesis is presented with its limitations and future research directions.

CHAPTER 7

Summary and Conclusions

This chapter presents the summary of the entire thesis work, limitations of current work, and future research directions.

7.1 Summary of the Thesis

In this thesis, novel VTL-warped Gaussian posteriorgram is presented to remove the speaker variations between query and test utterance. Proposed GMM-based framework does not require phonetic transcription yet powerful to estimate VTLN warping factor estimation and phoneme recognition. We further proposed an iterative scheme to VTLN and reduced number of features for QbE-STD. A mixture of GMM was proposed to introduce broad phonetic priors in unsupervised GMM training. A mixture of GMM posteriorgram performs better for QbE-STD than the GMM posteriorgram. Their score-level fusion is found to improve the search performance further.

The modified DTW search algorithm to deal with partial matching is presented. The need for partial matching is essential, when the instance of query present in test document has variations either at suffix, prefix or word re-order. The proposed modified DTW search algorithm combines the evidences from various partial matching strategies via different functions, namely, harmonic mean, arithmetic mean and minimum value. MediaEval QUESST 2014 database was used to investigate the effectiveness of partial matching strategies. We found that the combined distance pulled using harmonic mean gave relatively better performance than the subDTW for a non-exact match (i.e., T2 and T3 queries), whereas slightly degrades the performance for the exact match (T1 query) because of the influence of other partial matching distances.

In the next part of the thesis, we discussed computational improvements during DTW-based searching using two approaches, namely, feature reduction approach and BoAW approach. We adopted two feature reduction schemes, namely, approach that merges consecutive features, and proposed approach merges consecutive features within phone segment. In proposed feature reduction approach, we merged consecutive feature vectors within phonetic segment boundaries and executed subDTW by reduced number of feature vectors. Thus, a lesser number of feature vectors reduces the computational cost by reducing the number of comparison operations. We found that feature reduction factor $\beta = 2$ gave comparable performance with no feature reduction case (i.e., $\beta = 1$). Theoretically, computation time is reduced from $\mathcal{O}(MN)$ to $\mathcal{O}(\frac{MN}{\beta^2})$, where M and N are the number of features in test utterance and query, respectively, for feature reduction factor $\beta \in \mathbb{N}$. Practically, we obtained slightly lower MTWV with almost two times computational requirements for SWS 2013 evaluation. For QUESST 2014 evaluation, we observed that feature reduction $\beta = 2$ with modified DTW search algorithm gave relatively better performance than the conventional subDTW.

BoAW is two-stage search approach for a QbE-STD task. In the first stage, BoAW models are built by computing the term-frequency (tf) and inverse-document frequency (idf). In posterior feature framework, term corresponds to the phoneme class. The term-frequencies are computed from the entire test set of SWS 2013 and QUESST 2014. Proposed segment-based BoAW model constructs term vector over phoneme segments, and tf and idf vectors are computed. As a comparison, we built BoAW per test utterance and per query by merging frames. The similarity score between test utterance and a query is performed by modified cosine similarity. After scoring, test documents are retrieved based on the scoring. Proposed segment-level BoAW gave more recall values than the frame-based BoAW and random selection. We conduct subDTW (for SWS 2013) or modified DTW (for QUESST 2014) to evaluate the search on selected test utterances by BoAW. Proposed approach reduces the search space and gave better performance (in terms of recall, MTWV and C_{nxe}^{min}) over random selection.

There is a trade-off between search performance and search execution time. Given advanced hardware and computation, the performance is relatively more important than the execution time. The speed of execution is an important metric to quantify real-time processing of searching and to improve the detection by searching again with multiple query examples as well as relevance feedback scenario. Under PRF scenario [226], we can use the detected location at the top hit within test utterance as a pseudo-query. This query can be used to perform QbE-STD again and improve the detection [121]. Hence, faster execution time helps in detection error correction. The QbE-STD system is formed by cascading multiple subsystems. Thus, the errors introduced by one subsystem affects the performance of other subsystems. To avoid such errors, we eliminate spurious detection at different stages, such as, speech activity detection and warping path selection. SAD removes the silences around queries. During warping path selection, the length of warping path and duration of a query is compared. To improve the detection performance, we may include pseudo-relevance query, i.e., detection within the test utterance at the top hit.

7.2 Limitations of the Work

- We considered 5 iterations in an iterative framework for VTLN warping factor estimation. As observed in Figure 4.5, the likelihood increases as iteration increases. We need Dev query set to stop the iterations, which might not give the optimal performance (MTWV) for Eval set.
- In mixture GMM posteriorgram, we considered fixed number of Gaussian components. Several studies, such as, Dirichlet Process Gaussian Mixture Model (DPGMM), Minimum Description Length (MDL) can be used to self-determine the optimal number of Gaussian components, when modeling each broad phoneme class. This might improve the design of posteriorgram and can avoid overfitting.
- All the experiments are conducted on SWS 2013 and QUESST 2014. One of the limitations of these data is they do not contain reverberant speech. However, QbE-STD in reverberation and noisy environment definitely improves the applicability of the system in real-life settings.
- We presented the partial matching approach for the detection of the non-exact query. The results for partial matching shows slight degradation for the case of the exact query matching (Type-I query).

7.3 Future Research Directions

The future modification can be suggested from various aspects of subcomponents. Few future research directions could be as follows:

1. *Use of Language Recognition:* In multilingual audio retrieval scenario, the usage of language identification (LID) provides additional information. The LID scores can prune many hypotheses at first-level and hence, it can be useful in search space reduction. In addition, it can also be useful as a side information along with detection score [148, 149].

2. User Interface and Deployment: The information technology (IT) development demands rapid deployment of a user interface as to retrieval audio information. Various applications can be thought, which exploits voice search that includes spoken dialog systems [231]. In a practical system, the user feedback can be used to generate the secondary (pseudo) query from the detection [232].

3. *Use of signal processing for better audio representation :* One can explore the possible signal information processing schemes to exploit voice source, vocal tract and modulation information. It was observed that syllabic information is prominent around the vowel onset location [233]. In fact, few studies have been exploited the syllable nuclei at first level of audio search [234]. In addition, Hierarchical Agglomerative Clustering (HAC) has been used for the segmenting the speech signal in terms of phone-like units [121]. The future research efforts can be directed towards exploiting region-specific VTLN warping factors [206], to investigate their effect on the performance of QbE-STD tasks (the regions formed by a division of frames of an utterance). In the present work, VTLN warping is performed on the frequency-domain and hence, Jacobian computation is difficult, thus the objective function neglects Jacobian into consideration. It would be interesting to observe the effect of Jacobian by considering LT-VTLN as suggested in [202, 235].

4. *Use of Signal Processing for Fast Audio Matching:* DTW-based approach has been widely used for QbE-STD system design. DTW takes more time as data size grows, in particular, the number of comparison operation to execute the search task for a query having length N frames and test utterance having length M frames is of the order of $\mathcal{O}(MN)$ using subsequence DTW. The signal processing-based constraint might avoid few detection [97, 128]. Earlier, syllable nuclei and the syllable segmentation were explored to reduce the search complexity. This is still an open area of research. In addition, we have not explored the advanced hardware, such as, Graphical Processing Units (GPU) to parallelize of computation during searching the query [123, 124, 236].

5. *Use of deep learning models:* It has been analyzed that restricted Boltzmann Machines (RBM) [79] can learn the distribution of features and can form an alternative representation of Gaussian posteriorgram. In addition, it was also observed that deep belief networks could be used to modify the Gaussian posteriorgram [80]. There is further scope to exploit unsupervised and supervised models, such as, convolution network, a recurrent network for modeling the speech data. Furthermore, weighted mean representation from Gaussian posteriorgram can be useful to convert Gaussian posteriorgram representation to

feature representation [97]. Recently, multilingual bottleneck features (BNF) were used to capture the multilingual information into lower dimensional bottleneck layer [237].

6. *Other detection sources:* Many times the performance QbE-STD depends only on the DTW scores. Other detection evidences, such as, depth of the detection valley [37], SSM [146], PRF [226] and term-frequency vector [36] are expected to give the additional information, which might be useful for QbE-STD. Many other pieces of evidences were exploited, such as, duration, a number of phonemes, language recognition score, etc. in order to combine the scores for improving the QbE-STD performance [60]. The use of LID can be very much useful especially in the case of multilingual QbE search [149]. A potential future work could be to remove the redundant features in the score-level fusion. In addition, one can explore the data augmentation approaches [238] and query expansion approaches [158] to improve the performance.
7. *Non-exact query :* DTW due to its monotonic property has to be modified to perform non-exact query matching task. Partial matching techniques were used in [83,239], where multiple warping paths were used for backtracking. For non-exact matching, query splitting approach into two bands or three bands were employed in [148]. The study presented in [60] discusses the phoneme boundaries and approximation phoneme search. The major issue with all these approaches is their average performance. Each approach is specifically designed for particular kind of partial matching. Hence, to combine the scores from various strategies without affecting the performance much, is still an open area of research. That could be a possible reason for slight degraded performance of Type-I query with proposed partial matching approach.
8. *Query-by-Humming (QbH) :* Consider a scenario where a person wants to retrieve a ringtone of a particular song. If a person forgets the correct lyrics and words about this song, then the text-based information retrieval system cannot be used here [182]. In that context, the QBH paradigm has been introduced to perform music information retrieval (MIR) task [240]. The approaches used in QbE-STD, such as, posteriorgram, DTW, etc. can be explored for QbH framework as an independent research problem in its own right. Earlier, derivative DTW-based approach was used to retrieve Hindi humming songs with humming query [241]. In addition, the progressive filter (PF) framework was proposed to speed up the DTW computation for query-by-singing/humming (QbSH) task in [242]. QbSH systems can also be extended to person-dependent mode [182].

Appendix A. TIMIT QbE-STD

TIMIT data contains training and testing sets. Spoken queries are taken from the training set and testing set forms audio documents, where query needs to be searched. TIMIT database contains good quality speech recording in American English [243]. TIMIT dataset is used to perform ranked evaluation of QbE-STD system, where QbE-STD system rank the utterance according to their relevance to the query. The task is to retrieve the test utterances that contains the query rather locating them in utterance.

This experimental setup neglects /sa/ sentences from TIMIT, which is common across all the speakers of TIMIT. Table A.1 shows the list of keywords used for the experiments of QbE-STD in this thesis. #Train and #Test indicate the number of times query present in training (total examples) and testing set (total occurrences), respectively. We refer to this setup as *Q-20*. We have also considered more number of queries in TIMIT QbE-STD. In this experimental setup, we have used 84 queries that contains 7 to 20 occurrences in the testing dataset and having at least six letters. Spoken queries are taken from the training dataset. We will refer this to as *Q-84*. All the queries are distributed across all the speakers such that at least one speaker contains at least one query.

Table A.1: List of keywords used in TIMIT QbE-STD. Inst. refers to total instances of a query, Occ. refers to total number of occurrences

Keyword	**Inst.**	**Occ.**	**Keyword**	**Inst.**	**Occ.**	**Keyword**	**Inst.**	**Occ.**
Artists	7	7	Intelligence	8	7	Problems	5	7
Beautiful	3	8	Love	4	16	Shellfish	14	7
Birth	4	7	Marriage	3	7	Simple	8	8
Destroy	9	7	Meeting	7	7	Surface	3	7
Development	9	8	Morning	15	14	Tomorrow	3	7
Garbage	8	7	Ocean	7	7	Youngsters	7	7
Government	14	8	Organizations	7	7			

A.1 VTL-warped Gaussian Posteriorgram

This Section discusses the QbE-STD system developed for TIMIT dataset. The TIMIT QbE-STD system performance is evaluated on p@N and MAP [7,43]. QbE-STD systems with different posteriorgram representations are considered. Gaussian posteriorgrams and VTL-warped Gaussian posteriorgrams are extracted on *64* mixture components. ASM models are built using Gaussian Component Clustering (GCC)-based segment labeling as suggested in [96, 244]. Here, we considered *61* labels and performed VTLN using the Lee-Rose method using unsupervised decoded ASM labels. To evaluate the clustering using ASM, we compute clustering purity and Normalized Mutual Information (NMI) as defined in [96]. The clustering purity of ASM model for MFCC, PLP and MFCC-TMP is *0.386*, *0.392* and *0.404*, respectively, and NMI of MFCC, PLP and MFCC-TMP is *0.371*, *0.373* and *0.383*, respectively. It can be seen from Table A.2, the performance of VTLN is better than the conventional Gaussian posteriorgram for each case.

The performance improvement using GMM-based VTLN warping factor estimation is consistent with all the three cepstral representations. The performance of MFCC-TMP is better than the MFCC and PLP. This might be because MFCC-TMP exploits Teager Energy Operator (TEO) on the subband signal, which is a different form of energy measure than the usual l^2 norm (in the sense of computation) used (for short-time energy calculation) in MFCC.

The performance of GMM-based VTLN warping factor estimation is better than the ASM. This can be explained as follows. The HMM-based framework requires transcription in order to compute the likelihood and estimate VTLN warping factor, α (as per Eq.(4.2)). The warping factors in VTLN for short utterances are not consistent because of incorrectly decoded transcriptions. This problem is apparent for QbE-STD tasks due to the short duration of queries. It was expected that Lee-Rose VTLN warping factor estimation gave better performance. However, study shows that Lee-Rose VTLN warping factor estimates are *inconsistent* across the same speaker in case of short utterances [191]. The reasons for this can be explained as follows. The short duration of utterance contains very fewer observation vectors, and the warping factor estimation depends on the decoded transcription and hence, any error in decoded transcription affects the VTLN warping factor estimation significantly. In GMM-based approach, VTLN warping factor estimation depends on only the model parameters, not the transcription. The estimates are much away in the case of HMM-based VTLN warping factor estimation than the GMM-based VTLN warping factor estimation, which

Table A.2: Performance of TIMIT QbE-STD systems for individual query example for VTL-warped Gaussian posteriorgram.

Feature sets	VTLN	p@N	MAP
MFCC	×	34.91	36.71
	P	**36.26**	**39.62**
	ASM-×	28.77	28.89
	ASM-✓	30.69	31.54
PLP	×	35.50	37.58
	P	**37.48**	**39.95**
	ASM-×	30.89	32.57
	ASM-✓	34.26	36.17
MFCC-TMP	×	40.03	41.50
	P	**42.88**	**45.52**
	ASM-×	30.41	31.29
	ASM-✓	31.94	32.00

(× = No VTLN, P= GMM-based approach, ASM-× = ASM No VTLN and ASM-✓=ASM with VTLN)

is shown in Figure A.1. The HMM-based approach for VTLN warping factor estimation is not reliable for shorter duration utterances [193].

