

Spoken Term Detection on Low Resource Languages

B Naresh Reddy

A Thesis Submitted to
Indian Institute of Technology Hyderabad
In Partial Fulfillment of the Requirements for
The Degree of Master of Technology



Department of Electrical Engineering

Dec 2015

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

B.N. Reddy.
(Signature)

B. Naresh Reddy
(B Naresh Reddy)

EE13M0001
(Roll No.)

Approval Sheet

This Thesis entitled Spoken Term Detection on Low Resource Languages by B Naresh Reddy is approved for the degree of Master of Technology from IIT Hyderabad

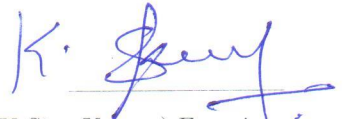


(Dr. Sumohana Channappayya) Examiner
Dept. of Electrical Engineering
IITH

(Dr. Challa Subrahmanya Sastry) Examiner
Dept. of Mathematics
IITH



(Dr. Sri Rama Murty Kodukula) Adviser
Dept. of Electrical Engineering
IITH



(Dr. K Siva Kumar) Examiner
Dept. of Electrical Engineering
IITH

Acknowledgements

First of all, I would like to thank to my thesis advisor, Dr. Sri Rama Murty Kodukula, for his help and support during this thesis work. He has always given me valuable comments during this work and has also guided me at every stage of my M.Tech studies. I am also thankful for allowing me to be a part of the SIPLab group, where this thesis has been mainly developed. The group provided me with the perfect environment to achieve my goals.

Finally, I would like to thank to my parents, my brother who have been always there for me. Without them, this work had not been possible. And thanks to all people who, as much as they possibly could, have supported and guided me through this thesis.

Dedication

This thesis is dedicated to my parents. For their endless love, support and encouragement.

Abstract

Developing efficient speech processing systems for low-resource languages is an immensely challenging problem. One potentially effective approach to address the lack of resources for any particular language, is to employ data from multiple languages for building speech processing sub-systems.

This thesis investigates possible methodologies for Spoken Term Detection (STD) from low-resource Indian languages. The task of STD intend to search for a query keyword, given in text form, from a considerably large speech database. This is usually done by matching templates of feature vectors, representing sequence of phonemes from the query word and the continuous speech from the database. Typical set of features used to represent speech signals in most of the speech processing systems are the mel frequency cepstral coefficients (MFCC). As speech is a very complex signal, holding information about the textual message, speaker identity, emotional and health state of the speaker, etc., the MFCC features derived from it will also contain information about all these factors. For efficient template matching, we need to neutralize the speaker variability in features and stabilize them to represent the speech variability alone.

The process of stabilizing MFCC features can be done in any of the following ways, namely, (1) Supervised process (2) Unsupervised process and (3) Semi supervised process. Supervised method requires a large amount of labeled training data, which are generally used to train a hybrid model including a Hidden Markov Model (HMM) and a Multi-Layer Perceptron (MLP) network. The phonetic posterior features extracted using this hybrid model are more stable against speaker variability than MFCC and will replace MFCC in template matching. The posterior features had delivered 25 percent improvement in STD performance, in comparison with MFCC. The supervised setting is suitable for STD from high-resource languages where considerable training data is available. The unsupervised method does not call for labeled training data and is usually carried out by training Gaussian mixture models (GMM) to capture joint density functions of MFCC features. Posterior features extracted from trained GMMs will replace MFCC in STD and it is observed that they had delivered 8 percent improvement in STD performance in comparison to MFCC. The unsupervised setting is better suitable for low-resource languages, than the supervised setting is. But for high-resource languages, the supervised method delivers superior performance.

The semi supervised method combines the capabilities of supervised and unsupervised learning strategies. This setting is optimal for low-resource languages, provided the availability of databases from high- resource languages. The MFCC features extracted from labeled training data from high-resource languages are used to train a HMM to obtain phone labels, which are later used to train a MLP network with a bottleneck layer. The output of the bottleneck layer of MLP network are later cascaded with MFCC features to obtain a set of hybrid features. These hybrid features are employed for unsupervised training of GMM, and the GMM posterior features are used for template matching in STD. In this work, we have included labeled database from 3 high-resource Indian languages to build the MLP network and extract bottleneck features, and evaluate STD performance for 7 low-resource Indian languages. It is observed that the semi supervised STD performance of low-resource languages had improved up to 10comparison with unsupervised learning, while the performance of semi supervised STD with respect high-resource languages did not improve in comparison to supervised learning.

Contents

Declaration	ii
Approval Sheet	iii
Acknowledgements	iv
Abstract	vi
Nomenclature	viii
1 Introduction to Spoken Term Detection	1
1.1 Importance of Spoken Term Detection	1
1.2 Issues in Spoken Term Detection compared to Speech Recognition	2
1.3 Organization of Thesis	3
2 Literature survey	4
2.1 Unsupervised Method	4
2.1.1 GMM	4
2.2 Supervised Method	5
2.2.1 Word Index STD	5
2.2.2 Phone Index STD	5
2.2.3 Acoustic STD	6
2.3 Semisupervised Method	7
2.4 Back Ground of Speech Recognition	8
2.4.1 Speech Recognition Problem	8
2.4.2 Speech Recognition Process	9
3 Posterior Features Extraction	11
3.1 GMM Posteriors	11
3.2 MLP Posteriors	13
3.2.1 Introduction to Neural Network	14
3.2.2 Elementary Structure of Single Layer Neural Network	15
3.2.3 Multilayer Perceptron	15
3.2.4 Posterior Features Extraction using MLP	18
3.3 Hybrid MLP-GMM Posteriors	18
3.3.1 Bottle-neck Features	19
3.3.2 Use of phoneme Class Bottle-Neck Features for STD	19
3.3.3 STD using MLP-GMM posteriors	20

4	Matching Posterior Features for a STD	23
4.1	DTW and Variants	23
4.1.1	Classical DTW	23
4.1.2	Subsequence Dynamic Time Warping (SubDTW)	24
4.2	Different Cost Measure Function	25
5	Experimental Evaluation	28
5.1	Creation of the Database	28
5.2	Based on Unsupervised GMM Method	28
5.3	Based on Supervised MLP Method	29
5.3.1	Study on Different Size of Phoneme Classes	30
5.4	Based on Semisupervised MLP-GMM Method	31
6	Conclusion	33
	References	34

Chapter 1

Introduction to Spoken Term Detection

Human life is becoming more and more digital in nature, we interact with computers, mobile phones, televisions, etc. frequently and continuously. Accessing these equipments demand physical contacts with remote controls, mouse etc, which can be inconvenient at times. Further, on account of data deluge over the Internet, it often becomes cumbersome to find relevant information. In this and more importantly in a rural setting, dissemination of information via some public kiosk or any personal device, where the query can be submitted by merely speaking out a keyword might become a promising solution. In this case, STD technique assumes significance. STD is used to retrieve the occurrences of the user-spoken-term from the given speech database. This system searches the different words in the database, which is subset of the information retrieval. Thus controlling these digital devices by merely speaking out a keyword can be a promising solution. Such access control involves speech recognition and prompting corresponding actions.

The task of STD can be formulated as (i) detecting the presence of a spoken query, and (ii) locating the time-instants of its occurrence in a given test utterance. The task of STD involves matching the sequence of feature vectors extracted from the query word and the test utterance, and decide on the presence of the query word in the test utterance. The performance of a query word spotting system mainly depends on the choice of the feature vectors, and the strategy employed for matching the sequence of feature vectors. The feature vectors extracted from the speech signal should be independent of the speaker, and should carry phoneme specific information.

1.1 Importance of Spoken Term Detection

Applications of STD range from simple tasks, such as retrieving information from an existing database (traffic reports, document retrieval, train schedules), to interactive problem solving tasks involving complex planning and reasoning (travel planning, traffic routing), to support for multi-lingual and multimedia interactions. STD has diverse applications such as audio database indexing [1], routing telephone calls [2], telephone conversation monitoring, web audio search etc. STD provides a convenient way to interact with computers using speech, which is the most natural human mode of communication. These systems are capable of changing the way that people interact with

machines. Because STD will support human-machine interaction in a better way that requires no special training, these interfaces will also be available to many new groups of users (handicapped users, telephone users, hands-busy or eyes-busy users, users with a different native language). STD system has made rapid advances in the past decade, supported by progress in speech and language technology as well as in computing technology. As a result, there are now several research prototype STD that support limited interaction in domains such as travel planning, urban exploration, and office management. Depending upon the application, STD is classified into speaker dependent or speaker independent. If it is used for home application like opening your personnel computer and some security purpose, then we need speaker dependent feature. In this case the problem is simple because of isolated words, we do not need a robust algorithm. But if we need it for general purpose like if it is an application for speech recognition over a public telephone network where the speaker variability and the environment that speech passes through are different among different calls, we need a robust algorithm. If the recognition is for isolated keywords or phrases with respect to isolated keyword, then the problem is simple. A significant amount of work has been done related to speech recognition which matches isolated keywords with other keywords. But if the user wants to retrieve the keyword from continuous speech corpus, then the problem is difficult and a robust algorithm is required. If we go further to retrieve the entire query sentence from the speech corpus, it is even harder problem which is known as dictation. In this thesis we will concentrate on isolated STD on a continuous speech corpus using supervised technique. A spoken language system combines speech recognition, natural language processing and human interface technology. It functions by recognizing the spoken out words, interpreting the sequence of words to obtain a meaning in terms of the application, and providing an appropriate response back to the user.

