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-
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.
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 ($\Delta$) and delta-delta ($\Delta^2$) 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 $\psi$) for discrete-time signal, $x(n)$, is defined as [181]:

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

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.,

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

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 \( S_{i,j} = \text{logarithm of subband energy for } i^{th} \text{ frame index and } j^{th} \text{ subband filter.} \)

### 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 \cite{183}. Each such acoustic events are caused due to average vocal tract configuration, which is characterized by the mean component \( \mu_i \). The variation in vocal tract structure is characterized by the covariance \( \Sigma_i \). 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 \cite{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 \( P(C_k|o_t) \) (for \( k^{th} \) cluster and \( t^{th} \) speech frame index) is:

\[
P(C_k|o_t) = \frac{\pi_k \mathcal{N}(o_t; \mu_k, \Sigma_k)}{\sum_{j=1}^{N_p} \pi_j \mathcal{N}(o_t; \mu_j, \Sigma_j)},
\]

where the likelihood of feature \( o_t \) being in \( k^{th} \) cluster is,

\[
\mathcal{N}(o_t; \mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^N |\Sigma_k|}} \exp \left( -\frac{1}{2} (o_t - \mu_k)^T \Sigma_k^{-1} (o_t - \mu_k) \right).
\]

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 \cite{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, $q_y = (q^1_y, q^2_y, \cdots, q^N_y)$ and a test utterance, $t_x = (t^1_x, t^2_x, \cdots, t^M_x)$. The local distance between two posterior vectors, namely, $t^i_x$ and $q^j_y$, is computed using the symmetric Kullback-Leibler (KL) divergence and is defined as [82, 98]:

$$
\text{Dis}_{t^i_x, q^j_y} = \sum_{k=1}^{N_q} t^i_x(k) \log \frac{t^i_x(k)}{q^j_y(k)} + \sum_{k=1}^{N_q} q^j_y(k) \log \frac{q^j_y(k)}{t^i_x(k)}.
$$

(3.5)
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 \(pdfs\) [82, 98]. The KL divergence between two \(pdfs\) 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 \(D = \text{Dis}_{t_x,q_y}\).

Initialization:
Figure 3.3: The graphical representation of matrices associated in subDTW operation using local symmetrical local constraints. After [2], the graphical representation of matrices are rotated as row indicates query frame index and column indicates test utterance frame index. Note that image plots are shown in Figure 3.2: (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.
For $j = 1, i = 1, 2, \cdots, M$:

$$S(i, j) = D(i, j),$$
$$T(i, j) = 1,$$
$$P(i, j) = i. \quad (3.6)$$

**Path-tracing**: For $i = 1$:

$$S(i, j) = \sum_{k=1}^{j} D(i, k),$$
$$T(i, j) = j,$$
$$P(i, j) = i = 1. \quad (3.7)$$

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

$$\Omega = \{(i, j-1), (i-1, j-1), (i-1, j)\}, \quad (3.8)$$
$$\begin{align*}
(r, s) &= \arg\min_{(a, b) \in \Omega} S(a, b), \quad (3.9) \\
S(i, j) &= S(r, s) + D(i, j), \quad (3.10) \\
T(i, j) &= T(r, s) + 1, \quad (3.11) \\
P(i, j) &= P(r, s). \quad (3.12)
\end{align*}$$

An algorithm 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:** # 1st column, i.e., $j = 1$

1: for $i = 1$ to $M$ do
2: $S(i, 1) = D(i, 1)$
3: $T(i, 1) = 1$
4: $P(i, 1) = i$
5: end for

**Path tracing:** # 1st row, i.e., $i = 1$

6: for $j = 2$ to $N$ do
7: $S(1, j) = S(1, j - 1) + D(1, j)$
8: $T(1, j) = j$
9: $P(1, j) = 1$
10: end for

**Path tracing:** # For the rest: $i > 1$ and $j > 1$

11: for $i = 2$ to $M$ do
12: for $j = 2$ to $N$ do
13: $\Omega = \{(i, j - 1), (i - 1, j - 1), (i - 1, j)\}$
14: $(r, s) = \text{argmin}_{(a, b) \in \Omega} S(a, b)$
15: # Selecting the predecessor
16: $S(i, j) = S(r, s) + D(i, j)$
17: $T(i, j) = T(r, s) + 1$
18: $P(i, j) = P(r, s)$
19: end for
20: 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 \(\mu_q\) and \(\sigma_q\) 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:**

1: **for** $i = 1$ to $M$ do
2:  $dist(i) = \frac{S(i,N)}{T(i,N)}$ \# Path length normalized DTW distance
3:  $wr(i) = \frac{(i-P(i,N))}{N}$ \# Slope constraint
4:  \textit{Check valid warping path :}
5:  \textbf{if} $(wr(i) \leq 2) \& (wr(i) \geq \frac{1}{2})$ \textbf{then}
6:  $V_{wr}(i) = 1$
7:  \textbf{else}
8:  $V_{wr}(i) = 0$
9:  \textbf{end if}
10: $G = \{0\}_{M \times 1}$ \# Initialize group assignment as 0 (no group)
11: $CG = 1$ \# Current group
12: \textbf{if} $V_{wr}(i) = 1$ \textbf{then}
13:  $G(i) = CG$
14:  \textbf{else}
15:  $CG = CG + 1$
16:  \textbf{end if}
17: \textbf{end for}

**Select best warping paths:**

18: \textbf{for} $k = 1$ to max($G$) \textbf{do}
19:  Let, the set $Sk := \{i | G(i) = k\}$
20:  $dv(k) = \min_{i \in Sk} dist(i)$
21:  $epv(k) = \arg\min_{i \in Sk} dist(i)$
22:  $spv(k) = P(epv(k), N)$
23: \textbf{end for}
24: 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) Unnor-
malized scores (DTW distance/cost), and (b) normalized scores. The PDF is ap-
proximated from the histogram with 20 bins.