A.1.1 Effect of Number of Gaussians

It can be analyzed from Figure A.2 that an increasing number of mixture components improves the performance of a QbE-STD system. This finding matches a previous study reported in [53]. This might be because of the increasing number of clusters better represents the speech signal at the frame-level. However,

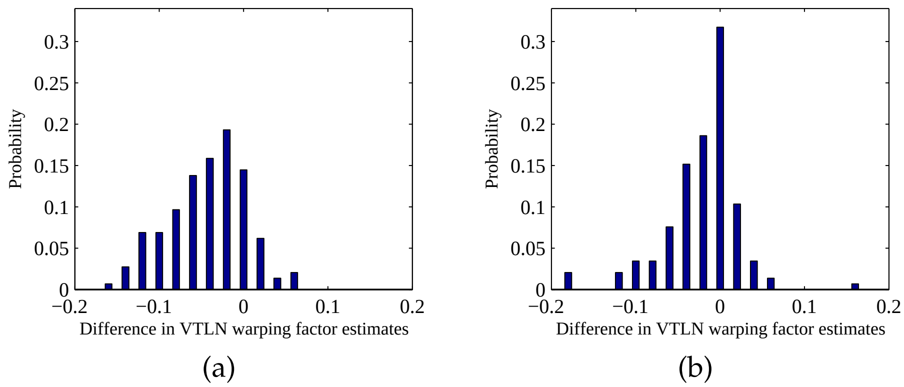
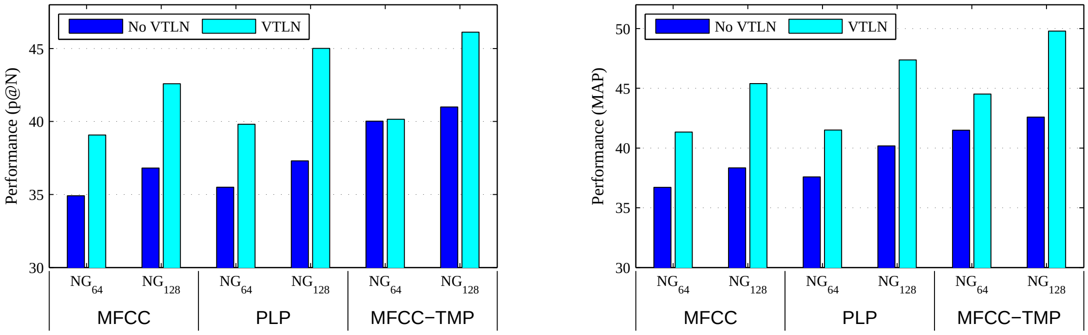


Figure A.1: Probability density function (*pdf*) of the difference between two VTLN warping factor estimates of shorter and longer utterances from the same speaker: (a) using the Lee-Rose (HMM-based) method, and (b) using the GMM-based method.



(a) (b)

Figure A.2: Effect of the number of Gaussians on Q-84 TIMIT QbE-STD systems on performance. (a) p@N, and (b) MAP. After [1].

increasing number of Gaussians demands additional processing and storage cost and hence, we restrict our experiments till *128* number of clusters. In addition, performance using the proposed approach is better than the Gaussian posteriorgram. Hence, we will use 128 Gaussian components in GMMs in the next set of experiments.

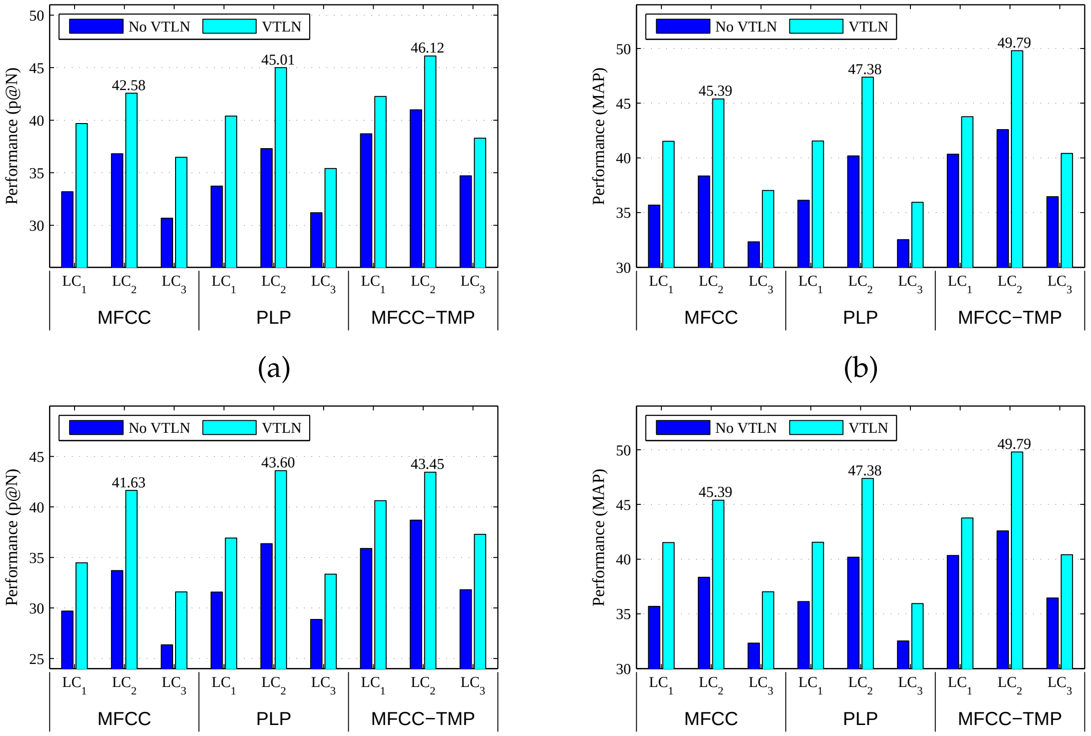
A.1.1.1 Effect of Local Constraints

Figure A.3 shows the performance of QbE-STD systems for different local constraints, namely, LC_1, LC_2 and LC_3. It can be observed from Table A.3 that LC_2 performs better than other local constraints (especially, relatively better jump in performance from no VTLN case to VTLN for LC_2), probably due to its ability to capture a wide range of features along test utterances. For each local constraint, it can be also observed that VTL-warped Gaussian posteriorgrams improve QbE-STD performance over Gaussian posteriorgrams.

A.1.1.2 Effect of Number of Iterations in Proposed Iterative Approach

Figure A.4 shows the performance with various iteration index used in VTLN warping factor estimation. It can be observed that performance improves as the iteration index increases. After a certain number of iterations, performance saturates that might be due to possible overfitting to training dataset.

A.1.2 Deterministic Annealing Expectation Maximization (DAEM)



(a) (b) (c) (d)

Figure A.3: Effect of local constraints on TIMIT QbE-STD. (a) p@N for Q-20 TIMIT, (b) MAP for Q-20 TIMIT, (c) p@N for Q-84 TIMIT, and (d) MAP for Q-84 TIMIT. After [1].

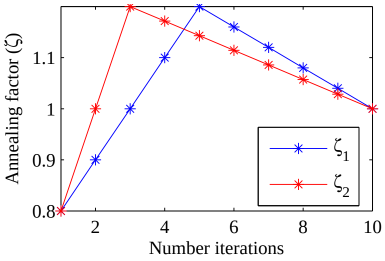
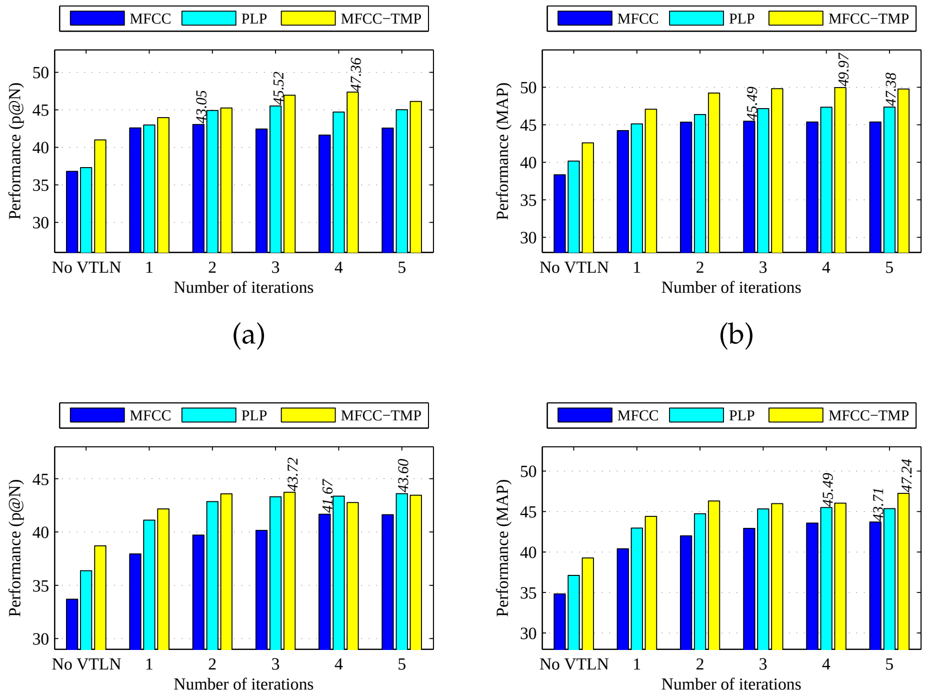


Figure A.5: Values of annealing factor (ζ) at every iterations.

We experimented with DAEM-based parameter estimation approach for TIMIT Q-84 database (as discussed in sub-Section 4.2.5.6). The values of ζ are varied as shown in Figure 4.13. It can be seen that performance of DAEM is comparable to the EM. This might be due to initial parameters that are set from vector quantization (which are the common for all two DAEM approaches, i.e., ζ_1 and ζ_2).

A.2 Mixture of GMMs

The results are reported in Table A.4 which clearly shows the significance of mixture of GMMs posteriorgram over the traditional GMM posteriorgram. It can be observed that the proposed approach of the mixture of GMM posteriorgram gave more MAP scores than the GMM posteriorgram (indicating that proposed mix-



(a) (b)

(c) (d)

Figure A.4: Effect of number of iterations in iterative VTLN warping factor estimation on TIMIT QbE-STD. (a) p@N for Q-20 TIMIT, (b) MAP for Q-20 TIMIT, (c) p@N for Q-84 TIMIT, and (d) MAP for Q-84 TIMIT. After [1].

Table A.3: Performance of DAEM on Q-84 TIMIT QbE-STD. After [1].

Feature sets	VTLN	EM		DAEM (ζ_1)		DAEM (ζ_2)	
		p@N	MAP	p@N	MAP	p@N	MAP
MFCC	×	33.70	34.84	33.72	35.04	33.72	35.07
	✓	37.95	40.40	37.77	40.41	37.84	40.39
PLP	×	36.36	37.12	35.95	37.06	36.08	37.01
	✓	41.11	**42.95**	40.91	**42.96**	40.96	**43.00**

(× = No VTLN, ✓ = VTLN)

ture of GMM approach has a good promise for the QbE-STD task). It is due to the restriction imposed by broad phoneme posterior probabilities.

In order to investigate the effect of proposed mixture of GMM approach w.r.t. various local constraints, we consider three local constraints (as shown in Figure 3.6). Figure A.6 shows the performance of mixture of GMMs vs. GMM posteriorgram for local constraints, LC_1, LC_2, and LC_3. It can be observed that local constraint LC_2 gave better search performance than the local constraints for most of the cases. The reasoning for this is given in sub-Section 3.6.

Table A.4: Performance of proposed mixture of GMM (mixGP) posteriorgram for TIMIT QbE-STD task in terms of (a) p@N, and (b) MAP (with local constraint LC_2 and 128 Gaussian components)

Feature Sets	GMM		Mixture of GMMs	
	p@N	MAP	p@N	MAP
MFCC	36.81	38.34	**41.34**	**44.29**
PLP	37.30	40.18	**43.14**	**45.14**
MFCC-TMP	40.99	42.59	**41.87**	**43.52**

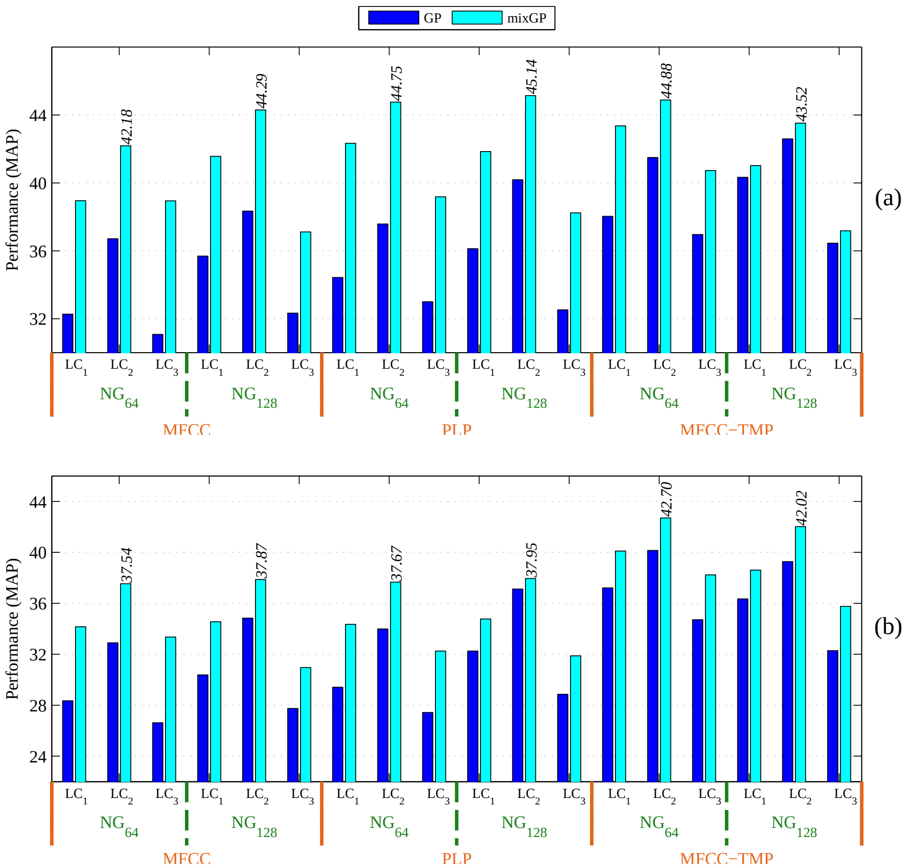


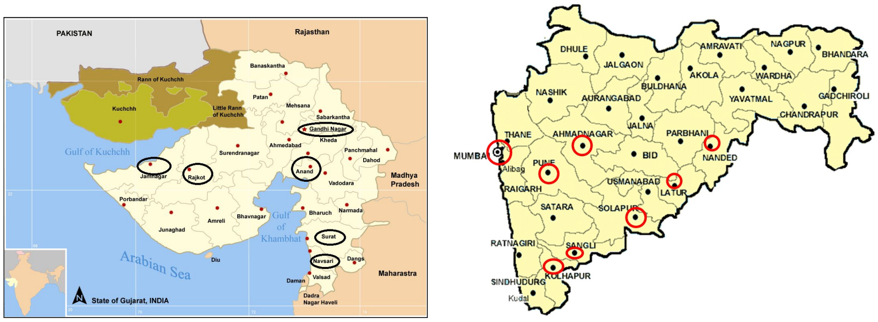
Figure A.6: Performance in MAP of mixture of GMM posteriorgram for (a) Q-20 TIMIT QbE-STD, and (b) Q-84 TIMIT QbE-STD (NG=Number of Gaussians).