1.2 Issues in Spoken Term Detection compared to Speech Recognition

Speech recognition is a complex system which maps speech signals into a string of words [3]. In standard automatic speech recognition (ASR) system, the recognizer tries to recognize all the input speech as a set of words in its vocabulary, i.e. all the words presented in the system should be the part of the known dictionary. In STD system, whole utterance need not be recognized, the recognizer is only concern about the occurrences of the particular keyword of interest. Therefore it ignores the extraneous speech, resulting a much less computation. In speech recognition, more weight is given to grammatically correct sequence. Also, this systems integrate the knowledge of linguistic and acoustic context into the speech recognition process. However, in case of STD, a single keyword has to be detected where the system can not have the contextual, grammatical, linguistic and lexical information, i.e. some words have similar sounds, such as 'there' and 'their', can be accurately spelled when the context of speech is unknown. Difference in pronunciation, accent and speaking way combines to make speech recognition problem more complex. When a single word pronounced in several ways, the system may not able to interpret what is being said correctly. The same happens when speaks faster or slower compared to a normal speaking. The most complex problem of speech recognition is identifying the context of word being spoken by the user. In case of independent STD system, the speaker variability haunts the performance, because the general features like MFCCs and perceptual linear prediction (PLP) exhibits speaker variability across the speakers. If the training

and testing data are from different database (one is recorded through microphone and another through TV-channels), then performance reduces because of channel mismatch problem.

In this thesis, only speaker independent STD problem is solved, i.e. the keyword should be retrieved from the given reference by any speaker. Thus it needs a set of speaker invariant acoustic features. To get stabilized feature a certain class of artificial neural network (ANN) i.e. MLP is applied to handled the spectral variability of the speech signal [4, 5]. A 3-layer neural network is applied in supervised manner to get the posterior probability as the output, which represents the speech frames. Next, sequence matching techniques like dynamic time warping (DTW) [6, 7] and subsequence dynamic time warping (subDTW) [8] are applied on the set of stabilized features to the best alignment between the query and reference. The final score is measured using P@N [9], where P@N is average precision for top N hits, where N is the number of times query occurs in reference utterance. This STD system is evaluated on Telugu news data collected from different news channels. A comparison study is done using a broad phoneme classes and then broad phoneme class is quantized into different phoneme classes depending upon the speech production. A classical phoneme recognition task is performed taking different phoneme classes.

1.3 Organization of Thesis

The rest of the thesis is structured as follows:

Chapter 2 provides the brief history of STD method that has been explored in the recent past years. In this chapter, the pros and cons of different method are discussed clearly. The problem associated with speech recognition and STD is also discussed.

Different feature stabilizing techniques such as Unsupervised technique using GMM ,Semisupervised technique using MLP-GMM and Supervised leaning technique using MLP is described for classification problem and how to use it for STD is discussed in **Chapter 3**.

Chapter 4 deals with the post-processing procedure after getting the required features from Chapter 3. Different type of sequence matching techniques are briefly explained.

Finally experimental setup and results are demonstrated in **Chapter 5**. The conclusion drawn from the above methods is discussed in **Chapter 6**.

Chapter 2

Literature survey

STD can be classified into three main different categories given as follows. The first one is “unsupervised”, the second is “Supervised” which requires label data for training process and third is “Semisupervised” used the capabilities of supervised and unsupervised learning strategies.

2.1 Unsupervised Method

The most advantage of using this method is very low resource by the cost of transcription of the training data, which is a time consuming as well as difficult task. As we are entering into digital media where each moment we handle a lot of data, how do we learn the system from this data without any supervised input. In this case the unsupervised method is a promising solution.

2.1.1 GMM

Recently, a posteriorgram-based template matching framework was proposed for query-by-example STD [10]. This framework represents speech segments by phoneme posteriorgrams, and matches query posteriorgrams with test posteriorgrams using the conventional dynamic time. This method has completely get rid of out of vocabulary (OOV), but the disadvantage is it needs an independently trained phoneme recognizer. In [11], Y. Zhang et. al. proposed an unsupervised approach for STD, using MFCCs as features to train the Gaussian mixture model (GMM). The posterior probabilities of the query utterance and reference utterance were calculated and a segmental dynamic time warping (SDTW) technique is used to compare the Gaussian posteriorgrams between keyword and test data. The keyword detection is done depending upon their relevant warping path score.

H. Wang et. al. proposed an unsupervised acoustics segment model (ASM) for unsupervised STD [12]. They use ASM to transcribe the speech and perform template matching on ASM posteriorgrams. This method has advantages over GMM method as ASM takes the temporal informations into considerations, which is ignored in case of GMM. So it can directly integrate the ASM into the existing phoneme-based frame work. The most crucial problem is that the STD performance is poor and inefficient because of unavailability of a supervisor. The more about GMM is explained in section 3.1.

2.2 Supervised Method

In supervised method, assumes the availability of a teacher or supervisor who classifies the training examples into classes and utilizes the given information for training the classes. The label of the speech utterance will be given and accordingly the supervised system will be trained. The performance of the system better as compared to the unsupervised method as it requires a teacher to train the system. In this thesis we concatenated on particular discriminative class of neural network known as MLP for supervised method for classification problem. The following are some supervised STD methods explained below, like Large Vocabulary Continuous Speech Recognition (Word Index STD), Indexing and searching sub-word content (Phone Index STD) and Acoustic STD.

2.2.1 Word Index STD

The most general approach used for STD is word index spotting of query words. Large Vocabulary Continuous Speech Recognizer (LVCSR) tries to recognize all the query words from the input speech. This STD task reduces to searching keyword in the word level hypothesis generated by LVCSR system [13]. The results could be in the form of 1-best hypothesis, n-best hypothesis list or word lattice. Accuracy of the STD system depends upon the accuracy of the speech recognizer. When the user submit a query word, an online search is performed through the word index. However the cost is very much because the word level indexing is done offline and when the spoken keyword is not present in database it is impossible to locate it using LVCSR system. Hence a recognizer dictionary has to be maintained and updated from time to time. Another drawback is that LVCSR do not include that a statistical language information to model the system. Large amount of domain specific text data is required to train the model.

2.2.2 Phone Index STD

In this case, phonetic content present in the speech is encoded in the form of words or subwords (phoneme or syllable) lattice. The phoneme sequence corresponding to the keyword to be searched is first obtained either from a predefined dictionary or by using a grapheme-to-phoneme(G2P) converter(used to generate the most probable phoneme list for a word not contained in the pronunciation dictionary (i.e. OOV words) used to create the G2P rules). Phone recognition accuracy is generally lower than word recognition due to lack of knowledge about correct phoneme sequences. To account for errors in phoneme recognition, an approximate search is run through the phoneme index. James et al. [14] have proposed a dynamic programming based lattice search technique, where the keyword phonemes were labeled as either ‘strong’ or ‘weak’. The strong phonemes are those which must be present in hypothesized lattice segment whereas weak phonemes could be deleted or substituted. In [15], the Phone-Lattice keyword Spotting (PLS) system achieves significantly faster query speeds than HMMs by first indexing speech files for subsequent rapid searching. For each speech file, a phoneme-lattice representation of the speech is generated by performing a N-best Viterbi recognition pass. The resulting phoneme-lattices compactly encode multiple observed phoneme sequence hypotheses for any particular region in the speech. In [16], an extension of PLS system was used known as Dynamic Match Phone-Lattice keyword Spotting (DMPLS), which uses Minimum Edit Distance (MED)[6] that calculates the minimum cost of transforming the source sequence to the target sequence using a combination of insertion, deletion, substitution and match operations, where

each operation has an associated cost. Each lattice phoneme sequence is scored against the keyword phoneme with a MED score. A threshold is kept on MED to compensate recognition errors. PLS is a special case of DMPLS where a threshold of 0 is used.

2.2.3 Acoustic STD

The presence of non-keywords or OOV words within the users speech has deteriorates effect on recognition performance. A good STD system aims to model in vocabulary (IV) and OOV in appropriate way, thus the performance accuracy becomes high. Rose and Paul [17] purpose a different approach to solve oov problem. Automatic speech recognizer (ASR) systems models a define sets of allowable words or subwords. One solution to solve OOV problem, is to build the system which contains all the possible words. However it is not possible practically. This model generates a large language model, which take a lot of effort and also expensive in terms of system resources. Another approach to solve this problem is restricting the words/subwords into small closed finite grammar, but this also feels unnatural to the user.