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

**Input:** Unnormalized $N_{top}$ distance values: $d_{s1}, d_{s2}, \ldots, d_{sN_{top}}$.

**Output:** Normalized $N_{top}$ score values: $s_1, s_2, \ldots, s_{N_{top}}$.

1: Sample mean: $\mu_q = \frac{1}{N_{top}} \sum_{k=1}^{N_{top}} d_{sk}$

2: Sample standard deviation: $\sigma_q = \sqrt{\frac{1}{N_{top}-1} \sum_{k=1}^{N_{top}} (d_{sk} - \mu_q)^2}$

3: Normalized scores: $s_i = -\frac{(d_{si} - \mu_q)}{\sigma_q}$, 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 distribu-
tion 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 nor-
malized 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 achieve 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 $\theta$ 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].

$$\hat{s}_i = \xi_0 + \sum_{i=1}^{NS} \xi_i s_i^t,$$  \hspace{1cm} (3.13)

where $s_i^t$ is the score of $i^{th}$ system for $t^{th}$ trial, $\xi_i$’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. for $i = 1$ to $M$
2. $S(i, 1) = D(i, 1)$
3. $T(i, 1) = 1$
4. $P(i, 1) = i$
5. end for

### Path tracing:

1. for $i = 3$ to $M$
2. for $j = 2$ to $N$
3. $\Omega = \{(i, j - 1), (i - 1, j - 1), (i - 2, j - 1)\}$
4. $(r, s) = \arg\min_{(a, b) \in \Omega} S(a, b)$
5. Selecting the predecessor
6. $S(i, j) = S(r, s) + D(i, j)$
7. $T(i, j) = T(r, s) + 1$
8. $P(i, j) = P(r, s)$
9. end for
10. end for

---

Note that, in local constraints $LC_2$, the values in matrix $T$ depends only on the number of frames in query, i.e., $N$. 
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:* # 1st two columns, i.e., $j = 1, 2$

1: for $j = 1$ to $2$ do
2:     for $i = 1$ to $M$ do
3:         $S(i, j) = D(i, j)$
4:         $T(i, j) = j$
5:         $P(i, j) = i$
6:     end for
7: end for

*Path tracing:* # 1st row, i.e., $i = 1$

8: for $j = 2$ to $N$ do
9:     $S(1, j) = S(1, j - 1) + D(1, j)$
10:    $T(1, j) = j$
11:    $P(1, j) = 1$
12: end for

*Path tracing:* # For the rest: $i > 1$ and $j > 2$

13: for $i = 2$ to $M$ do
14:     for $j = 3$ to $N$ do
15:         $\Omega = \{(i - 1, j - 2), (i - 1, j - 1), (i - 1, j)\}$
16:         $(r, s) = \arg\min_{(a, b) \in \Omega} S(a, b)$  
17:         Selecting the predecessor
18:         $S(i, j) = S(r, s) + D(i, j)$
19:         $T(i, j) = T(r, s) + 1$
20:         $P(i, j) = P(r, s)$
21:     end for
22: end for
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 LC2 gives relatively better performance than the LC1 and LC3 [2, 53]. The local constraint LC2 allows more frames to be inserted from test utterance to the query, which is suited for QbE-STD [53]. Hence, we have used, LC2, 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 LC1 is shown in Table 3.4. It can be observed that a slight improvement in LC1 is obtained. However, the performance is still not better than the LC2.

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].

<table>
<thead>
<tr>
<th>Feature Vector</th>
<th>Dev Set no OLN</th>
<th>Dev Set OLN</th>
<th>Eval Set no OLN</th>
<th>Eval Set OLN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CZ</td>
<td>0.348</td>
<td>0.357</td>
<td>0.318 (0.313)</td>
<td>0.324 (0.322)</td>
</tr>
<tr>
<td>HU</td>
<td>0.357</td>
<td>0.359</td>
<td>0.331 (0.325)</td>
<td>0.334 (0.332)</td>
</tr>
<tr>
<td>RU</td>
<td>0.367</td>
<td>0.360</td>
<td>0.340 (0.338)</td>
<td>0.349 (0.346)</td>
</tr>
</tbody>
</table>

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].

The matrix $T$ 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,

$$(r, s) = \arg \min_{(a,b) \in \Omega} \frac{S(a,b)}{T(a,b)} = \arg \min_{(a,b) \in \Omega} \frac{S(a,b)}{j-1} = \arg \min_{(a,b) \in \Omega} S(a,b). \quad (3.14)$$

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 log-cosine 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 $C_{	ext{mix}}^{\min}$. In the next chapter, we will discuss the representation perspective for the design of QbE-STD system.