Appendix B. Miscellaneous Studies

B.1 Prosodically Guided Phonetic Engine

This work is done under the DeitY sponsored consortium project at DA-IICT, in which author of this thesis was a project staff (during April 2012-June 2014). DA-IICT team has collected speech data and other relevant metadata in two Indian languages, namely, Gujarati and Marathi. These two languages are spoken mostly in two states of India, i.e., Gujarat and Maharashtra, respectively. The data is recorded in three different modes, namely, read, spontaneous and lecture modes. The data has been collected using portable handy recorder (Zoom H4n) as most of the data was recorded from remote villages and real field environments (i.e., real-life settings). The recording was performed at *44.1 kHz* sampling frequency with 16 bits/sample resolution. For the collection of Gujarati speech data, author of this thesis along with other team member visited several places of Gujarat state to collect speech data and other metadata. The places selected includes Gandhinagar (Vavol, Paliyad), Navsari (Moti kakrad, Navsari), Surat, Anand (Umreth), Jamnagar (Vijarkhi, Mota thavariya), Rajkot, Bhavnagar (Chamardi, Bhavnagar) and Kutch (Kera, Anjar). These places cover three dialectal regions of Gujarat state, namely, Saurashtra, South Gujarat and North East Gujarat. For the collection of Marathi speech data, other team member visited several places of Maharashtra state. The places for both the states are mainly, Ahmedanagar (Kakti), Nanded (Basmath), Latur, Solapur, Sangli (Vibhutvadi), Kolhapur (Ichalkaranji), Pune and Lonavala. The places are shown by a circle around the surrounding region as in Figure B.1. The places for data collection, experiences, observation and various statistics related to phonetic transcription are discussed in [194, 245, 246]. The followings are the observations found while transcribing the speech signal [246].

- Many times listener finds overlap across two phonetic symbols.
- Due to ambiguity between aspirated plosive and fricative sounds, transcriber often get confused [245].
- Human perception of phonetic symbols at different-level is different. It



(a) (b)

Figure B.1: Places of Gujarat and Maharashtra states. The circles indicate the dialectal regions, where data has been collected. After [247,248].

means that person may not recognize the same phonetic symbol at word or syllable-level than at sentence-level. Thus, there are variation in perception at various levels of speech sound units.

- Any two transcribers may not identify the exact the same phone and word boundaries. Since human perception of hearing is subjective.
- Diacritic marks are very error prone in terms of an agreement between two transcribers.
- In a lecture mode, speech subject tries to prolong the vowels in order to create interest among listeners (Here, children of primary school are the listeners mostly).
- In both the languages, diphthongs and associated vowels may be perceived as two distinct vowels. Hence, the transcriber may mark as two different syllables instead of a vowel.

The motivation behind the data collection is to capture the diversity in regional languages in terms of recording modes.

B.1.1 Phonetic Engine (PE)

The objective here is to develop the resources from the spoken data and use it for QbE-STD task. The manual transcription is performed on spoken data in Gujarati and Marathi languages. This transcription along with spoken data is used to build the phonetic engine (PE). The phonetic posteriorgram obtained by training is used as a representation of spoken documents and spoken query. As a part of DeitY consortium project at DA-IICT, we built our in-house speech recognizer in Gujarati and Marathi languages. The objective was to annotate the speech sounds into International Phonetic Alphabet (IPA) by listening the speech

Table B.1: Phoneme recognition (% Correct) performance of MFCC and PLP in classification of phonetic units for Gujarati and Marathi

Feature Sets	G-R	G-C	G-L	M-R	M-C	M-L
MFCC	67.11	62.37	59.84	59.19	49.81	39.64
PLP	66.89	62.75	60.18	60.36	48.82	41.76

G = Gujarati, M= Marathi, R = Read, L=Lecture, and C= Conversational

signal carefully. This is generally called as transcribing the speech signal. Since IPA symbols are close to the production of speech signal, the output is production units, which is called as phones (generally independent of languages of speech signal). We built recognizer based on HMM statistical model. Since the speech signal is represented in terms of phonetic symbols, the recognizer is referred to as Phonetic Engine (PE). However, training is similar to conventional phoneme recognizer [209]. Two different sets of feature vectors, namely, MFCC and PLP are used in the development of PE. Phonetic units that are manually transcribed by the transcribers are used to train HMMs for each phonetic unit using HTK (HMM Toolkit) [180]. Since the speech signal is not aligned w. r. t. every phonetic symbol, a flat start-based approach is employed. Five-state HMMs, which include two non-emitting and three emitting states with single Gaussian model per state are initialized. HMM embedded re-estimation is performed several times. Finally, the test data is decoded into a single phonetic string. No phone language model (LM) is used here, since it is expected that consecutive phone sequence might not capture effective information, which is derived from manual phonetic transcription. The similar design procedure is used to develop PEs for both the languages, namely, Gujarati and Marathi and for all the three recording modes (namely, read, conversational and lecture). PE is designed using the two kinds of feature sets, namely, MFCC and PLP and their performance in the three modes of speech (as shown in Table B.1). Performance is evaluated in terms of % accuracy and % correct detection [180]. From Table B.1, the performance for read speech is observed to be better (in both Gujarati and Marathi databases) as compared to the spontaneous speech and lecture speech. This may be due to the fact that read speech has least prosodic variations, whereas lecture speech has higher variations in intonation (and thus, speech prosody in general). In read speech, the speakers are constrained by the given fixed text material and hence, there are less prosodic variations, which is not the case in spontaneous and lecture speech.

Major misclassification happen with aspirated and non-aspirated forms of consonants [249]. Most of the aspirated consonants are observed to be misclassified to their non- aspirated versions. This might be because of the same manner and place of articulation. For example, most confusing aspirated consonants are [b] - [b^{h}], [ʧ] - [ʧ^{h}], [d] - [d^{h}], [g] - [g^{h}], [k] - [k^{h}], [p] - [p^{h}] and [t] - [t^{h}]. The basic difference between the aspirated and non-aspirated consonants is that in aspirated ones, an aspiration occurs simultaneously with the voicing. The reason for non-aspirated consonants being detected as aspirated ones might be the presence of some noise followed by the consonant that is being detected as *aspiration*. On the other hand, the aspiration part of aspirated consonants may be missed leading to misclassification as non-aspirated. In addition, as most of the Indian languages have phones followed by schwa (i.e., [ə]), this results in confusion for transcribers as to whether to put [ə] or not and human errors take place. It is observed from confusion matrix that schwa is confused with almost all the phonemes and there have been a large number of insertions and deletions. This type of misclassification can be reduced to a certain extent by improving and making precise transcriptions. For very small occurrences, silence is detected as plosive, such as, [ʈ], [t], [k], [p], etc. Presence of bursts might be detected due to the presence of unavoidable noise in the real-field environment. It is found that most of the times, vowel gets confused with vowels, such as, [ɑ] gets confused with [ə], [ɛ] and [o]; [ɛ] gets confused with [i] and vice-versa; [o] gets confused with [ə] and [u]. In addition, plosive consonants get confused with plosive consonants. For example, [t] gets confused with [ʈ], [d], [k] and [p]; [d] gets confused with [ɖ], [b] and [g]; [b] gets confused with [d] and [g]. This is because of the short durations of plosive, which are not easily captured even though derivative (i.e, Δ) and acceleration (i.e., $\Delta - \Delta$) coefficients are used to capture dynamics of vocal tract. Fricatives get confused with other fricatives, such as, [z] gets confused with [ʤ] and [s] gets confused with [z] and nasals get confused with other nasals, such as, [m] gets confused with [n]. Another observation is misclassification of fricative [s] as aspirated consonants like [ʧ^{h}] and [p^{h}]. Similar analysis is observed across different phonetic representations.

B.1.2 QbE-STD system

To apply this GMM-HMM recognizer for QbE-STD, we estimated the likelihood probability for each state associated with HMM and treated as a posterior probability of GMM (with uniform prior assumption) and then, we normalize the probabilities for each frame vector. To reduce the dimensionality, we further summed up all the state posterior probability into a single posterior probability for each phonetic unit. This results into *37*-dimensional posteriorgram vector.

Table B.2: Performance (MTWV and C_{nxe}^{min}) of Phonetic Engine (PE) for SWS 2013 QbE-STD Task

PE	Dev Set		Eval Set	
	MTWV	C_{nxe}^{min}	MTWV	C_{nxe}^{min}
G	0.0598	0.8792	0.0464	0.8902
M	0.0526	0.8977	0.0319	0.9089
G-M	0.0544	0.8822	0.0431	0.8934

G= Gujarati, M=Marathi, and G-M=Average posteriorgram of Gujarati and Marathi.

Table B.2 shows the performance of phonetic engine (PE) for QbE-STD task conducted on SWS 2013 data. The poor performance of QbE-STD might be because of an inability of Gujarati and Marathi PE to cover the phones of the languages that are preset in the SWS 2013 database (which is mostly African and European). This might need verification of transcription as manual transcription is error prone task and was performed by non-professional transcribers. QbE-STD performance of Gujarati PE is better than the Marathi PE. This is due to the performance (in terms of phone recognition) of Gujarati PE is better than the Marathi PE as discussed in Table B.1.

B.2 Effect of Isolated Query *vs.* Query in Carrier Phrase

The earlier studies have been subjective and tested the intelligibility of syllables and words in isolation and within the carrier sentences [250,251]. The objective is to investigate significance of embedding a query in a carrier phrase for automatic recognition of words via template matching using DTW algorithm. For this, a query word is matched with a reference word from a carrier phrase. The main difference between a query spoken in isolation and in carrier phrase is the transition of articulatory features, such as, tongue position, velum position, etc. For the query spoken in isolation, the articulatory features come into a particular position for the production of query from the rest position. On the other hand, when a query is spoken after another word, i.e., query is *embedded* in a carrier phrase, the articulators are already in motion. This induces constraints in the production of a word, which is not present when the articulators are at rest (as in the case of production of isolated query). Though the query in a carrier phrase has effects of coarticulation, the movement of articulators is constrained by the previously spoken word (due to local coarticulation) and many words in the future (due to global articulation) and variations are less. This leads to lesser variations in acoustic features of query in carrier phrase than in the isolation.

The effects of coarticulation occur mostly at the beginning and the end of the spoken words. These effects can be handled by removing certain frames of the reference and query digits from the beginning and the end. Since the durations of digits are not the same, frames are truncated w. r. t. % of the total number of frames. Various experiments were performed to find the optimal % number of frames to be truncated from the beginning and the end. In these experiments, isolated queries are taken. Table B.3 shows the performance of template matching with different % number of frames truncated from the beginning. It is observed that for *10* % truncation of the frames from the beginning provides better performance (in terms of % EER) than the case, where queries are taken from the carrier phrases.

Table B.3: Performance for isolated queries truncated by different % number of frames from the beginning (in % Precision and % EER).

Query Type	% Truncation	% Precision	% EER
Carrier	-	51.31	24.94
Isolation	-	45.40	29.40
Isolation	5	**52.88**	**23.39**
Isolation	10	52.74	23.22
Isolation	15	51.08	24.13
Isolation	20	48.05	25.86

B.3 Non-uniform Frequency Warping Approaches

We explore universal frequency warping in two spectral features, namely, Scale Transform Cepstral Coefficients (STCC) and Warped Linear Prediction Cepstral Coefficients (WLPCC) in order to develop speaker-invariant features for audio search task. STCCs compensate for the differences in Vocal Tract Length (VTL) using log-warping. WLP coefficients (WLPC) are easily obtained by Levinson-Durbin algorithm using warped autocorrelation function. Bark scale-warped LP spectrum is obtained by the WLPCs. Cepstral features are obtained by taking DCT of logarithm of the warped spectra [252]. The detailed procedure of STCC and WLPCC feature extraction is shown in Figure B.2.

The overall performance of the audio search system is shown in Table B.4. It can be observed that the VTLN-based feature sets, namely, STCC and WLPCC perform better than the MFCC alone. About *3* % absolute improvement can be

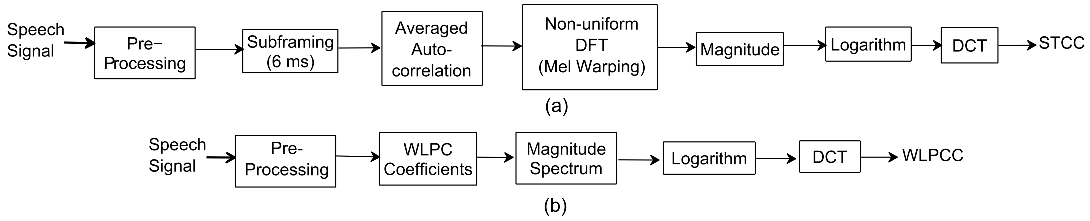


Figure B.2: Schematic block diagram for feature extraction of (a) STCC and (b) WLPCC. After [199,227].

Table B.4: Experimental results of non-uniform frequency warping features for TIMIT QbE-STD task. After [253]

Feature Sets	p@N	%EER	Feature Sets	p@N
MFCC	24.65	25.30	MFCC-fused	40.17
STCC	**27.98**	**23.73**	STCC-fused	**44.68**
WLPCC	**27.13**	**23.25**	WLPCC-fused	41.29

observed using STCC and WLPCC features. In Table B.4, the performance of each isolated query of each feature sets are mentioned as MFCC, STCC, and WLPCC, respectively. In addition, distortion score from the same query is fused. This will improve the statistical confidence about the query detection task. The fused features are called as MFCC-fused, STCC-fused and WLPCC-fused, respectively. From Table B.4, it can be observed that the fusing of multiple evidences indeed improves the audio search performance for all these three feature sets. The averaging of distortion score is used as fused score. 5^{th} column of Table B.4 shows the performance using fused score improves.

B.4 QbE-STD System for Gujarati Language

In this Section, we present the results of VTL-warped Gaussian posteriorgram and mixture of GMMs posteriorgram for our in-house database. This dataset contains 1400 test utterances (duration is about 3 hours). We have used 25 queries spoken by the two speakers (one male and one female) in the experimental setup. Table

Table B.5: The list of queries used in Gujarati QbE-STD

agaNAnevu	kudaratI	taMdurastI	paMchataMtra	mahadaaMshe
asafaLatA	gAjara	turiyA	paMchyAshI	ratALu
Adato	jamarukha	temaNe	paushhTIka	saMgharshha
OgaNapachAsa	TAmeTA	duraMdeshI	filosofI	sItAfaLa
kArelA	tarabucha	nAnapaNa	magafaLI	svAdIshhTa

Table B.6: Experimental results for Gujarati QbE-STD

Feature	p@N			MAP		
Sets	GP	VTL-warp	mix GMM	GP	VTL-warp	mix GMM
MFCC	16.71	22.10	**22.12**	14.93	**21.86**	19.76
PLP	17.70	23.36	**24.41**	17.34	24.89	**24.99**
MFCC-TMP	19.02	**24.19**	23.73	18.31	**23.36**	22.47

B.5 shows the list of 25 queries used in Gujarati QbE-STD. Table B.6 shows the results of Gujarati QbE-STD for ranked evaluation task. It can be shown from the Table B.6 that the performance of mixture of GMM and VTL-warped GP outperforms GP. The average p@N is increased by 5.4 % and 5.6 % for VTL-warped GP and mixture of GMMs posteriorgram, respectively. Thus, the results confirms VTL-warped GP and mixture of GMM posteriorgram are better audio representation than the GP for QbE-STD task.