This OOV problem can be solved using filler model [18]. Generally filler model are used to absorb OOV words and allow the system to classify IV and OOV words. Here, they apply filler model to represent OOV word using K-mean algorithm. This k-mean algorithm provides more IV acceptance and OOV rejection rate. The computational complexity is also low.

Originally Dynamic programming (DP), a template matching technique used to match the keyword is introduced by Bridle [19]. These systems produce a score based template matching against every portion of the input. then a decision rule is kept to disambiguate the correctly decoded keywords from the false alarms. Some threshold is considered for separating the true keywords from false alarms. ASR systems need to model both keywords or in vocabulary (IV) words and OOVs and manage them in an appropriate way. In [20] Higgins and Wohlford, used filler templates to represent oov or non-keyword speech and in the next stage they used template based dynamic time warping connected speech recognition system to match a sequence of templates against an input utterance. The output of such a system is a continuous stream of keyword and filler templates, and the occurrence of a keyword template in this output stream is taken as a putative hit.

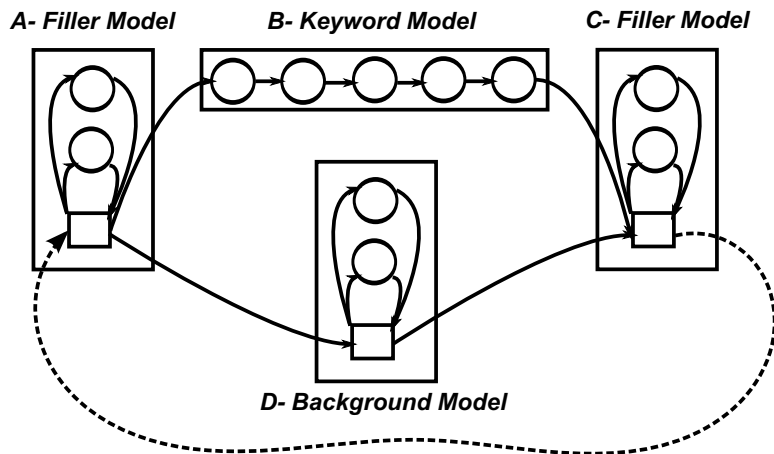


Figure 2.1: General acoustic model of STD

Rohlicek et. al. [21] have proposed an HMM keyword recognition systems, where they trained

the keywords using their known location. As the models for IV keywords and OOV keywords are kept parallel, both IV and OOV keywords compete with each other during recognition. A advantage of using an HMM representation of acoustic filler models, is that training of statistical HMMs allows acoustical filler models get more information over many different speakers and word contexts. Therefore, it is reasonable to assume that an HMM based filler model is better to model arbitrary speech than a template based system. Figure 2.1 shows a general recognition network for keyword detection. Parts denoted A and C are filler models (phoneme loops) which model OOV parts of utterance and Part B models IV parts created by concatenation of keyword phoneme models. Part D is the background model of the same part of utterance as the keyword model. Considering B and D model exactly the same part of the utterance, they calculated ratio of likelihoods A-D-C and A-B-C:

$$L_{Ratio} = L_{ABC}/L_{ADC} \quad (2.1)$$

where L_{ADC} is likelihood path through the model containing the non-keyword and L_{ABC} is likelihood of a path containing the keyword. If L_{Ratio} will approach 1, for the keyword and will low for non-keywords.

The advantage of Acoustic STD is speed of computation and on-line functionality. But the disadvantages are absence of good word-models. Simple language model leads to a lot of false alarms. The system can detect the keywords which are built in the recognition network. Once the new keyword came, the network has to rebuild again and considerably time consuming process. This method is under the category of a indirect STD method.

Indirect STD method: It is a two step method where the first step consists of the time consuming speech recognition and the second step is to search the spoken keyword. The method inherits main characteristics of the recognizer used. Input keyword must be converted to a sequence of units similar to recognizers output units (e.g. words, syllables, phones, etc.). Then the sequence is searched in the output of the recognizer. The recognizer (usually the slowest step of whole STD) is run only once. The STD is run each time a term or keyword has to be found. In comparison to the acoustic STD, the search is very fast because it is done over textual data (output of speech recognizer). Advantages of STD are the speed of search and detection accuracy (depends on recognizers accuracy).

Direct STD method: It is one step method, where the keyword will match with the reference by extracting some speaker independent features. This method is comparatively faster than the indirect method of STD. The method explained in this thesis is a direct method.

2.3 Semisupervised Method

Developing efficient speech processing systems for low-resource languages is an immensely challenging problem. One potentially effective approach to address the lack of resources for any particular language, is to employ data from multiple languages for building speech processing sub-systems. Semisupervised method is a kind of supervised method and techniques that also make use of unlabeled data for training typically a small amount of labeled data with a large amount of unlabeled data. The semi supervised method [48] combines the capabilities of supervised and unsupervised learning strategies. This setting is optimal for low-resource languages, provided the availability of databases from high- resource languages. The MFCC features extracted from labeled training data

from high-resource languages are used to train a HMM to obtain phone labels, which are later used to train a MLP network with a bottleneck layer. The output of the bottleneck layer of MLP network are later cascaded with MFCC features to obtain a set of hybrid features. These hybrid features are employed for unsupervised training of GMM, and the GMM posterior features are used for template matching in STD.

2.4 Back Ground of Speech Recognition

Speech signal is a quasi-periodic natural signal. The next section explains the basic problems associated with the speech recognition. After that speech recognition process is discussed in detail.

2.4.1 Speech Recognition Problem

Some of problem associated with ASR are listed below

- **Speaker Variability** - Different speaker have different voice quality because of their unique physic and personality. Even if the the speaker is same the voice is also vary. Some of these variation is given below.
 - *Realization* - If a same word is spoken over and over, the resulting speech would never look same even though the speaker tries to produce same, there will be some acoustic signal difference. The realization of speaker changes over time.
- **Speaking Style**- All the human speak differently according to their personality. They have unique way to pronounce and emphasize the word. Speaking style also varies according to the situation, where the speaker is producing the speech signal.e.g-with his friends, in front of parent, etc.
Speakers also communicate their emotion over the speech. The speaking style will be different, if they are happy, sad, disappointed, frustrated, etc.
- **The gender of the speaker** - Women has shorter vocal length compared to men. So the voices changes according to the gender of the speaker.
- **Anatomy of vocal track** - The shape and the size of the vocal track, size of the vocal cavity, the size of the lungs changes over time e.g-depending on age, health condition, etc.
- **Speed of Speech** - Some speakers are fast speaker and slow speaker, which vary the quality of the voice.
- **Regional and social Dialect** -
 - *Regional Dialect* - It involves the features of pronunciation which varies over geographical area.
 - *Social Dialect* - Social dialects are distinguished by features of pronunciation, vocabulary and grammar according to the social group of the speaker.

2.4.2 Speech Recognition Process

A general speech recognition system is shown in Figure 2.2. The features of the speech signal is calculated over 10 milliseconds and these features are used to calculate the most likely word candidate, making the constraints on acoustic, lexical and language models. Any random sequence of phonemes will not make a word, only meaningful phoneme sequence form a word and also depend upon the language in which it is trained. For these reasons model parameters of the system are determined by training datasets and is more biased towards the training datasets.