Appendix C. Frequency Warping

There are many ways to incorporate warping in feature extraction. In VTLN problem formulation, the first problem is to define warping relation between source and target spectrum and then estimate the warping factor (i.e., amount of warping (compression or dilation)). It is very difficult to know the exact warping relation between both the spectra. Mostly linear or piecewise linear warping relation is considered. The primary motivation behind using piecewise linear relation is due to the fact that warping relation across different frequency bands is found to behave differently. Hence, it is better to approximate in terms of piecewise linear relationship. The second task is to estimate warping factor, which is estimated in the feature domain instead of frequency-domain. It means that warping relation across the frequency points is well exploited in terms features in mel-warped triangular filters. Hence, mel filterbank is modified in order to capture these warping relation into it.

C.1 Time-Resampling

The time scaling property suggest that the scaling in time, means changing sampling rate compresses or expands the spectrum [254]. In particular, for a continuous-time signal,

$$x(t) \underset{\longleftrightarrow}{F} X(f), \tag{C.1}$$

$$x(at) \underset{\longleftrightarrow}{F} \frac{1}{|a|} X\left(\frac{f}{a}\right). \tag{C.2}$$

Thus, resampled version of the signal can be used for different warped feature computation $\mathbf{x}_t^{\alpha}$.

C.2 Filterbank Modification

There can be possibilities to design filterbank, which are used in feature extraction [255]. This can be done by changing the frequency relationship by introducing warping factor into the conventional auditory frequency scales, namely, mel and

Bark scale. Bark scale is originally defined as [55]:

$$B(f) = 6\ln\left(\frac{f}{600} + \left((\frac{f}{600})^2 + 1\right)^{0.5}\right). \tag{C.3}$$

That can be modified in warped frequency-domain by considering new scale,

$$B_\alpha(f) = 6\ln\left(\frac{f}{\alpha 600} + \left((\frac{f}{\alpha 600})^2 + 1\right)^{0.5}\right). \tag{C.4}$$

Similarly, mel scale is originally defined as [256]:

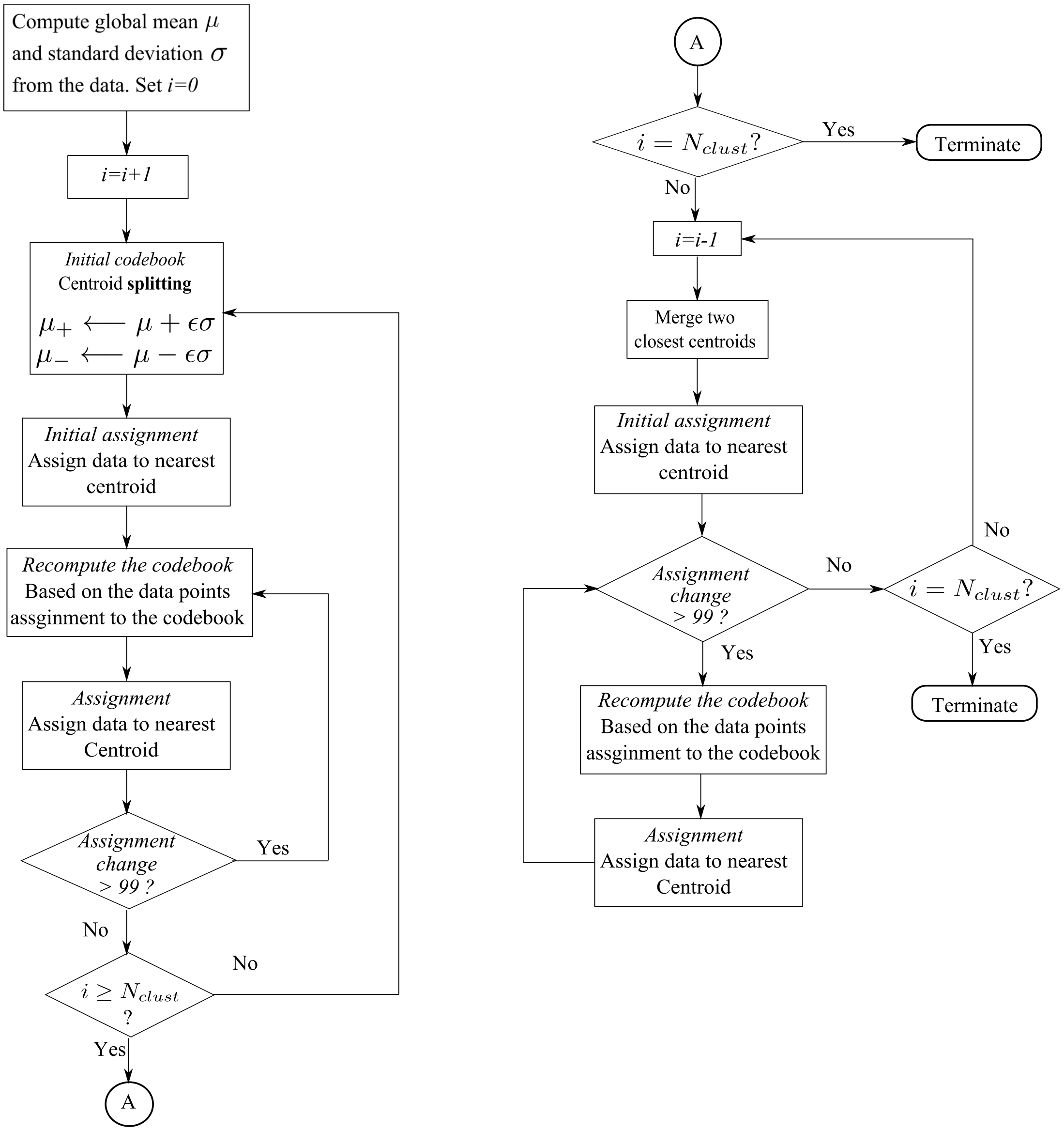
$$M(f) = 2595\log_{10}\left(1 + \frac{f}{700}\right). \tag{C.5}$$

That can be modified in warped frequency-domain by

$$M_\alpha(f) = 2595\log_{10}\left(1 + \frac{f}{700\alpha}\right). \tag{C.6}$$

Bandwidth Issue in Filterbank design: For lower warping factor, subband filters are stretched away. In this case, the last filter crosses the bandwidth or Nyquist frequency, which leads to the aliasing effect due to violation of condition given by Shanon sampling theorem. Hence, a cutoff frequency is introduced to overcome such cases, where these exceeding frequency components are mapped to the Nyquist frequency. In this thesis work, we set higher cutoff frequency as 3500 Hz, and 7000 Hz corresponding to sampling frequencies of 4000 Hz, and 8000 Hz, respectively.

Appendix D. Flowchart of Vector Quantization



Compute global mean μ and standard deviation σ from the data. Set *i=0*
i=i+1
Initial codebook
Centroid **splitting**
$\mu_+ \longleftarrow \mu + \epsilon\sigma$
$\mu_- \longleftarrow \mu - \epsilon\sigma$
Initial assignment
Assign data to nearest centroid
Recompute the codebook
Based on the data points assginment to the codebook
Assignment
Assign data to nearest Centroid
Assignment change > 99 ?
Yes
No
$i \geq N_{clust}$?
No
Yes
A
A
$i = N_{clust}$?
Yes
Terminate
No
i=i-1
Merge two closest centroids
Initial assignment
Assign data to nearest centroid
Assignment change > 99 ?
No
$i = N_{clust}$?
No
Yes
Terminate
Yes
Recompute the codebook
Based on the data points assginment to the codebook
Assignment
Assign data to nearest Centroid

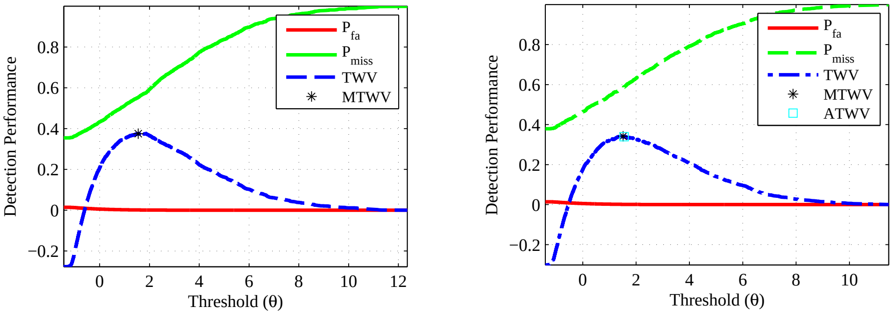
Figure D.1: Flowchart of K-means vector quantization (VQ) [184]. N_{clust} is the number of desired clusters (codebook vectors). $\epsilon = 0.2$ constant and mean μ and standard deviation σ is per dimension with data uncorrelated assumption.

Appendix E. Selection of Operating Threshold θ in TWV

The objective of QbE-STD is to maximize Term Weighted Value (TWV). Since $P_{miss}(q, \theta)$ and $P_{fa}(q, \theta)$, the selection of θ is made by maximizing TWV on the Dev set. As θ increases, $P_{miss}(q, \theta)$ increases, whereas $P_{fa}(q, \theta)$ decreases. We select θ that maximizes TWV, i.e., $TWV(\theta) = MTWV$. This threshold is optimal threshold, θ_{opt}. Now, threshold θ_{opt} is used to evaluate the performance of evaluation set and TWV at θ_{opt} is called as Actual TWV (ATWV). It is expected that ATWV should be very much close to the Maximum TWV (MTWV) indicating the well optimized QbE-STD. Table E.1 shows the ATWV and MTWV for CZ posteriorgram and θ_{opt} at which TWV is maximum. From Dev set, optimal threshold θ_{opt} is obtained, which is *1.552*. The TWV performance of Eval set at $\theta = 1.552$, i.e., ATWV is *0.339*. However, the MTWV is *0.342*, which is obtained at $\theta = 1.512$. The TWV and threshold for Dev and Eval sets are plotted in Figure E.1.

Table E.1: TWV values for Dev and Eval sets

Query set	MTWV	ATWV	θ_{opt}
Eval	0.375	0.375	1.552
Dev	0.342	0.339	1.512



(a)

(b)

Figure E.1: The selection of optimal threshold θ optimizing TWV: Detection performance for (a) Dev set, and (b) Eval set.

Appendix F. Role of β in TWV

Consider the following notations for defining the Term Weighted Value (TWV):
q = query, t = test utterance, θ = threshold,
$N_{act}(q)$ = number of actual occurrences for a query q,
$N_{nt}(q)$ = number of non-targets (non-occurrences)for a query q,
$N_{miss}(q,\theta)$ = number of misses at a given threshold θ for a query q,
$N_{fa}(q,\theta)$ = number of false alarms (false acceptances) at given threshold θ for a query q,
$P_{fa} =$ probability of false alarm (false acceptance),
$P_{miss} =$ probability of miss detection,
$n_{tps} =$ number of targets per second, which is taken as 1,
$T_{audio} =$ total duration of audio documents in seconds,
C_{Det} = Detection Cost Function (DCF),
$C_{Default}$ = Default trivial value of DCF,
$C_{Norm} =$ Normalized value of DCF,
$C_{miss} =$ Cost associated with miss detection,
$C_{fa} =$ Cost associated with false acceptance (false alarm),
$P_{target} =$ prior probability of target trial.

The number of non-occurrences, $N_{nt}(q)$, are spread across the speech and hence, it is not explicitly defined. The total number of virtual targets are $N = n_{tps}T_{audio}$ and hence, $N_{nt}(q) = N - N_{act}(q)$.

$P_{fa} = \frac{N_{fa}(q,\theta)}{N_{nt}(q)}, \qquad P_{miss} = \frac{N_{miss}(q,\theta)}{N_{act}(q)}.$

DCF value, i.e., C_{Det} as per NIST 2010 Speaker Recognition Evaluation [257]:

$$C_{Det} = C_{miss}P_{miss}P_{target} + C_{fa}P_{fa}(1 - P_{target}). \tag{F.1}$$

Consider the trivial cases of accepting all the hypothesis or rejecting all the hypothesis:

$$C_{Default} = \min \begin{cases} C_{miss}P_{target}, \\ C_{fa}(1 - P_{target}). \end{cases}$$

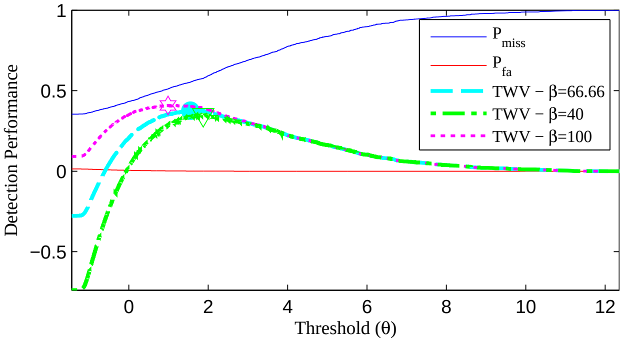


Figure F.1: Impact on TWV for different β.

For trivial system that reject all the hypothesis, i.e., $C_{Default} = C_{miss}P_{target}$

$$C_{Norm} = \frac{C_{Det}}{C_{Default}}. \tag{F.2}$$

Using Eq.(F.1),

$$C_{Norm} = P_{miss} + \frac{C_{fa}(1 - P_{target})}{C_{miss}P_{target}}P_{fa}. \tag{F.3}$$

Intuitively, TWV is given by $TWV = 1 - C_{Norm}$. The parameter β is defined as :

$$\beta = \frac{C_{fa}(1 - P_{target})}{C_{miss}P_{target}}. \tag{F.4}$$

For MediaEval 2013 SWS, $P_{target} = 0.00015, C_{fa} = 1, C_{miss} = 100$, so $\beta = 66.66$. TWV for query q and threshold θ is:

$$TWV(\theta) = 1 - \frac{1}{|Q|}\sum_{q}\left(P_{miss}(q,\theta) + \beta P_{fa}(q,\theta)\right). \tag{F.5}$$

As per eq. (F.5), TWV is function of $(P_{miss}(q,\theta)$ and $P_{fa}(q,\theta)$, where β is kept constant. β and θ are independent. We conducted QbE-STD task on SWS 2013 Dev set using on CZ phonetic posteriorgram (BUT phoneme recognizer in Czech language). The plot of P_{fa} and P_{miss} w.r.t. threshold (θ) is shown in Figure F.1. It can be seen from Figure F.1 that P_{miss} increases as θ increases, P_{fa} increases as θ increases. P_{miss} and P_{fa} are dependent on threshold θ. As indicated by Fig. F.1 that the value of P_{fa} is much lower, so β emphasizes the P_{fa}. In order to compute TWV, we considered different β in eq. (F.5), i.e., β = 40, 66.66 and 100. The Maximum TWV (MTWV) are different for different values of β. At the same time, the optimal value of P_{miss} and P_{fa} are different.