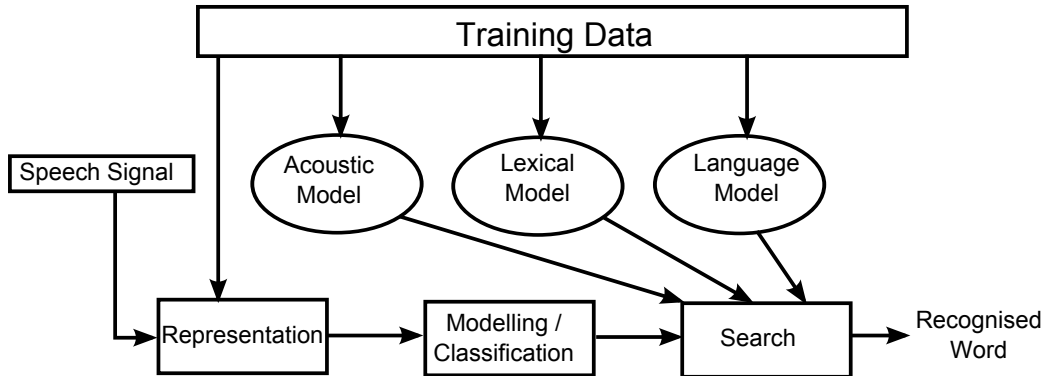


Figure 2.2: Component of a typical speech recognition system

Application of HMM to speech recognition: Quasi stationary nature of speech is captured by piecewise processing of speech signal each of duration typically 10 to 30ms, called a frame. There are many techniques which extracts useful features from speech. The Fourier analysis based MFCCs are taken as features of speech, which is based on source-filter theory for modeling vocal tract, which is assumed as filter whose source or excitation is airflow at glottis. There are several acoustic models for modeling speech. Due to its natural variability of speech the time domain techniques such as Dynamic Time Warping or other template matching techniques are not suitable to model speech sound units for large data. Hidden Markov model [22, 23, 24, 25], one among many speech modeling techniques is a statistical frame work that suites both acoustic and temporal modeling. For each file in the training set MFCCs are extracted from each frame of duration 10ms and corresponding transcriptions are taken and given as input to the HMM system. The HMM system used is developed by considering 3 states for each of the phoneme classes and two starting and ending dummy states and is left to right model. Here each state indicates the vocal tract configuration at a given time. Based on the assumption that each phoneme may take typically three transitions in the vocal tract shape during its utterance. More number of states can be taken at the cost of computations. The system initially computes a 39 dimensional mean vector and a 39×39 diagonal covariance matrix for whole training data and initializes the parameters of HMM for each and every phoneme. The parameters of HMM includes transition probabilities and initial probabilities of states and emission probabilities of each state. The emission probabilities of each state is modeled by parametric GMM [26] with weights, means and covariance matrices as parameters, as the gaussian mixtures have the power to approximate any arbitrary distribution with a probability density function. The number of mixtures can be chosen accordingly based on the amount of training data. Then the system finds all those sections of a phoneme in the training set and iteratively re-estimates the parameters

for that phoneme by Baum Welch algorithm and Expectation Maximization algorithm until the system converges and builds a HMM. Initially system is trained for single mixture. The initial big gaussian is sub divided into two mixture components whose means are the two other points shifted equidistant from original mean this is Expectation step and iteratively these means are modified so as to maximize our cost function that is log likely hood, and is Maximization step of Expectation Maximization algorithm. This gives a HMM with two gaussian mixtures. And each time it repeats the whole above process for an increment in the number of gaussian mixtures by a factor of two until chosen number of Gaussians are reached. This gives an individual HMM to each phoneme. Having the models for all phonemes now the system is tested with separate testing data set. HMM for word “cat” was given in Figure 2.3, where the phoneme ‘c’ , ‘a’ and ‘t’ are modeled as the model states. Each phoneme is divided into 3-states. e.g- ‘c’ is divided into ‘c₁’, ‘c₂’ and ‘c₃’. Detailed algorithm

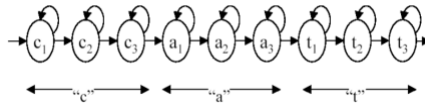


Figure 2.3: HMM model for keyword 'cat'

of HMM is given in [27].

In this chapter, a brief explanation is given about the different STD methods like supervised, unsupervised and semisupervised. Unsupervised method is convenient for low resource language model, where the labeling or the transcription of the language is not needed. But in case of supervised method, the labeling or the transcription of the language is required. The result will be better compared to unsupervised method by providing labeled data to train the supervised system. As a teacher is present in case of the supervised system the performance accuracy will be better. In Semisupervised the labeled data of multilingual language can use to build MLP in supervised manner and can use those models for low resource scenario.

For independent STD a stabilized set of feature is required, which is invariant to the speaker. Conventional features like MFCCs and PLP are less resistant to speaker variability [28]. So the required set of features which are invariant to the speaker can be extracted from the speech using GMM, MLP and Hybrid MLP-GMM which are the posterior probabilities and known as posterior features. The explained methods in this thesis are for getting speaker invariant features. The next chapter will give a brief introduction to GMM, ANN, MLP and posterior features.

Chapter 3

Posterior Features Extraction

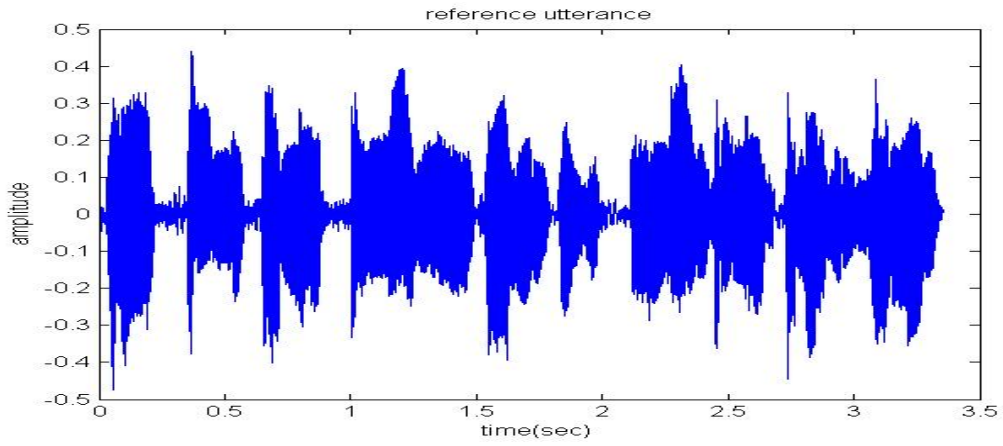
Posteriors are the posterior probability/posterior distribution over the defined class. These are the stabilized set of features which are not varying with speakers. The objective of this work is to match a keyword with same keyword present in the reference. The starting point itself, if it can be matched in waveform label, then no need of using any speech processing technique. Speech is a natural signal, which is the output of a dynamic vocal track system excited with a time varying input. Even if the different speakers produced same utterance, that will not match i.e. in waveform label the keyword will be different which is shown in the Figure 3.2 and Figure 3.3. This figure depicts the variability of the keyword in the waveform label. Figure 3.1 represent the waveform of the reference, Figure 3.2 represent the waveform of the keyword spoken by a different speaker and Figure 3.3 represent the waveform of the keyword spoken by the same speaker as the reference. In all the three waveforms are not matching in time domain

The next we will try to match in the frequency domain i.e. by using spectrogram. Spectrogram is used to represent the frequency component present in the speech signal. From the Figure 3.4 and Figure 3.5, it is clearly visualized that the keywords frequency components are not same for the keyword spoken by the same and the different speaker. Figure 3.4 represent the spectrogram of the keyword spoken by a different speaker and Figure 3.5 represent the spectrogram of the keyword spoken by the same speaker as the reference. So in both the time domain and the frequency domain, the keyword spoken by the same and the different speaker is not going to match. So in the next section we will adapt some speech processing technique to get rid of speaker variability and impose some technique to get a best match between the keywords irrespective of the speakers.

The rest of the chapter of the system describes about different method to extracts posterior features like using GMM, MLP and Hybrid MLP-GMM. A summary is given with the best method to extract the posterior features.

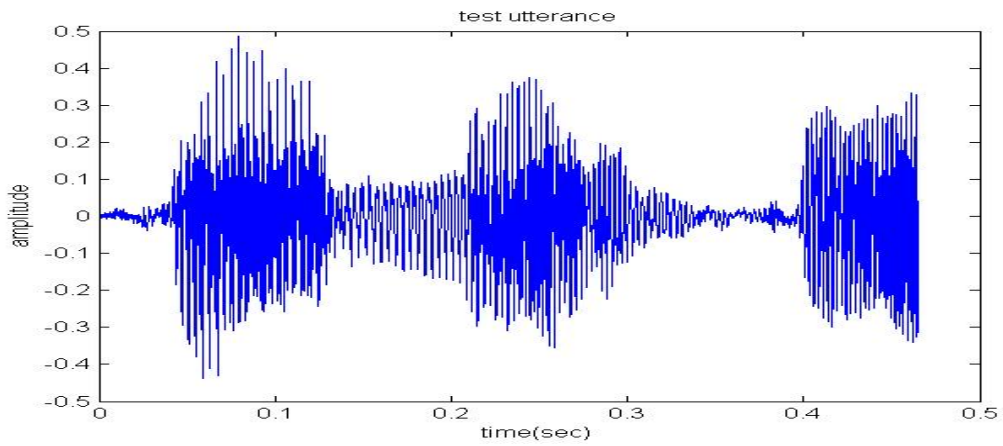
3.1 GMM Posteriors

Main objective of modelling techniques is to build a system which can discriminate different classes of inputs. In unsupervised approaches, this goal is accomplished without using labels. In the case of low resource languages, where less or no labelled data is available, unsupervised approaches can be a promising solution. In this study, generative model, namely GMM is studied for unsupervised



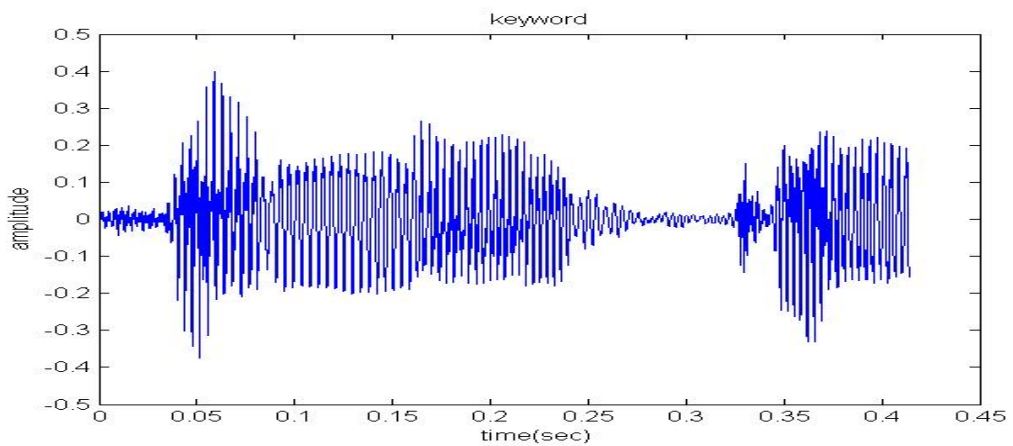
(a)

Figure 3.1: Waveform of the reference.



(a)

Figure 3.2: Waveform of the test query spoken by different speaker.



(a)

Figure 3.3: Waveform of the same keyword from reference.

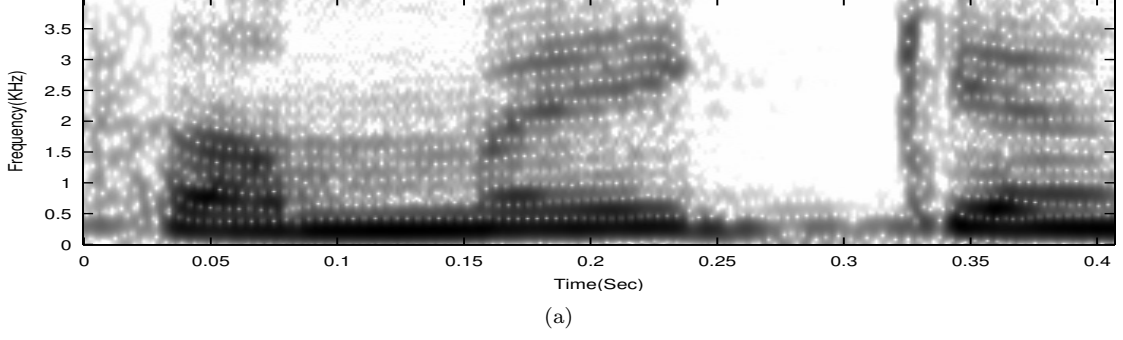


Figure 3.4: Spectrogram of the query word ‘Samaikhya’ spoken by same speaker

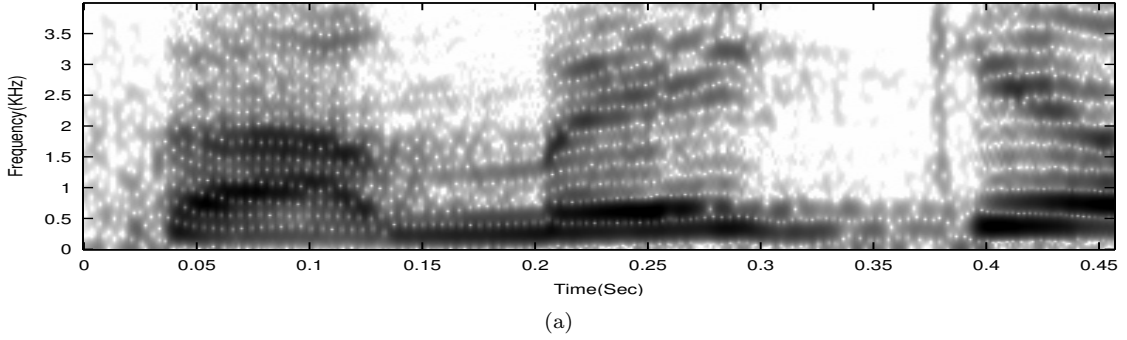


Figure 3.5: Spectrogram of the query word ‘Samaikhya’ spoken by different speaker

posterior feature extraction.

GMM will model the probability distribution of the data set as a weighted linear combination of Gaussian densities. That is, given a data set $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, the probability of data X drawn from GMM is

$$p(X) = \sum_{i=1}^M w_i \mathbf{N}(X/\mu_i, \Sigma_i),$$

where $\mathbf{N}(\cdot)$ is Gaussian distribution and M is number of mixtures. w_i , μ_i and Σ_i are weight, mean and covariance matrix of i^{th} Gaussian density respectively. GMM training estimates the parameter set $\theta_i = \{w_i, \mu_i, \Sigma_i\}$ for $i = 1, 2, \dots, M$ by maximizing the log-likelihood of the data X . Re-estimation is done by employing Expectation-Maximization (EM) algorithm. EM algorithm is guaranteed to give optimal solution [29] to represent the underlying probability distribution. Label information is not used for training GMM i.e., unsupervised training.

3.2 MLP Posteriors

In this method, the posterior feature are extracted using MLP. MLP has a very complex structure as compared to simple neural network, which helped it to solve complex problem. The next is given a brief introduction to neural network and it’s variant.

3.2.1 Introduction to Neural Network

The simplest definition of the Neural Network (also known as Artificial Neural Network), is provided by the neuro-computer Dr. Robert HechtNielsen [30]. He defines a neural network as: ‘a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs’. Artificial neural networks are generally presented as systems of interconnected ‘neurons’ which can compute values from inputs, and are capable of machine learning as well as pattern recognition as they are adaptable in nature. The key element of this paradigm is the novel structure of the information processing system. There was very enthusiastic research done on neural networks in fifties and sixties (Rosenblatts Perceptron), but it is forgotten for almost two decades and again it started at eighties. Basically a neural network has the components: a network, a activation function, and a learning rule

- *Network* is a connection of connection of set of nodes in direct connection or bilateral connection.
- *Activation function* is a local rule that can be followed by each nodes to compute the final output.
- *Learning rule* is a set of rules used to update the local weights to minimize the error.

ANN has different types of activation function i.e. Sigmoid function, linear function, Softmax function.

The ANN learning paradigms can be classified into following three types

- *Supervised* method assumes the availability of a teacher or supervisor to guide the training procedure. The output labels are provided and it has feedback loop to correctly classify the patterns.
- *Unsupervised* method has no supervisor to guide the training procedure. It has no prior information of class labels and classify the pattern class heuristically. Unsupervised learning aims at finding a certain kind of regularity in the data represented by the examples.
- *Reinforcement* learns through trial and error method. It is in between the supervised and unsupervised method. In this case system receives a feedback that tells the system whether its output response is right or wrong (reward/penalty assignment), but no information on what the right output should be is provided.

The application of ANN has divided broadly into 2 categories:

- Classification (Pattern recognition): The network tries to classify the input signal into predefined class.
- Regression (Prediction): It’s a problem of predicting the dependent variable from independent variable.

3.2.2 Elementary Structure of Single Layer Neural Network

In a neural network, a group of neurons connected together. The idealized model of neuron is much simpler than biological model. A neuron is a simple computational unit that receives signals from other units, sum them up and send out to output signal according to a simple thresholding function. The output of the j th node is denoted as y_j and is defined as

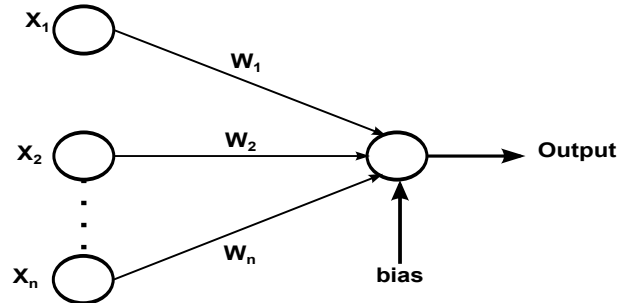


Figure 3.6: Single neuron

$$y_j = \Theta\left(\sum_{i=1}^n w_{ij}x_i\right) \quad (3.1)$$

Where x_i is the output from the input unit as shown in Figure. 3.6. The neuron has two modes of operation; the training mode and the testing mode. In the training mode, the neuron can be trained to fire (or not), for particular input patterns. In the testing mode, when a taught input pattern is detected at the input, its associated output becomes the current output. If the input pattern does not belong to the list of input patterns, the firing rule is used to determine whether to fire or not.

The perceptron and the adaptive filter using least mean square (LMS) algorithm are almost related. LMS algorithm is built around a linear neuron, where as a perceptron is built around a non-linear neuron, namely McCulloch-Pitts model of a neuron [31].

- **Problem associated with single layer perceptron and solution:**

The perceptron in the simplest form of neural network used for classification of patterns said to be linearly separable (i.e. pattern lies at the two sides of a hyperplane). The limitation of the peceptron built on a single neuron is, it can classify only two classes (hypothesis). By expanding output layer by more than two neurons, we may classify more than two layer. But the classes have to be linearly separable for perceptron to work properly.