References

[1] M. C. Madhavi and H. A. Patil, "VTLN-warped Gaussian posteriorgram for QbE-STD," in *EUSIPCO*, Kos island, Greece, 2017, pp. 563–567.

[2] M. C. Madhavi and H. A. Patil, "Partial matching and search space reduction for QbE-STD," *Computer Speech & Language*, vol. 45, pp. 58 – 82, September 2017.

[3] R. Rosenfeld, D. R. Olsen, and A. I. Rudnicky, "Universal speech interfaces," *Interactions*, vol. 8, no. 6, pp. 34–44, 2001.

[4] F. Weng, P. Angkititrakul, E. Shriberg, L. P. Heck, S. Peters, and J. H. L. Hansen, "Conversational in-vehicle dialog systems: The past, present, and future," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 49–60, 2016.

[5] J. Hauswald, M. A. Laurenzano, Y. Zhang, C. Li, A. Rovinski, A. Khurana, R. G. Dreslinski, T. N. Mudge, V. Petrucci, L. Tang, and J. Mars, "Sirius: An open end-to-end voice and vision personal assistant and its implications for future warehouse scale computers," in *Proc. of the* 20^{th} *Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS'15)*, Istanbul, Turkey, 2015, pp. 223–238.

[6] Y. Ju and T. Paek, "A voice search approach to replying to SMS messages in automobiles," in *Proc. INTERSPEECH*, Brighton, UK, 2009, pp. 987–990.

[7] L. Lee, J. R. Glass, H. Lee, and C. Chan, "Spoken content retrieval-beyond cascading speech recognition with text retrieval," *IEEE/ACM Trans. on Audio, Speech, and Language Process.*, vol. 23, no. 9, pp. 1389–1420, 2015.

[8] M. Larson and G. J. Jones, "Spoken content retrieval: A survey of techniques and technologies," *Foundations and Trends in Information Retrieval*, vol. 5, no. 3, pp. 235–422, 2012.

[9] C. Yonekura, Y. Furuya, S. Natori, H. Nishizaki, and Y. Sekiguchi, "Evaluation of the usefulness of spoken term detection in an electronic note-taking support system," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, (APSIPA)*, Kaohsiung, Taiwan, 2013, pp. 1–4.

[10] T. J. Hazen, B. Sherry, and M. Adler, "Speech-based annotation and retrieval of digital photographs," in *Proc. INTERSPEECH*, Antwerp, Belgium, 2007, pp. 2165–2168.

[11] X. Anguera, J. Xu, and N. Oliver, "Multimodal photo annotation and retrieval on a mobile phone," in *Proc. of the* 1^{st} *ACM SIGMM Int. Conf. on Multimedia Information Retrieval, MIR*, Vancouver, British Columbia, Canada, 2008, pp. 188–194.

[12] NIST, "The Spoken Term Detection (STD) 2006 Evaluation Plan," http://www.itl.nist.gov/iad/mig/tests/std/2006/docs/std06-evalplan-v10.pdf, 2006, {Last Accessed on 22 September, 2016}.

[13] L. R. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, 1st ed. Prentice Hall, 1993.

[14] I. Szöke, "Hybrid word-subword spoken term detection," Ph.D. Thesis, Faculty of Info. Tech., Dept. of Computer Graphics and Multimedia, Brno University of Technology (BUT), Czech Republic, 2010.

[15] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proc. ACM SIGIR Searching Spontaneous Conversational Speech*, vol. 7, Amsterdam, The Netherlands, 2007, pp. 51–57.

[16] C. Chelba, T. Hazen, and M. Saraclar, "Retrieval and browsing of spoken content," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 39–49, May 2008.

[17] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 19, no. 8, pp. 2338–2347, 2011.

[18] R. Wallace, R. Vogt, and S. Sridharan, "A phonetic search approach to the 2006 NIST spoken term detection evaluation," in *Proc. INTERSPEECH*, Antwerp, Belgium, 2007, pp. 2385–2388.

[19] J. P. Pinto, I. Szoke, S. R. M. Prasanna, and H. Hermansky, "Fast Approximate Spoken Term Detection from Sequence of Phonemes," IDIAP, Tech. Rep. Idiap-RR-45-2008, 0 2008.

[20] K. Katsurada, S. Sawada, S. Teshima, Y. Iribe, and T. Nitta, "Evaluation of fast spoken term detection using a suffix array," in *Proc. INTERSPEECH*, Florence, Italy, 2011, pp. 909–912.

[21] T. Kaneko and T. Akiba, "Metric subspace indexing for fast spoken term detection," in *Proc. INTERSPEECH*, Makuhari, Chiba, Japan, 2010, pp. 689–692.

[22] A. Mandal, K. R. P. Kumar, and P. Mitra, "Recent developments in spoken term detection: A survey," *Int. Journal of Speech Tech. (IJST)*, vol. 17, no. 2, pp. 183–198, 2014.

[23] C. Parada, A. Sethy, and B. Ramabhadran, "Query-by-example spoken term detection for OOV terms," in *IEEE Workshop on Automatic Speech Recognition & Understanding, (ASRU)*, Merano/Meran, Italy, 2009, pp. 404–409.

[24] F. Metze, X. Anguera, E. Barnard, M. Davel, and G. Gravier, "Language independent search in MediaEval's Spoken Web Search task," *Computer Speech and Language*, vol. 28, no. 5, pp. 1066–1082, 2014.

[25] C. Chan, "Unsupervised spoken term detection with spoken queries," Ph.D. Dissertation, Comm. Eng. Dept., National Taiwan University, Taipei, Taiwan, 2012.

[26] F. Metze, N. Rajput, X. Anguera, M. Davel, G. Gravier, C. van Heerden, G. Mantena, A. Muscariello, K. Prahallad, I. Szoke, and J. Tejedor, "The spoken web search task at MediaEval 2011," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Kyoto, Japan, 2012, pp. 5165–5168.

[27] N. Rajput and F. Metze, "Spoken Web Search," in *Working Notes Proc. of the MediaEval 2011 Workshop*, Pisa, Italy, 2011.

[28] X. Anguera, F. Metze, A. Buzo, I. Szoke, and L. J. Rodriguez-Fuentes, "The spoken web search task," in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[29] X. Anguera, L. J. Rodriguez-Fuentes, I. Szöke, A. Buzo, and F. Metze, "Query by example search on speech at MediaEval 2014," in *Working Notes Proc. of the MediaEval 2014 Workshop*, vol. 1263, Barcelona, Spain, October 16-17 2014.

[30] I. Szöke, L. J. Rodríguez-Fuentes, A. Buzo, X. Anguera, F. Metze, J. Proença, M. Lojka, and X. Xiong, "Query by example search on speech at MediaEval 2015," in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[31] 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 *Proc. 4th Int. Workshop on Spoken Language Tech. for Under-Resourced Languages*, St. Petersburg, Russia, May 2014, pp. 24–31.

[32] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Glasgow, UK, 1989, pp. 627–630 vol.1.

[33] I. Szöke, P. Schwarz, P. Matějka, L. Burget, M. Karafiát, and J. Černocký, "Phoneme based acoustics keyword spotting in informal continuous speech," in 8th *Int. Conf. on Text, Speech and Dialogue (TSD)*, V. Matoušek, P. Mautner, and T. Pavelka, Eds. Karlovy Vary, Czech Republic: LNAI, Springer, 2005, pp. 302–309.

[34] S. R. Madikeri and H. A. Murthy, "Acoustic segmentation using group delay functions and its relevance to spoken keyword spotting," in *Text, Speech and Dialogue: 15th International Conference, TSD 2012, Brno, Czech Republic, September 3-7, 2012. Proceedings*, P. Sojka, A. Horák, I. Kopeček, and K. Pala, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 496–504.

[35] S. Salvador and P. Chan, "FastDTW: Toward accurate dynamic time warping in linear time and space," *Intell. Data Anal.*, vol. 11, no. 5, pp. 561–580, 2007.

[36] M. C. Madhavi and H. A. Patil, "Two stage zero-resource approaches for QbE-STD," in *(accepted)* 9th *Int. Conf. on Advances in Pattern Recognition, ICAPR 2017*, Bangalore, India.

[37] M. C. Madhavi and H. A. Patil, "Combining evidences from detection sources for QbE-STD," December 12-15 2017, (accepted) *Asia-Pacific Signal and Information Processing Association Annual Summit and Conf., APSIPA*, Kuala Lumpur, Malaysia.

[38] W. Shen, C. M. White, and T. J. Hazen, "A comparison of query-by-example methods for spoken term detection," in *Proc. INTERSPEECH*, Bringhton, UK, 2009, pp. 2143–2146.

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

[40] K. Vu, K. A. Hua, and N. Jiang, "Improving image retrieval effectiveness in query-by-example environment," in *Proc. ACM Symposium on Applied Computing (SAC)*, Melbourne, FL, USA, 2003, pp. 774–781.

[41] W. Tsai and H. Wang, "A query-by-example framework to retrieve music documents by singer," in *Int. Conf. on Multimedia and Expo, (ICME)*, Taipei, Taiwan, 2004, pp. 1863–1866.

[42] Y. Gao, S. S. Vedula, G. I. Lee, M. R. Lee, S. Khudanpur, and G. D. Hager, "Query-by-example surgical activity detection," *Int. J. of Computer Assisted Radiology and Surgery*, vol. 11, no. 6, pp. 987–996, 2016.

[43] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press (Online), 2009, http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf {Last Accessed on Dec. 29, 2016}.

[44] L. J. Rodriguez Fuentes and M. Penagarikano, "MediaEval 2013 spoken web search task: System performance measures," Department of Electricity and Electronics, University of the Basque Country, Tech. Rep. TR-2013-1, 2013.

[45] M. Skácel, "Query-by-example keyword spotting," M.S. Thesis, Faculty of Info. Tech., Dept. of Computer Graphics and Multimedia, Brno University of Technology (BUT), Czech Republic, 2015.

[46] M. Fapso, "Query-by-example spoken term detection," Ph.D. Thesis, Faculty of Info. Tech., Dept. of Computer Graphics and Multimedia, Brno University of Technology (BUT), Czech Republic, 2014.

[47] X. Anguera, L. J. Rodríguez-Fuentes, A. Buzo, F. Metze, I. Szöke, and M. Peñagarikano, "QUESST2014: Evaluating Query-by-Example Speech Search in a zero-resource setting with real-life queries," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, South Brisbane, Queensland, Australia, 2015, pp. 5833–5837.

[48] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The DET curve in assessment of detection task performance," in *Proc. EUROSPEECH*, Rhodes, Greece, 1997, pp. 1895–1898.

[49] J. Picone, "Fundamentals of Speech Recognition: A Short Course," *Inst. for Signal and Info. Process., Mississippi State University, USA*, May 15-17 1996.

[50] J. Makhoul, "Linear prediction: A tutorial review," *Proc. of the IEEE*, vol. 63, no. 4, pp. 561–580, April 1975.

[51] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. on Acoustics, Speech and Signal Proces.*, vol. 28, no. 4, pp. 357–366, 1980.

[52] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Proc. INTERSPEECH*, Florence, Italy, 2011, pp. 821–824.

[53] G. Mantena, S. Achanta, and K. Prahallad, "Query-by-example spoken term detection using frequency domain linear prediction and non-segmental dynamic time warping," *IEEE/ACM Trans. on Audio, Speech, and Language Process.*, vol. 22, no. 5, pp. 946–955, 2014.

[54] H. Wakita, “Normalization of vowels by vocal-tract length and its application to vowel identification,” *IEEE Trans. on Acoust., Speech and Signal Process.*, vol. 25, no. 2, pp. 183–192, 1977.

[55] H. Hermansky, “Perceptual linear predictive (PLP) analysis of speech,” *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[56] F. Hönig, G. Stemmer, C. Hacker, and F. Brugnara, “Revising perceptual linear prediction (PLP),” in *INTERSPEECH 2005 - EUROSPEECH, 9th European Conf. on Speech Comm. and Tech.*, Lisbon, Portugal, 2005, pp. 2997–3000.

[57] M. Athineos and D. P. W. Ellis, “Frequency-domain linear prediction for temporal features,” in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, St. Thomas, Virgin Islands, 2003, pp. 261–266.

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

[59] J. Tejedor, I. Szöke, and M. Fapso, “Novel methods for query selection and query combination in query-by-example spoken term detection,” in *Proceedings of the 2010 International Workshop on Searching Spontaneous Conversational Speech*, ser. SSCS ’10, New York, NY, USA, 2010, pp. 15–20.

[60] H. Xu, J. Hou, X. Xiao, V. T. Pham, C. Leung, L. Wang, V. H. Do, H. Lv, L. Xie, B. Ma, E. S. Chng, and H. Li, “Approximate search of audio queries by using DTW with phone time boundary and data augmentation,” in *Int. Conf. on Acoustics, Speech and Signal Processing, (ICASSP)*, Shanghai, China, 2016, pp. 6030–6034.

[61] P. Schwarz, “Phoneme recognition based on long temporal context,” Ph.D. Dissertation, Faculty of Info. Tech., Dept. of Computer Graphics and Multimedia, Brno University of Technology (BUT), Czech Republic, 2008.

[62] A. Abad, L. J. Rodríguez-Fuentes, M. Peñagarikano, A. Varona, and G. Bordel, “On the calibration and fusion of heterogeneous spoken term detection systems,” in *Proc. INTERSPEECH*, Lyon, France, 2013, pp. 20–24.

[63] J. Vavrek, M. Pleva, and J. Juhár, “TUKE MediaEval 2012: Spoken web search using DTW and unsupervised SVM,” in *Working Notes Proc. of the MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[64] A. Buzo, H. Cucu, and C. Burileanu, “SpeeD @ MediaEval 2014: Spoken term detection with robust multilingual phone recognition,” in *Working Notes Proc. of the MediaEval 2014 Workshop*, vol. 1263, Barcelona, Spain, October 16-17 2014.

[65] N. Lazic and P. Aarabi, “Spoken term detection using visual spectrogram matching,” in *IEEE Int. Symp. on Multimedia (ISM)*, Berkeley, California, USA, 2008, pp. 637–642.

[66] P. Lopez-Otero, L. Docio-Fernandez, and C. Garcia-Mateo, “Finding relevant features for zero-resource query-by-example search on speech,” *Speech Communication, Elsevier*, vol. 84, pp. 24–35, 2016.

[67] P. Lopez-Otero, L. D. Fernández, and C. García-Mateo, "Phonetic unit selection for cross-lingual query-by-example spoken term detection," in *IEEE Workshop on Automatic Speech Recognition and Understanding, (ASRU)*, Scottsdale, AZ, USA, 2015, pp. 223–229.

[68] X. Anguera, "Telefonica system for the spoken web search task at MediaEval 2011," in *Working Notes Proc. of the MediaEval 2011 Workshop*, Pisa, Italy, 2011.

[69] G. V. Mantena, B. Bollepalli, and K. Prahallad, "SWS task: Articulatory phonetic units and sliding DTW," in *Working Notes Proc. of the MediaEval 2011 Workshop*, Pisa, Italy, 2011.

[70] J. Vavrek, M. Pleva, M. Lojka, P. Viszlay, E. Kiktová, D. Hládek, J. Juhár, M. Pleva, E. Kiktova, D. Hladek *et al.*, "TUKE at MediaEval 2013 spoken web search task," in *MediaEval*, Barcelona, Spain, 2013.