3.2.3 Multilayer Perceptron

To achieve higher level of computational capabilities, a more complex structure of neural network is required. Multilayer perceptron is a most popular ANN architecture where the neurons are grouped into layers and forward connection exit [32]. This network contains a set of an input layer, one or more hidden layer and an output layer of computation nodes. Each layer consists of a set of nodes (neurons). The input signal propagates through input layer to output layer on layer-by-layer basis. MLP was stimulated by the advent of the Back-propagation (BP) algorithm [32] and able to solve some difficult and diverse problem using supervised learning method. BP or error back

propagation algorithm is a generalization of LMS filtering algorithm. The R-layer MLP network

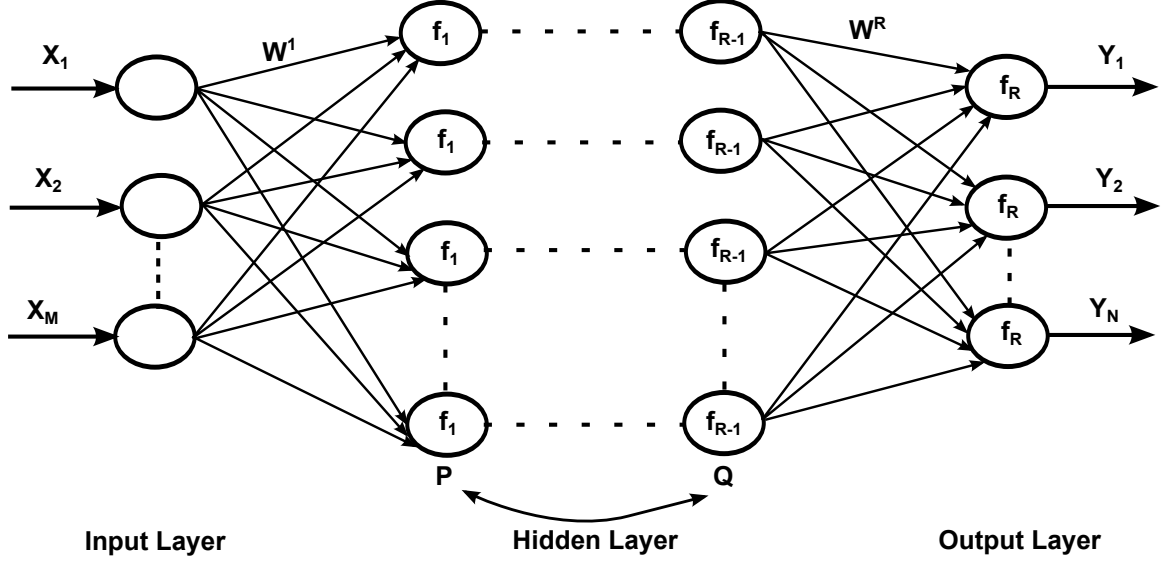


Figure 3.7: MLP Network

structure is shown in given in the Figure 3.7, where $[W^1 \dots W^R]$ is the weight matrix of the input to output layer respectively. X and Y are the input and output vectors respectively. M , N denote the number nodes at the input, output layer respectively. P and Q are the hidden layer with any number of nodes. f_1 , f_{R-1} and f_R denotes the activation function at P^{th} , $(R-1)^{th}$ and R^{th} layer respectively. Generally Sigmoid function is used as the activation function in hidden layer. If you want the outputs of a network to be labeled as posterior probabilities, it is highly desirable for those outputs to lie in between zero to one and sum to one. The purpose of the softmax activation function is to enforce these constraints on the outputs. Lets the input to each of the output nodes be R_i , $i = 1, 2, \dots, N$, where N is the number of different classes. Then the Softmax output function

$$S_i = \frac{e^{R_i}}{\sum_{j=1}^N e^{R_j}} \quad (3.2)$$

A MLP has the following distinct characteristics as compared to perceptron

- Each network includes a nonlinear activation function, which is differentiable, as opposed to perceptron (it has hard limiting function as its activation function).
- The network contains one or more hidden layer to learn more complex function by extracting more meaningful features from input pattern.
- Network has high degree of connectivity determined by the weights of the network.

Back propagation Algorithms: Back-propagation is a simple algorithm, which aims to minimize the error function using the weight space by using gradient descent. The outline of the BP algorithm is provided by Lippman [33] is as follows.

- *Step 1:* Initialize weights and offsets. Set all weights and node offsets to small random values.

- *Step 2*: Present input and desired output. The desired output is 1. The input could be new on each trial or samples from a training set.
- *Step 3*: Calculate actual outputs. Use the sigmoid nonlinearity functions or linear functions to calculate outputs depending upon the application.
- *Step 4*: Adapt weights.
- *Step 5*: Repeat by going to step 2.

The important point here to observe is the initial weights should be small as well as very large, which leads the network to a untrained stage. One disadvantage with this algorithm required large number of training set to converge. The convergence of the BP algorithm is given in [34][35][36]. Supervised learning paradigm solves many linear and non-linear problem such as classification, prediction, forecasting, etc.[37] and [38]. In [39] is shown that MLP with a single hidden layer and a non-linear activation function are universal classifier.

- **Points to speed-up the BP algorithm** The followings are some methods which could be taken care to speed-up the BP algorithm.
 - *Sequential verses Batch mode update*: Generally there are two ways to update the weights: batch training and sequential (also known as on-line or pattern based) training. On-line mode involves updating the value of weights after each pattern is submitted to the network. But in case of batch mode, the weights are updated after all the pattern submitted to the network. As long as the learning rate parameter is small, batch mode approximates gradient descent. On-line mode is not a simple approximation to gradient descent method [40]. The on-line mode has an advantage over batch mode, since it is a stochastic algorithm, it has the possibility of escape from local minima, but batch mode has pure gradient descent method has no chance to escape from local mimima. Further, the on-line mode is superior to batch mode if there is a high degree of redundancy in the training data, since, when using a large training dataset, the network will simply update weights more often in a given iteration, while a batch-mode network will simply take longer to evaluate a given iteration. An advantage of batch mode is that it can settle on a stable set of weight values, without wandering about training dataset.
 - *Initialization*: A good choice in choosing the initial weight and threshold value tremendously help in network design. If weights are initialized with high initial value, it is highly driven towards saturation. This process sluggish the speed of the learning process by taking very small value for local gradient descent in BP-algorithm. On the other way of the weights are assigned with very small value, then the BP algorithm may operate on the flat around the origin in their error surface. The origin is known as saddle point which refers to a stationary point where the curvature of the error surface is negative and the curvature along the saddle is positive. For these reasons the both the extreme pint are not applicable and initialization weights should be in between these points.
 - *Learning rates*: All the neurons should be learn at the same rate. The last layer usually have large local gradient than the previous layer. Hence, the learning rate parameter

should be assigned less value compared to last layer than the front layer. In [41], Lecun had suggested for a given neuron the learning rate should be inversely proportional to the square-root of synaptic connection made to that neuron.

3.2.4 Posterior Features Extraction using MLP

A word can be represented as a sequence of phonemes. Given a speech frame, posterior probabilities represent the posterior distribution over the defined class of phonemes. The sequence of frames for a speech utterance is defined as

$$F = [f_1, f_2, \dots, f_t, \dots, f_T] \quad (3.3)$$

and the sequence of posterior features are

$$G = [g_1, g_2, \dots, g_t, \dots, g_T] \quad (3.4)$$

Each posterior feature is represented by

$$g_t = [P(C_1|f_t), \dots, P(C_k|f_t), \dots, P(C_N|f_t)] \quad (3.5)$$

where $\{C_k\}_{k=1}^N$ represents set of phoneme class and $P(C_i|f_t)$ represents the posterior probability of i^{th} class given frame f_t . Each speech frame can be written as posterior vector of given phoneme class size. Thus, speech utterance containing T frames can be written as a $N \times T$ matrix, where each column represents the posterior probabilities of the corresponding frame.

To visualize the significance of posteriorgrams, the phoneme posteriorgrams for 25 phoneme classes are shown in Figure 3.8. Posteriorgram of reference is plotted in Figure 3.8(a) and the highlighted region shows the presence of query word. Figure 3.8(b) and 3.8(c) represent the posteriorgrams of query spoken by a different speaker and same speaker as that of the reference. It can be seen from this figure that, same phonemes are getting activated upon time frames irrespective of the speaker. Thus the posterior features were able to overcome the speaker dependency in MFCC features, and hence are more stable than MFCCs. Therefore we move further by considering posterior features to represent the speech frames.

3.3 Hybrid MLP-GMM Posteriors

Generation of phone posteriorgrams requires labelled data which would be difficult for languages with low resources. One solution is to build models from rich resource languages and use them in the low resource scenario. we use phoneme information and their derivatives such as bottle-neck (BN) features (also referred to as phoneme BN features) for STD. We obtain Gaussian posteriorgrams of phoneme BN features in tandem with the acoustic parameters such as mel-frequency cepstral coefficients (MFCCs) to perform the search.

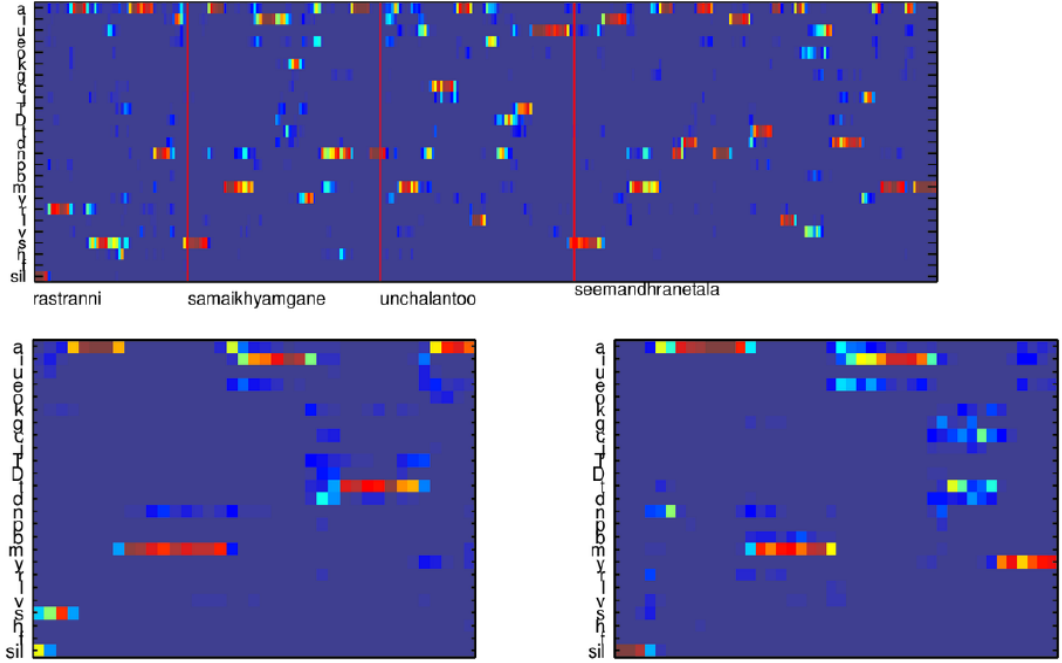


Figure 3.8: This figure illustrate stability of posterior representation. (a) Reference posteriorgram. Posteriorgrams of same query word spoken by (b) different speaker, (c) same speaker in a different context

3.3.1 Bottle-neck Features

Bottleneck features are generated from a multi-layer perceptron in which one of the internal layers has a small number of hidden units, relative to the size of the other layers. This small layer creates a constriction in the network that forces the information pertinent to classification into a low dimensional representation. The advantages of BN features are as follows [Gr ezl et al., 2007]: (a) they are compressed features and are of lower dimension, and (b) classification properties of the target class is reflected in the BN features.

3.3.2 Use of phoneme Class Bottle-Neck Features for STD

BN features can be obtained by post processing the phoneme posteriorgrams as follows (as shown in Figure. 3.8). A multi-layer perceptron (MLP) is trained by mapping each MFCC parameter to its corresponding phoneme class. From the MLP network, BN features [48] are derived (also referred to as phoneme BN features) to compute Gaussian posteriorgrams. we train phone MLPs using labelled database of 3 languages namely Marathi ,Manipuri and Gujarati (15 hours) consisting of 25 phones. MLP is trained to obtain 25 dimensional phone posteriorgrams using 39 dimensional MFCC features.

- **Feature Extraction:** The speech signal is invariant over a short range of 10-30ms. Here, each utterance is segmented into frames of 25ms duration with a shift of 10ms and P dimensional MFCC features are extracted by taking first P DCT coefficients.
- **Training of MLP:** The MLP is trained in supervised manner supplied by label of the phoneme. Doing manually, to get the labels/transcriptions in phoneme level is difficult and

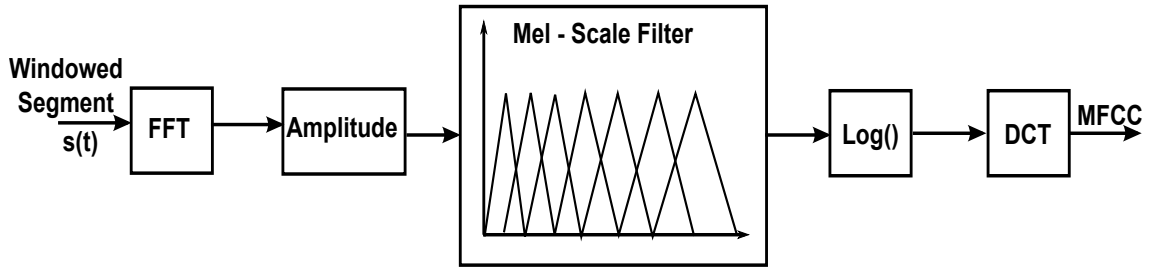


Figure 3.9: Block diagram to extract MFCC Feature

time consuming task. So for easy of making label in phoneme label, force-alignment process is used. Force-alignment is a process of labeling or aligning the phonemes of transcription data to speech. Using 25 phoneme classes, the force-alignment of the keyword is shown in Figure 3.10. The framework of MLP training used in this work is shown in Figure 3.11. MFCC vectors

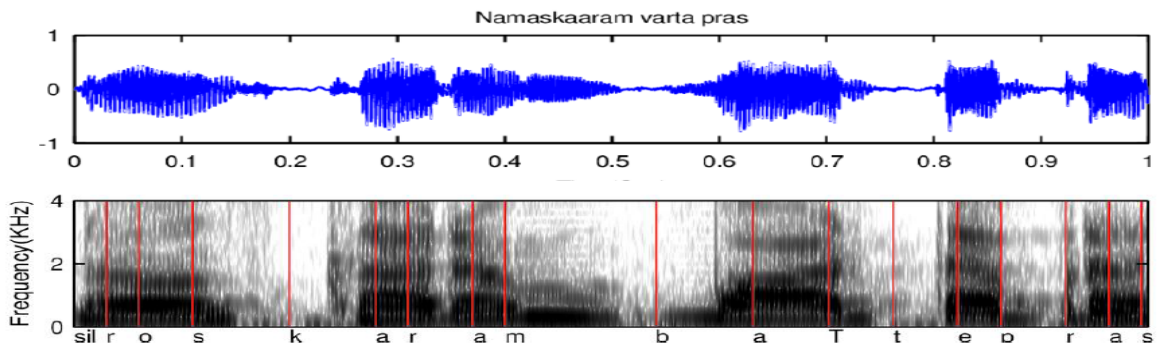


Figure 3.10: Force alignment for the keyword

are fed into MLP at the input layer and training is carried out in supervised manner. Back propagation (BP) algorithm is used to maximize the cross entropy function for minimizing the error at the output [32]. The MLP contains 5 layers, namely input, output and 3 hidden layers. The architecture used for training an phone MLP [48] is 39L 120N 13L 120N 25S. The integer values in the MLP architecture indicate the number of nodes, and L (linear), N (non-linear) represent the activation functions in each of the layers. We use 39 dimensional acoustic parameters as the input for the phone MLPs. Sigmoid function is used as the activation function in hidden layer. To get the posterior probability we use Softmax function at the output layer.

3.3.3 STD using MLP-GMM posteriors

In this study, a GMM is built by pooling MFCC feature vectors extracted from speech data collected from low resource languages in tandem with BN features extracted from trained MLP using high resource languages.

Posteriorgrams are generated from GMM, which we call GMM Posteriorgrams. The research in [28], shows that posterior-based features are more suitable for template matching compared to

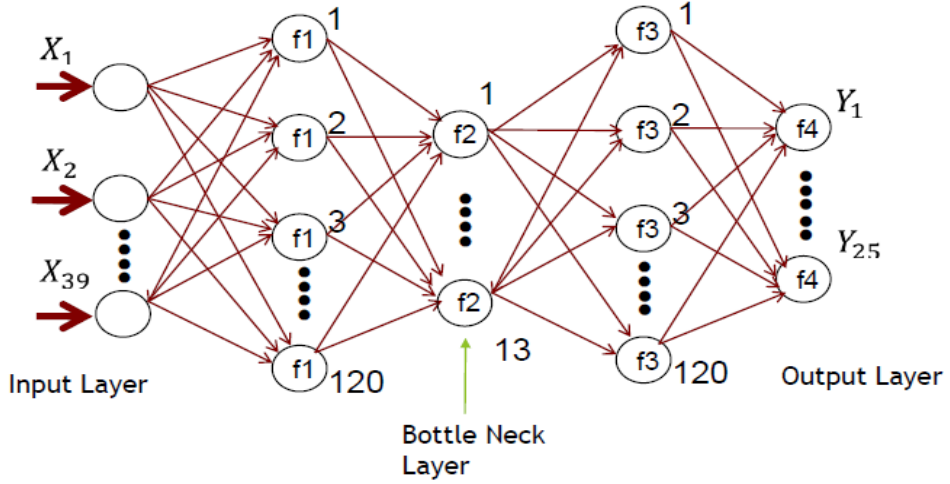


Figure 3.11: MLP Network with bottleneck layer.