[71] J. Vavrek, P. Viszlay, E. Kiktová, M. Lojka, J. Juhár, and A. Cizmar, "Query-by-example retrieval via fast sequential dynamic time warping algorithm," in *Proc. Int. Conf. on Telecommunications and Signal Processing, (TSP)*, Prague, Czech Republic, 2015, pp. 1–5.

[72] J. Vavrek, J. Juhár, and A. Cizmar, "Audio classification utilizing a rule-based approach and the support vector machine classifier," in 36^{th} *Int. Conf. on Telecommunications and Signal Process. (TSP)*, Rome, Italy, July 2013, pp. 512–516.

[73] J. A. Gómez, L. F. Hurtado, M. Calvo, and E. Sanchis, "ELiRF at MediaEval 2013: Spoken web search task," in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[74] 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 *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Florence, Italy, 4-9 May 2014, pp. 7819–7823.

[75] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, "GTTS-EHU systems for QUESST at MediaEval 2014," in *Working Notes Proc. of the MediaEval 2014 Workshop*, vol. 1263, Barcelona, Spain, October 16-17 2014.

[76] L. J. Rodríguez-Fuentes, A. Varona, M. Peñagarikano, G. Bordel, and M. Díez, "GTTS systems for the SWS task at MediaEval 2013," in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[77] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, Merano, Italy, 2009, pp. 398–403.

[78] H. Wang, C.-C. Leung, T. Lee, B. Ma, H. Li, H. Wang, C. Leung, T. Lee, B. Ma, and H. Li, "An acoustic segment modeling approach to query-by-example spoken term detection," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Kyoto, Japan, 2012, pp. 5157–5160.

[79] P. R. Reddy, S. Nayak, and K. S. R. Murty, "Unsupervised spoken word retrieval using Gaussian-Bernoulli restricted Boltzmann machines," in *Proc. INTERSPEECH*, Singapore, 2014, pp. 1737–1741.

[80] Y. Zhang, R. Salakhutdinov, H. A. Chang, and J. Glass, "Resource configurable spoken query detection using deep Boltzmann machines," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Kyoto, Japan, 2012, pp. 5161–5164.

[81] G. Aradilla, J. Vepa, and H. Bourlard, "Using posterior-based features in template matching for speech recognition," in *Proc. INTERSPEECH-ICSLP*, Pittsburgh, PA, USA, 2006, pp. 2570–2573.

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

[83] J. Proença, A. Veiga, and F. Perdigão, "Query by example search with segmented dynamic time warping for non-exact spoken queries," in *23rd European Signal Process. Conf., EUSIPCO*, Nice, France, 2015, pp. 1661–1665.

[84] M. C. Madhavi and H. A. Patil, "Modification in sequential dynamic time warping for fast computation of query-by-example spoken term detection task," in *Proc. Int. Conf. on Signal Processing and Comm., (SPCOM)*, IISc, Bangalore, India, June 12-15 2016, pp. 1–6.

[85] G. Mantena and K. Prahallad, "Use of articulatory bottle-neck features for query-by-example spoken term detection in low resource scenarios," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 7128–7132.

[86] A. Popli and A. Kumar, "Query-by-example spoken term detection using low dimensional posteriorgrams motivated by articulatory classes," in *Prof. Int. Workshop on Multimedia Signal Process., (MMSP)*, Xiamen, China, 2015, pp. 1–6.

[87] X. Liu, W. Guo, and N. Wang, "Query-by-example spoken term detection using bottleneck feature and hidden markov model," in *Int. Conf. on Fuzzy Systems and Knowledge Discovery, FSKD 2015*, Zhangjiajie, China, 2015, pp. 1319–1323.

[88] A. Saxena and B. Yegnanarayana, "Distinctive feature based representation of speech for query-by-example spoken term detection," in *Proc. INTERSPEECH*, Dresden, Germany, 2015, pp. 3680–3684.

[89] A. Juneja and C. Espy-Wilson, "Speech segmentation using probabilistic phonetic feature hierarchy and support vector machines," in *Proc. of the Int. Joint Conf. on Neural Networks (IJCNN)*, vol. 1, Portland, Oregan, 2003, pp. 675–679.

[90] C. Leung, L. Wang, H. Xu, J. Hou, V. T. Pham, H. Lv, L. Xie, X. Xiao, C. Ni, B. Ma, E. S. Chng, and H. Li, "Toward high-performance language-independent query-by-example spoken term detection for MediaEval 2015: Post-evaluation analysis," in *Proc. INTERSPEECH*, San Francisco, USA, 2016, pp. 3703–3707.

[91] M. Versteegh, R. Thiollière, T. Schatz, X. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015," in *Proc. INTERSPEECH*, Dresden, Germany, 2015, pp. 3169–3173.

[92] A. Jansen, E. Dupoux, S. Goldwater, M. Johnson, S. Khudanpur, K. Church, N. Feldman, H. Hermansky, F. Metze, R. C. Rose, M. Seltzer, P. Clark, I. McGraw, B. Varadarajan, E. Bennett, B. Börschinger, J. Chiu, E. Dunbar, A. Fourtassi, D. Harwath, C. Lee, K. Levin, A. Norouzian, V. Peddinti, R. Richardson, T. Schatz, and S. Thomas, "A summary of the 2012 JHU CLSP workshop on zero resource

speech technologies and models of early language acquisition," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Vancouver, BC, Canada, 2013, pp. 8111–8115.

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

[94] C. Chung, C. Chan, and L. Lee, "Unsupervised discovery of linguistic structure including two-level acoustic patterns using three cascaded stages of iterative optimization," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Vancouver, BC, Canada, 2013, pp. 8081–8085.

[95] H. Wang, T. Lee, and C. Leung, "Unsupervised spoken term detection with acoustic segment model," in *Proc. Oriental COCOSDA*, Hsinchu City, Taiwan, 2011, pp. 106–111.

[96] H. Wang, T. Lee, C. Leung, B. Ma, and H. Li, "Acoustic segment modeling with spectral clustering methods," *IEEE/ACM Trans. on Audio, Speech, and Language Process.*, vol. 23, no. 2, pp. 264–277, 2015.

[97] A. Saxena, "Significance of knowledge-based representation of speech for spoken term detection," M.S. Thesis, Int. Inst. of Info. Technology (IIIT-H), Electronics and Communications Engineering, Hyderabad, India, 2015.

[98] P. R. Reddy, K. Rout, and K. S. R. Murty, "Query word retrieval from continuous speech using GMM posteriorgrams," in *Proc. Int. Conf. on Signal Process. and Comm. (SPCOM)*, IISc, Bangalore, India, 2014, pp. 1–6.

[99] X. Anguera, "Speaker independent discriminant feature extraction for acoustic pattern-matching," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Kyoto, Japan, 2012, pp. 485–488.

[100] D. Vasudev, S. V. Gangashetty, K. K. A. Babu, and K. S. Riyas, "Query-by-example spoken term detection using bessel features," in *2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, Kerala, India, 2015, pp. 1–4.

[101] C. Gracia, X. Anguera, and X. Binefa, "Combining temporal and spectral information for query-by-example spoken term detection," in *Proc. European Signal Processing Conference, EUSIPCO*, Lisbon, Portugal, 2014, pp. 1487–1491.

[102] C. Gracia, X. Anguera, and X. Binefa, "The CMTECH spoken web search system for MediaEval 2013," in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[103] P. Yang, C. Leung, L. Xie, B. Ma, and H. Li, "Intrinsic spectral analysis based on temporal context features for query-by-example spoken term detection," in *Proc. INTERSPEECH*, Singapore, 2014, pp. 1722–1726.

[104] A. Norouzian, R. C. Rose, and A. Jansen, "Semi-supervised manifold learning approaches for spoken term verification," in *Proc. INTERSPEECH*, Lyon, France, 2013, pp. 2594–2598.

[105] C. Chan, C. Chung, Y. Kuo, and L. Lee, "Toward unsupervised model-based spoken term detection with spoken queries without annotated data," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Vancouver, BC, Canada, 2013, pp. 8550–8554.

[106] Y. Qiao, N. Shimomura, and N. Minematsu, "Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Las Vegas,USA, 2008, pp. 3989–3992.

[107] C. Chung, W. Hsu, C. Lee, and L. Lee, "Enhancing automatically discovered multi-level acoustic patterns considering context consistency with applications in spoken term detection," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, South Brisbane, Queensland, Australia, 2015, pp. 5231–5235.

[108] C. Chung, C. Chan, and L. Lee, "Unsupervised spoken term detection with spoken queries by multi-level acoustic patterns with varying model granularity," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 7814–7818.

[109] Y. Zhang, "Unsupervised speech processing with applications to query-by-example spoken term detection," Ph.D. Dissertation, Dept. of Elect. Eng. and Computer Science, Massachusetts Institute of Technology (MIT), USA, 2013.

[110] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "An auto-encoder based approach to unsupervised learning of subword units," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 7634–7638.

[111] H. Chen, C. Leung, L. Xie, B. Ma, and H. Li, "Unsupervised bottleneck features for low-resource query-by-example spoken term detection," in *Proc. INTERSPEECH*, San Francisco, USA, 2016, pp. 923–927.

[112] E. Barnard, M. Davel, C. van Heerden, N. Kleynhans, and K. Bali, "Phone recognition for spoken web search," in *Working Notes Proc. of the MediaEval 2011 Workshop*, Pisa, Italy, 2011.

[113] I. Szöke, L. Burget, F. Grézl, and L. Ondel, "BUT SWS 2013-massive parallel approach," in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[114] A. Buzo, H. Cucu, M. Safta, and C. Burileanu, "Multilingual query by example spoken term detection for under-resourced languages," in *7^{th} Conf. on Speech Tech. and Human-Computer Dialogue (SpeD)*, Cluj-Napoca, Romania, 2013, pp. 1–6.

[115] C.-H. Lee, F. K. Soong, and B.-H. Juang, "A segment model based approach to speech recognition," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, New York, USA, Apr 1988, pp. 501–541 vol.1.

[116] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. on Acoust., Speech and Signal Process.*, vol. 26, no. 1, pp. 43–49, 1978.

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

[118] H. Wang and T. Lee, "CUHK system for the spoken web search task at MediaEval 2012," in *Working Notes Proc. of the MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[119] G. Mantena, "Query-by-example spoken term detection on low resource languages," Ph.D. Thesis, Int. Inst. of Info. Technology (IIIT-H), Computer Science and Engineering, Hyderabad, India, 2015.

[120] M. Müller, *Information Retrieval for Music and Motion*. Springer, 2007, vol. 2.

[121] C. Chan and L. Lee, "Integrating frame-based and segment-based dynamic time warping for unsupervised spoken term detection with spoken queries," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Prague, Czech Republic, 2011, pp. 5652–5655.

[122] C. Joder, F. Weninger, M. Wöllmer, and B. Schuller, "The TUM cumulative DTW approach for the MediaEval 2012 spoken web search task," in *Working Notes Proc. of the MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[123] M. R. Gajjar, R. Govindarajan, and T. Sreenivas, "Online unsupervised pattern discovery in speech using parallelization," in *Proc. INTERSPEECH*, Brisbane, Australia, 2008, pp. 2458–2461.

[124] Y. Zhang, K. Adl, and J. Glass, "Fast spoken query detection using lower-bound dynamic time warping on graphical processing units," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Kyoto, Japan, 2012, pp. 5173–5176.

[125] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, 2005.

[126] Y. Zhang and J. R. Glass, "An inner-product lower-bound estimate for dynamic time warping," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Prague, Czech Republic, 2011, pp. 5660–5663.

[127] P. Yang, L. Xie, Q. Luan, and W. Feng, "A tighter lower bound estimate for dynamic time warping," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Vancouver, BC, Canada, 2013, pp. 8525–8529.

[128] D. S. K. Pandia, M. S. Saranya, and H. A. Murthy, "A fast query-by-example spoken term detection for zero resource languages," in *Proc. Int. Conf. on Signal Processing and Comm., (SPCOM)*, IISc, Bangalore, India, June 12-15, 2016 2016, pp. 1–6.

[129] X. Anguera, R. Macrae, and N. Oliver, "Partial sequence matching using an unbounded dynamic time warping algorithm," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Dallas, Texas, USA, 2010, pp. 3582–3585.

[130] X. Anguera, "Information retrieval-based dynamic time warping," in *Proc. INTERSPEECH*, Lyon, France, 2013, pp. 1–5.

[131] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, Waikoloa, HI, USA, 2011, pp. 401–406.

[132] A. Jansen and B. Van Durme, "Indexing raw acoustic features for scalable zero resource search," in *Proc. INTERSPEECH*, Portland, Oregon, USA, 2012, pp. 2466–2469.

[133] B. George and B. Yegnanarayana, "Unsupervised query-by-example spoken term detection using segment-based bag of acoustic words," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 7133–7137.

[134] R. Shekhar and C. V. Jawahar, "Word image retrieval using bag of visual words," in *10th IAPR International Workshop on Document Analysis Systems, DAS 2012*, Gold Coast, Queenslands, Australia, 2012, pp. 297–301.

[135] B. George, A. Saxena, G. Mantena, K. Prahallad, and B. Yegnanarayana, "Unsupervised query-by-example spoken term detection using bag of acoustic words and non-segmental dynamic time warping," in *Proc. INTERSPEECH*, Singapore, 2014, pp. 1742–1746.

[136] K. Aoyama, A. Ogawa, T. Hattori, T. Hori, and A. Nakamura, "Graph index based query-by-example search on a large speech data set," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Vancouver, BC, Canada, 2013, pp. 8520–8524.

[137] K. Aoyama, A. Ogawa, T. Hattori, T. Hori, and A. Nakamura, "Zero-resource spoken term detection using hierarchical graph-based similarity search," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 7093–7097.

[138] K. Aoyama, A. Ogawa, T. Hattori, and T. Hori, "Double-layer neighborhood graph based similarity search for fast query-by-example spoken term detection," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, South Brisbane, Queensland, Australia, 2015, pp. 5216–5220.

[139] I. Szöke, M. Fapso, and K. Veselý, "BUT2012 approaches for spoken web search - MediaEval 2012," in *Working Notes Proc. of the MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[140] S. Kesiraju, G. Mantena, and K. Prahallad, "IIIT-H system for MediaEval 2014 QUESST," in *Working Notes Proc. of the MediaEval 2014 Workshop*, vol. 1263, Barcelona, Spain, October 16-17 2014.

[141] C. Allauzen, M. Mohri, and M. Saraclar, "General indexation of weighted automata - application to spoken utterance retrieval," in *HLT-NAACL 2004 Workshop: Interdisciplinary Approaches to Speech Indexing and Retrieval*, B. Ramabhadran and D. Oard, Eds. Boston, Massachusetts, USA: Association for Computational Linguistics, 2004, pp. 33–40.

[142] J. Tejedor, M. Fapšo, I. Szöke, J. Černockỳ, F. Grézl *et al.*, "Comparison of methods for language-dependent and language-independent query-by-example spoken term detection," *ACM Trans. on Information Systems (TOIS)*, vol. 30, no. 3, p. 18, 2012.

[143] A. Jansen, B. Van Durme, and P. Clark, "The JHU-HLTCOE spoken web search system for MediaEval 2012," in *Working Notes Proc. of the MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[144] A. Ali and M. Clementsby, "Spoken web search using an ergodic hidden Markov model of speech," in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[145] C. Chen, H. Lee, C. Yeh, and L. Lee, "Improved spoken term detection by feature space pseudo-relevance feedback," in *Proc. INTERSPEECH*, Makuhari, Japan, 2010, pp. 1672–1675.

[146] A. Muscariello, G. Gravier, and F. Bimbot, "Zero-resource audio-only spoken term detection based on a combination of template matching techniques," in *Proc. INTERSPEECH*, Florence, Italy, 2011, pp. 921–924.

[147] R. Shankar, A. Jain, K. T. Deepak, C. M. Vikram, and S. R. M. Prasanna, "Spoken term detection from continuous speech using ANN posteriors and image processing techniques," in *National Conference on Communication (NCC)*, IIT Guwahati, India, 2016, pp. 1–6.

[148] I. Szőke, M. Skácel, J. Černocký, and L. Burget, "Coping with channel mismatch in query-by-example - BUT QUESST 2014," in *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP)*, South Brisbane, Queensland, Australia, 2015, pp. 5838–5842.

[149] I. Szöke, M. Skácel, and L. Burget, "BUT QUESST 2014 system description," in *Working Notes Proc. of the MediaEval 2014 Workshop*, vol. 1263, Barcelona, Spain, 2014.

[150] H. Wang, T. Lee, C. Leung, B. Ma, and H. Li, "Using parallel tokenizers with DTW matrix combination for low-resource spoken term detection," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Vancouver, BC, Canada, 2013, pp. 8545–8549.

[151] F. Metze, E. Barnard, M. Davel, C. Van Heerden, X. Anguera, G. Gravier, and N. Rajput, "The spoken web search task," in *Working Notes Proc. of the MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[152] I. Szöke, J. Tejedor, M. Fapso, and J. Colás, "BUT-HCTLab approaches for spoken web search-MediaEval 2011," in *Working Notes Proc. of the MediaEval 2011 Workshop*, Pisa, Italy, 2011.

[153] A. Muscariello and G. Gravier, "Irisa MediaEval 2011 spoken web search system," in *Working Notes Proc. of the MediaEval 2011 Workshop*, Pisa, Italy, 2011.

[154] A. Buzo, H. Cucu, M. Safta, B. Ionescu, and C. Burileanu, "ARF @ mediaeval 2012: A Romanian ASR-based approach to spoken term detection," in *Working Notes Proc. of the MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[155] A. Varona, M. Penagarikano, L. J. Rodríguez-Fuentes, G. Bordel, and M. Diez, "GTTS system for the spoken web search task at MediaEval 2012," in *Working Notes Proc. of the MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[156] A. Abad and R. F. Astudillo, "The L2F spoken web search system for MediaEval 2012," in *Working Notes Proc. of the MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[157] X. Anguera, "Telefonica research system for the spoken web search task at MediaEval 2012," in *Working Notes Proc. of the MediaEval 2012 Workshop*, Pisa, Italy, 2012.

[158] H. Wang and T. Lee, "The CUHK spoken web search system for MediaEval 2013," in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[159] G. V. Mantena and K. Prahallad, “IIIT-H SWS 2013: Gaussian posteriorgrams of bottle-neck features for query-by-example spoken term detection,” in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[160] A. Buzo, H. Cucu, I. Molnar, B. Ionescu, and C. Burileanu, “SpeeD @ MediaEval 2013: A phone recognition approach to spoken term detection,” in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[161] R. Jarina, M. Kuba, R. Gubka, M. Chmulik, and M. Paralic, “UNIZA system for the spoken web search task at MediaEval2013,” in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[162] M. Bouallegue, G. Senay, M. Morchid, D. Matrouf, G. Linarès, and R. Dufour, “LIA@ MediaEval 2013 spoken web search task: An I-vector based approach,” in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[163] H. Wang and T. Lee, “CUHK system for QUESST task of MediaEval 2014,” in *Working Notes Proc. of the MediaEval 2014 Workshop*, vol. 1263, Barcelon, Spain, 2014.

[164] M. Calvo, M. Giménez, L.-F. Hurtado, E. Sanchis, and J. A. Gómez, “ELiRF at MediaEval2014: query by example search on speech task (QUESST),” in *Working Notes Proc. of the MediaEval 2014 Workshop*, vol. 1263, Barcelona, Spain, October 16-17 2014.

[165] J. Vavrek, P. Viszlay, M. Lojka, M. Pleva, and J. Juhár, “TUKE system for MediaEval 2014 QUESST,” in *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Catalunya, Spain, October 16-17, 2014.*, vol. 1263, Barcelona, Spain, October 16-17 2014.

[166] P. Yang, H. Xu, X. Xiao, L. Xie, C. Leung, H. Chen, J. Yu, H. Lv, L. Wang, S. J. Leow, B. Ma, C. E. Siong, and H. Li, “The NNI query-by-example system for MediaEval 2014,” in *Working Notes Proc. of the MediaEval 2014 Workshop*, vol. 1263, Barcelon, Spain, 2014.

[167] J. Proença, A. Veiga, and F. Perdigão, “The SPL-IT query by example search on speech system for mediaeval 2014,” in *Working Notes Proc. of the MediaEval 2014 Workshop*, vol. 1263, Barcelona, Spain, 2014.

[168] J. Proença, L. Castela, and F. Perdigão, “The SPL-IT-UC query by example search on speech system for MediaEval 2015,” in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[169] M. Skácel and I. Szöke, “BUT QUESST 2015 system description,” in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[170] S. Laguna, M. Calvo, L. F. Hurtado, and E. Sanchis, “Elirf at MediaEval 2015: Query by example search on speech task (QUESST),” in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[171] P. Lopez-Otero, L. D. Fernández, and C. García-Mateo, “GTM-UVigo systems for the query-by-example search on speech task at MediaEval 2015,” in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[172] H. Tulsiani and P. Rao, “The IIT-B query-by-example system for MediaEval 2015,” in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[173] J. Vavrek, P. Viszlay, M. Lojka, M. Pleva, J. Juhár, and M. Rusko, "TUKE at mediaeval 2015 QUESST," in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[174] M. Ma and A. Rosenberg, "CUNY systems for the query-by-example search on speech task at MediaEval 2015," in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[175] A. Caranica, A. Buzo, H. Cucu, and C. Burileanu, "SpeeD @ MediaEval 2015: Multilingual phone recognition approach to query by example STD," in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[176] C. Chung and Y. Chen, "NTU system at MediaEval 2015: Zero resource query by example spoken term detection using deep and recurrent neural networks," in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[177] J. Hou, V. T. Pham, C. Leung, L. Wang, H. Xu, H. Lv, L. Xie, Z. Fu, C. Ni, X. Xiao, H. Chen, S. Zhang, S. Sun, Y. Yuan, P. Li, T. L. Nwe, S. Sivadas, B. Ma, E. Chng, and H. Li, "The NNI query-by-example system for MediaEval 2015," in *Working Notes Proc. of the MediaEval 2015 Workshop*, Wurzen, Germany, 2015.

[178] X. Anguera, F. Metze, A. Buzo, I. Szöke, and L. J. Rodríguez-Fuentes, "The spoken web search task," in *Working Notes Proc. of the MediaEval 2013 Workshop*, Barcelona, Spain, 2013.

[179] H. A. Patil and M. C. Madhavi, "Significance of magnitude and phase information via VTEO for humming based biometrics," in *Proc. Int. Conf. on Biometrics, (ICB)*, New Delhi, India, 2012, pp. 372–377.

[180] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, *The HTK book (for HTK version 3.4)*. Cambridge University Engineering Department, 2006.

[181] J. F. Kaiser, "On a simple algorithm to calculate the energy of a signal," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Albuquerque, New Mexico, USA, 1990, pp. 381–384.

[182] H. A. Patil and M. C. Madhavi, "Combining evidences from magnitude and phase information using VTEO for person recognition using humming," *Special Issue of Recent advances in speaker and language recognition and characterization in Computer Speech and Language, Elsevier*.

[183] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Trans. on Speech and Audio Process.*, vol. 3, no. 1, pp. 72–83, 1995.

[184] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[185] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4^{th} ed. Tata McGraw-Hill Education, 2002.

[186] N. Brümmer and E. de Villiers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," *CoRR*, vol. abs/1304.2865, 2013. [Online]. Available: http://arxiv.org/abs/1304.2865

[187] N. Brümmer, L. Burget, J. Cernocký, O. Glembek, F. Grézl, M. Karafiát, D. A. van Leeuwen, P. Matejka, P. Schwarz, and A. Strasheim, "Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006," *IEEE Trans. Audio, Speech & Language Processing*, vol. 15, no. 7, pp. 2072–2084, 2007.

[188] L. J. Rodríguez, M. Peñagarikano, A. Varona, M. Díez, G. Bordel, D. M. González, J. A. V. López, A. Miguel, A. Ortega, E. Lleida, A. Abad, O. Koller, I. Trancoso, P. Lopez-Otero, L. D. Fernández, C. García-Mateo, R. Saeidi, M. Soufifar, T. Kinnunen, T. Svendsen, and P. Fränti, "Multi-site heterogeneous system fusions for the albayzin 2010 language recognition evaluation," in *Workshop on Automatic Speech Recognition & Understanding (ASRU)*, Waikoloa, HI, USA, 2011, pp. 377–382.

[189] L^2F, "STDfusion," https://www.l2f.inesc-id.pt/w/STDfusion, 2013, {Last Accessed on June 1, 2017}.

[190] T. F. Quatieri, *Discrete-Time Speech Signal Processing: Principles and Practice*, 3^{rd} ed. Pearson Education, 2006.

[191] L. Lee and R. Rose, "A frequency warping approach to speaker normalization," *IEEE Trans. on Speech and Audio Process.*, vol. 6, no. 1, pp. 49–60, 1998.

[192] S. Umesh, L. Cohen, and D. Nelson, "Frequency warping and the mel scale," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 104–107, 2002.

[193] L. Lee and R. C. Rose, "Speaker normalization using efficient frequency warping procedures," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Atlanta, Georgia, USA, 1996, pp. 353–356.

[194] S. Sharma, M. C. Madhavi, and H. A. Patil, "Vocal tract length normalization for vowel recognition in low resource languages," in *Int. Conf. on Asian Lang. Process., IALP 2014*, Kuching, Malaysia, 2014, pp. 54–57.

[195] E. Eide and H. Gish, "A parametric approach to vocal tract length normalization," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, vol. 1, Atlanta, Georgia, USA, 1996, pp. 346–348.

[196] E. B. Gouvêa and R. M. Stern, "Speaker normalization through formant-based warping of the frequency scale," in *EUROSPEECH 1997*, Rhodes, Greece, 1997, pp. 1139–1142.

[197] F. Müller and A. Mertins, "Enhancing vocal tract length normalization with elastic registration for automatic speech recognition," in *Proc. INTERSPEECH*, Portland, Oregon, USA, 2012, pp. 1364–1367.

[198] E. P. Neuburg, "Frequency warping by dynamic programming," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, New York, USA, 1988, pp. 573–575.

[199] S. Umesh, L. Cohen, N. Marinovic, and D. J. Nelson, "Scale transform in speech analysis," *IEEE Trans. on Speech and Audio Process.*, vol. 7, no. 1, pp. 40–45, 1999.

[200] J. W. McDonough, W. Byrne, and X. Luo, "Speaker normalization with all-pass transforms," in *Proc. 5^{th} Int. Conf. on Spoken Language Processing (ICSLP)*, Sydney, Australia, 1998, pp. 2307–2310.

[201] S. Panchapagesan, “Frequency warping by linear transformation of standard MFCC,” in *Proc. INTERSPEECH- ICSLP*, Pittsburgh, Pennsylvania, USA, 2006, pp. 397–400.

[202] D. R. Sanand and S. Umesh, “VTLN using analytically determined linear-transformation on conventional MFCC,” *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 20, no. 5, pp. 1573–1584, 2012.

[203] O. K. Shin, “A vector-quantizer based method of speaker normalization,” in *Int. Conf. on Computer and Information Science*, Jeju Island, South Korea, 2005, pp. 402–407.

[204] P. Zhan and M. Westphal, “Speaker normalization based on frequency warping,” in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Munich, Bavaria, Germany, 1997, pp. 1039–1042.

[205] R. Sinha and S. Umesh, “A shift-based approach to speaker normalization using non-linear frequency-scaling model,” *Speech Communication, Elsevier*, vol. 50, no. 3, pp. 191–202, 2008.

[206] M. G. Maragakis and A. Potamianos, “Region-based vocal tract length normalization for ASR,” in *Proc. INTERSPEECH*, Brisbane, Australia, 2008, pp. 1365–1368.

[207] L. Welling, S. Kanthak, and H. Ney, “Improved methods for vocal tract normalization,” in *Proc. Int. Conf. on Acoustics, Speech, and Signal Process., ICASSP*, Phoenix, Arizona, USA, 1999, pp. 761–764.

[208] L. Welling, H. Ney, and S. Kanthak, “Speaker adaptive modeling by vocal tract normalization,” *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 6, pp. 415–426, 2002.

[209] K.-F. Lee and H.-W. Hon, “Speaker-independent phone recognition using hidden Markov models,” *IEEE Trans. on Acoustics, Speech and Signal Proces.*, vol. 37, no. 11, pp. 1641–1648, 1989.

[210] N. Ueda and R. Nakano, “Deterministic annealing EM algorithm,” *Neural Networks*, vol. 11, no. 2, pp. 271–282, 1998.

[211] N. J. Shah, H. A. Patil, M. C. Madhavi, H. B. Sailor, and T. B. Patel, “Deterministic annealing EM algorithm for developing TTS system in Gujarati,” in 9th *Int. Symp. on Chinese Spoken Language Process. (ISCSLP), 2014*, Singapore, 2014, pp. 526–530.

[212] I. Naim and D. Gildea, “Convergence of the EM algorithm for Gaussian mixtures with unbalanced mixing coefficients,” in *Proc. of the* 29th *Int. Conf. on Machine Learning, ICML 2012*, Edinburgh, Scotland, UK, June 26 - July 1 2012, pp. 1655–1662.

[213] S. Garimella and H. Hermansky, “Factor analysis of mixture of auto-associative neural networks for speaker verification,” in *Odyssey*, Singapore, 2012, pp. 92–97.

[214] M. C. Madhavi and H. A. Patil, “Design of mixture of GMMs for query-by-example spoken term detection,” December 12-15 2017, submitted for possible publication in Computer Speech and Language, Elsevier.

[215] S. Garimella, “Mixture of GMMs,” https://sites.google.com/site/sivaramiisc/mixGMMs.pdf?attredirects=0, {Last Accesed on Dec 9, 2016}.

[216] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. New York, NY, USA: Cambridge University Press, 2004.

[217] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *INTERSPEECH 2013*, Lyon, France, 2013, pp. 2345–2349.

[218] D. W. Mitchell, "88.27 more on spreads and non-arithmetic means," *The Mathematical Gazette*, vol. 88, no. 511, pp. 142–144, 2004.

[219] S. Furui, "On the role of spectral transition for speech perception," *The Journal of the Acoustical Society of America*, vol. 80, no. 4, pp. 1016–1025, 1986.

[220] K. Levin, K. Henry, A. Jansen, and K. Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *Proc. IEEE Automatic Speech Recognition and Understanding (ASRU) Workshop*, Olomouc, Czech Republic, 2013, pp. 410–415.

[221] A. Moller, *Auditory Physiology*. Academic Press, 2012.

[222] S. Dusan and L. R. Rabiner, "On the relation between maximum spectral transition positions and phone boundaries," in *Proc. INTERSPEECH-ICSLP*, Pittsburgh, PA, USA, 2006, pp. 645–648.

[223] M. C. Madhavi, H. A. Patil, and B. B. Vachhani, "Spectral transition measure for detection of obstruents," in *23rd European Signal Process. Conf. (EUSIPCO)*, Nice, France, 2015, pp. 330–334.

[224] N. J. Shah, B. B. Vachhani, H. B. Sailor, and H. A. Patil, "Effectiveness of PLP-based phonetic segmentation for speech synthesis," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Florence, Italy, 2014, pp. 270–274.

[225] Y. Y. Ji Xu, Ge Zhang, "Effective utilization of multiple examples in query-by-example spoken term detection," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Shanghai, China, 2016, pp. 5440–5444.

[226] H. Lee, C. Chen, and L. Lee, "Integrating recognition and retrieval with relevance feedback for spoken term detection," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 20, no. 7, pp. 2095–2110, 2012.

[227] J. O. Smith and J. S. Abel, "Bark and ERB bilinear transforms," *IEEE Trans. on Audio, Speech, & Language Process.*, vol. 7, no. 6, pp. 697–708, 1999.

[228] H. A. Murthy and V. Gadde, "The modified group delay function and its application to phoneme recognition," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, vol. 1, Hong Kong, 2003, pp. 68–71.

[229] F. Eyben, F. Weninger, F. Groß, and B. W. Schuller, "Recent developments in opensmile, the munich open-source multimedia feature extractor," in *ACM Multimedia Conference, MM '13, Barcelona, Spain, October 21-25, 2013*, 2013, pp. 835–838.

[230] A. Muscariello, G. Gravier, and F. Bimbot, "Unsupervised motif acquisition in speech via seeded discovery and template matching combination," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 20, no. 7, pp. 2031–2044, 2012.

[231] Y. Wang, D. Yu, Y.-C. Ju, and A. Acero, "An introduction to voice search," *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 28–38, 2008.

[232] H.-y. Lee and L.-s. Lee, "Enhanced spoken term detection using support vector machines and weighted pseudo examples," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 21, no. 6, pp. 1272–1284, 2013.

[233] S. R. M. Prasanna, B. S. Reddy, and P. Krishnamoorthy, "Vowel onset point detection using source, spectral peaks, and modulation spectrum energies," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 17, no. 4, pp. 556–565, 2009.

[234] A. A. Reddy, "Using syllable boundary information for query-by-example spoken term detection," M.S. Thesis, Int. Inst. of Info. Technology (IIIT-H), Electronics and Communications Engineering, Hyderabad, India, 2015.

[235] S. Panchapagesan, "Frequency warping by linear transformation of standard MFCC," in *Proc. INTERSPEECH- ICSLP*, Pittsburgh, Pennsylvania, USA, 2006, pp. 397–400.

[236] G. Mantena and K. Prahallad, "Use of GPU and feature reduction for fast query-by-example spoken term detection," in 11^{th} *Int. Conf. on Natural Language Processing (ICON)*, Goa, India, 2014, p. 56.

[237] Y. Yuan, C. Leung, L. Xie, B. Ma, and H. Li, "Pairwise learning multi-lingual bottleneck features for low-resource query-by-example spoken term detection," in *Proc. Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, New orleans, USA, 2017, pp. 5645–5649.

[238] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH 2015*, Dresden, Germany, 2015, pp. 3586–3589.

[239] J. Proença and F. Perdigão, "Segmented dynamic time warping for spoken query-by-example search," in *Proc. INTERSPEECH*, San Fransisco, USA, 2016, pp. 750–754.

[240] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query-by-humming: Musical information retrieval in an audio database," in *Proc. of* 3^{rd} *ACM Int. Conf. on Multimedia*, San Francisco, CA, USA, 1995, pp. 231–236.

[241] P. K. Jain, R. Jain, H. A. Patil, and T. K. Basu, "Design of a query-by-humming system for Hindi songs using DDTW based approach," in *Int. Conf. on Asian Lang. Process., (IALP)*, Penang, Malaysia, 2011, pp. 240–243.

[242] J. R. Jang and H. Lee, "A general framework of progressive filtering and its application to query by singing/humming," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 16, no. 2, pp. 350–358, 2008.

[243] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1," *NASA STI/Recon Technical Report N*, vol. 93, p. 27403, 1993.

[244] H. Wang, T. Lee, C. Leung, B. Ma, and H. Li, "Unsupervised mining of acoustic subword units with segment-level Gaussian posteriorgrams," in *Proc. INTERSPEECH*, Lyon, France, 2013, pp. 2297–2301.

[245] H. A. Patil, M. C. Madhavi, K. D. Malde, and B. B. Vachhani, “Phonetic transcription of fricatives and plosives for Gujarati and Marathi languages,” in *Int. Conf. on Asian Lang. Processing*, Hanoi, Vietnam, 2012, pp. 177–180.

[246] K. D. Malde, B. B. Vachhani, M. C. Madhavi, N. H. Chhayani, and H. A. Patil, “Development of speech corpora in Gujarati and Marathi for phonetic transcription,” in *Oriental COCOSDA*, Gurgaon, India, 2013, pp. 1–6.

[247] “A map of Gujarat state,” http://upload.wikimedia.org/wikipedia/commons/3/3e/Map_GujDist_Kuchchh.png, {Last Accessed on 14th Sep. 2012}.

[248] “A map of maharashtra state,” http://www.besttofind.com/img/Map-of-Maharashtra.gif, {Last Accessed on 14th Sep. 2012}.

[249] S. Sharma, M. C. Madhavi, and H. A. Patil, “Development of vocal tract length normalized phonetic engine for gujarati and marathi languages,” in 17th *Oriental Chapter of the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA)*, Phuket, Thailand, 2014, pp. 1–6. [Online]. Available: https://doi.org/10.1109/ICSDA.2014.7051439

[250] J. C. Steinberg and N. R. French, “Factors governing the intelligibilty of speech sounds,” *The Journal of the Acoustical Society of America*, vol. 19, pp. 90–119, 1947.

[251] K. M. Carbonell and A. J. Lotto, “Degraded word recognition in isolation vs. carrier phrase,” *The Journal of the Acoustical Society of America*, vol. 134, no. 5, pp. 4073–4073, nov 2013.

[252] Y. Kim and J. Smith, “A speech feature based on Bark frequency warping-the non-uniform linear prediction (NLP) cepstrum,” in *IEEE Workshop on Applications of Signal Process. to Audio & Acoust.*, New Paltz, NY, 1999, pp. 131–134.

[253] M. C. Madhavi, S. Sharma, and H. A. Patil, “Vocal tract length normalization features for audio search,” in *Text, Speech, and Dialogue: 18th International Conference, TSD 2015, Pilsen,Czech Republic, September 14-17, 2015, Proceedings*, P. Král and V. Matoušek, Eds. Cham: Springer International Publishing, 2015, pp. 387–395.

[254] A. Andreou, T. Kamm, and J. Cohen, “Experiments in vocal tract normalization,” in *Proc. CAIP Workshop: Frontiers in Speech Recognition II*, Piscataway,NJ,US, 1994.

[255] P. Zhan and A. Waibel, “Vocal tract length normalization for large vocabulary continuous speech recognition,” DTIC Document, Tech. Rep., 1997.

[256] D. O’shaughnessy, *Speech Communication: Human and Machine*, 2nd, Ed. Universities press, 2001.

[257] “The NIST year 2010 speaker recognition evaluation plan,” http://www.itl.nist.gov/iad/mig/tests/spk/2010/NIST_SRE10_evalplan.r6.pdf, 2010, {Last Accessed on May 27, 2017}.

List of Publications from Thesis

Journals:

1. M. C. Madhavi, and H. A. Patil, "Partial Matching and Search Space Reduction for QbE-STD," in *Computer Speech & Language, Elsevier,* vol. 45, pp. 58-82, September 2017.
2. M. C. Madhavi, and H. A. Patil, "Design of Mixture of GMMs for Query-by-Example Spoken Term Detection," submitted for possible publication *Computer Speech and Language, Elsevier.*

Book Chapters:

3. M. C. Madhavi, and H. A. Patil, "Spoken Keyword Retrieval using Source and System Features," to be appeared in *Int. Conf. on Pattern Recognition and Machine Intelligence (PReMI)*, Lecture Notes in Computer Science, LNCS, Kolkata, India, Dec. 05 - 08, 2017.
4. M. C. Madhavi, S. Sharma, H. A. Patil, "VTLN Using Different Warping Functions for Template Matching," *Machine Intelligence and Big Data in Industry*, Springer International Publishing, D. Ryżko, et al. (Ed.), pp. 111-121, 2016.
5. M. C. Madhavi, S. Sharma, and H. A. Patil, "Vocal tract length normalization features for audio search," in *Int. Conf. Text, Speech, and Dialogue, TSD*, Pavel Král and Václav Matousek (Ed.), Pilsen,Czech Republic, pp. 387-395, 2015.

Conferences:

6. M. C. Madhavi, and H. A. Patil, "VTLN-Warped Gaussian Posteriorgram for QbE-STD," in 25th *European Signal Process. Conf., (EUSIPCO)*, Kos island, Greece, Aug. 28-Sep. 2, 2017, pp. 563-567.
7. M. C. Madhavi, and H. A. Patil, "Combining Evidences from Detection Sources for Query-by-Example Spoken Term Detection," to be appeared in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA 2017*, Kuala Lumpur, Malaysia, December 12-15, 2017.
8. M. C. Madhavi and H. A. Patil, "Two stage zero-resource approaches for QbE-STD," to be appeared in 9th *Int. Conf. on Advances in Pattern Recognition, ICAPR 2017*, Bangalore, India.
9. M. C. Madhavi and H. A. Patil, "Modification in Sequential Dynamic Time Warping for Fast Computation of Query-by-Example Spoken Term Detection Task," in *Int. Conf. on Signal Processing and Communications (SPCOM)* , IISc Bangalore, India, 12-15 June, 2016.
10. M. C. Madhavi, H. A. Patil, and B. B. Vachhani, "Spectral transition measure for detection of obstruents," in 23rd *European Signal Process. Conf., (EUSIPCO)*, Nice, France, pp. 330–334, 2015.
11. S. Sharma, M. C. Madhavi and H. A. Patil, "Vocal Tract Length Normalization for Vowel Recognition in Low Resource Languages," in *Int. Conf. on Asian Lang. Process. (IALP '14)*, Kuching, Malaysia, pp. 54-57, 2014.

12. M. C. Madhavi, S. Sharma, and H. A. Patil, “Development of language resources for speech application in Gujarati and Marathi,” in *Int. Conf. on Asian Lang. Process., (IALP)*, Kuching, Malaysia, pp. 115-118, 2014.
13. S. Sharma, M. C. Madhavi and H. A. Patil, “Development of Vocal Tract Length Normalized Phonetic Engine for Gujarati and Marathi Languages,” in *The* 17th *Oriental COCOSDA’14*, Phuket, Thailand, Sept. 10-12, 2014.
14. N. Shah, H. Patil, M. Madhavi, H. Sailor and T. Patel, “Deterministic Annealing EM Algorithm for Developing TTS System in Gujarati,” in *the* 9th *Int. Symposium on Chinese Spoken Language Processing, ISCSLP’14*, Singapore, pp. 526-530, 12-14 Sep.14.
15. K. D. Malde, B. B. Vachhani, M. C. Madhavi, N. H. Chhayani, and H. A. Patil. “Development of speech corpora in Gujarati and Marathi for phonetic transcription,” in *Int. Conf. Oriental COCOSDA held jointly with 2013 Conf. on Asian Spoken Lang. Research and Evaluation (O-COCOSDA/CASLRE)*, 2013, Gurgaon, India, pp. 1-6. 2013.
16. H. A. Patil and M. C. Madhavi, “Significance of magnitude and phase information via VTEO for humming based biometrics,” in *Proc. Int. Conf. on Biometrics (ICB)*, New Delhi, India, pp. 372-377, 2012.

Other relevant publications:

International Journals:

17. H. A. Patil, and M. C. Madhavi, “Combining Evidences from Magnitude and Phase Information using VTEO for Person Recognition using Humming,” in special issue of *Recent advances in speaker and language recognition and characterization, Computer Speech and Language, Elsevier*
18. H. A. Patil, M. C. Madhavi, and K. K. Parhi,“ Static and dynamic information derived from source and system features for person recognition from humming,” *I. J. Speech Technology* , vol. 15, no. 3, pp. 393-406, 2012.

International Conferences:

19. N. J. Shah, M. C. Madhavi, and H. A. Patil, “Unsupervised vocal tract length warped posterior features for non-parallel voice conversion,” submitted for possible publication in ICASSP 2018, Calgary, Alberta, Canada, 15-20 April 2018.
20. M. C. Madhavi, and H. A. Patil, “Exploiting Variable Length Teager Energy Operator in Melcepstral Features for Person Recognition from Humming,” in *the* 9th *Int. Symposium on Chinese Spoken Language Processing, ISCSLP’14*, Singapore, pp. 624-628, 12-14 Sep. 2014.
21. A. Undhad, H. Patil, and M. C. Madhavi, “Vowel Landmark Detection for Low Resource Languages,” in *the* 9th *Int. Symposium on Chinese Spoken Language Processing, ISCSLP’14*, Singapore, pp. 546-550, 12-14 Sep. 2014.

Brief Biography

Maulik Madhavi received B.E. degree (Electronics and Communication) from Saurashtra University in 2009. In 2011, he received M.Tech (ICT) degree (Communication Systems specialization) from Dhirubhai Ambani Institute of Information and Communication Technology (DA-IICT), Gandhinagar, India. Currently, he is a doctoral student at DA-IICT under the supervision of Prof. Hemant A. Patil. He has been a part of DeitY sponsored consortium project, "Development of Prosodically Guided Phonetic Engine for Searching Speech Databases in Indian Languages" during April 2012- June 2014 (2 years and three months). During his masters and doctoral studies at DA-IICT, he was teaching assistant/tutor at DA-IICT for eight different courses (August 2009-April 2012, July 2014-May 2017).

He received IAPR (International Association for Pattern Recognition) Travel Scholarship for presenting our joint paper in International Conference on Biometrics, ICB'12, Delhi, India. His research interests include query-by-example spoken term detection, pattern recognition, and humming-based person recognition. He is a student member of IEEE, IEEE Signal Processing Society and International Speech Communication Association (ISCA).