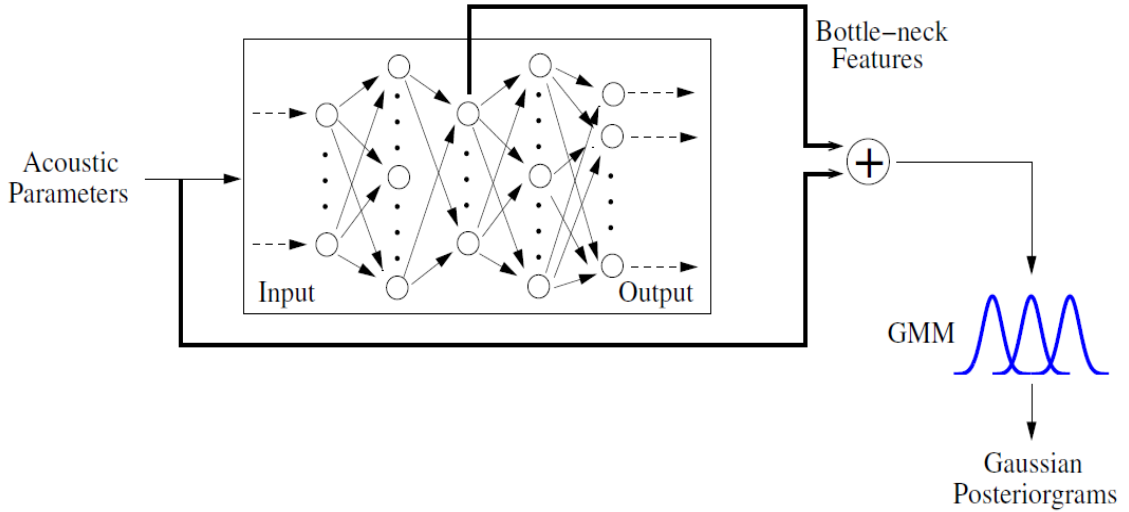


Figure 3.12: A general block diagram for computing Gaussian posteriorgrams of bottle-neck features in tandem with the acoustic parameters such as MFCC.

MFCCs. Posterior vector can be defined as the probability vector containing probabilities of frame belonging to each mixture. Mathematically, posterior vector P for a speech frame F is defined as

$$P = (P(G_1/F), P(G_2/F), P(G_3/F), \dots, P(G_M/F)),$$

where $P(G_i/F)$ is probability of frame F belonging to mixture G_i and it can be calculated as

$$P(G_i/F) = \frac{w_i \mathbf{N}(F/\mu_i, \Sigma_i)}{p(F)} \quad (3.6)$$

The numerator of Eq. 3.6 represents the responsibility of mixture G_i for the frame F and denominator is normalizing constant.

Our STD system is evaluated on 1 hour of reference data with 30 query words for every individual

language. Using this hybrid model, the reference and query utterances are represented as a sequence of Hybrid HMM-GMM posterior features. Speaker invariant nature of GMM posteriors is illustrated, in Fig. 4.4 and 4.5 using the task of searching for a query word in the reference utterance. We have considered two cases: the query word is from the same speaker as the reference utterance and the query word is from a different speaker. Euclidean distance and symmetric Kullback-Leibler Divergence (KL divergence) are used to find the distance between two MFCC and two posterior features, respectively. The distance matrices computed from MFCC features of reference and query utterances are shown in Fig. 4.2 and 4.3. The distance matrices computed from Hybrid HMM-GMM posterior features are shown in Fig. 4.4 and 4.5. In the case of matched speakers, there is a DTW path at correct location, indicating the presence of query word in the reference utterance, in distance matrices computed from both MFCC and posterior features. When the speakers do not match, the desired location of query word is successfully found in the distance matrix computed from the GMM posterior features, but not from the MFCC features. This case study depicts the speaker-invariant nature of GMM posterior features, and thereby their effectiveness in STD

In this chapter, it is shown that the posterior features are more stable than the conventional features. By looking into posteriorgram, it was concluded that same neurons are firing at same frame irrespective of the speakers. As MLP is a supervised method which is trained with transcription label information, the accuracy of the Hybrid MLP-GMM posteriors are more stable than GMM posteriors.

Chapter 4

Matching Posterior Features for a STD

After getting the posterior features, a matching technique is required for matching the posteriors. These posteriors can be considered as two time dependent sequences. There are some well-defined methods to match the time dependent sequence discussed below.

4.1 DTW and Variants

Dynamic time warping (DTW) algorithm is a well-known technique to find optimal alignment path between two time dependent sequences [6, 7, 42, 43]. Intuitively, the sequences are warped in a nonlinear fashion to match each other. DTW has been successfully applied in the fields of data mining and information retrieval for time-dependent sequences. In the next section, a brief explanation is given on the classical DTW and subDTW which is also a matching algorithm.

4.1.1 Classical DTW

The objective of the classical DTW algorithm is to compare two time-independent sequences $X := (x_1, x_2, \dots, x_N)$ of length $N \in \mathbb{N}$ and $Y := (y_1, y_2, \dots, y_M)$ of length $M \in \mathbb{N}$. For comparison of the two series, a local cost also known as local distance measure (similarity matrix) $C(x, y)$ is calculated. Typically $C(x, y)$ value is small if the similarity between x and y is more and vice-versa.

A warping path is a sequence $p = (p_1, \dots, p_L)$ with $p_i = (n, m) \in [1 : N] \times [1 : M]$ for $i \in [1 : L]$ satisfying the following three conditions.

1. *Boundary condition:* $p_1 = (1, 1)$ and $p_L = (N, M)$.
2. *Monotonicity condition:* $n_1 \leq n_2 \leq \dots \leq n_L$ and $m_1 \leq m_2 \leq \dots \leq m_L$
3. *Localization condition:* $p_{i+1} - p_i \in \{(1, 0), (0, 1), (1, 1)\}$ for $i \in [1 : L - 1]$.

An (N, M) -warping path $p = (p_1, \dots, p_L)$ defines an alignment between two sequences $X = (x_1, x_2, \dots, x_N)$ and $Y = (y_1, y_2, \dots, y_M)$ by assigning the element x_n of X to the element y_m of Y . The boundary condition enforces that the path should start from $(1, 1) \in (X, Y)$ and should reach $(n, m) \in (X, Y)$.

The monotonicity condition reflects the requirement of faithful timing: if an element in X precedes a second one this should also hold for the corresponding elements in Y , and vice versa. Finally, the localization condition expresses a kind of continuity condition: no element in X and Y can be omitted and there are no replications in the alignment.

The total cost $C_p(X, Y)$ of a warping path p between X and Y with respect to the local cost measure C is defined as

$$C_p(X, Y) := \sum_{l=1}^L C(x_{nl}, y_{ml}) \quad (4.1)$$

Further an optimal warping path between X and Y is having the minimum total cost among the all possible warping paths i.e p^*

$$C_{p^*}(X, Y) = \min\{C_p(X, Y) \mid p \text{ is an } (N, M) - \text{warping path}\} \quad (4.2)$$

To determine an optimal path p , one has to test every possible warping path between X and Y , which lead to a computational complexity, which can be reduced by using dynamic programming technique. The new term defined as accumulated matrix derived as

$$D(n, m) = \min\{D(n-1, m-1), D(n-1, m), D(n, m-1)\} + C(x_n, y_m) \text{ for } 1 < n \leq N \text{ and } 1 < m \leq M \quad (4.3)$$

Now, connect the path from $(1, 1) \in (X, Y)$ to the the point $(n, m) \in (X, Y)$ though the accumulated distance matrix, which will give optical warping path.

Finally the given explanation proofs that DTW is an efficient algorithm from two sequence matching. In this case the optimal path is only one which provides less distortion between the two sequences. It is applicable when the matching done a single keyword with another keywords i.e digit recognition. In case of Keyword spotting, the full sequence need not be matched with the other sequence, only the test query should match with the location of the reference where the query is present. In this case the length of the query is significantly small compared to the length of the reference. For this we need another technique which is explained in the next section.

4.1.2 Subsequence Dynamic Time Warping (SubDTW)

The objective of this technique is to match the two sequence with a significant difference in their length. So instead of matching globally with these sequences, one often has find a subsequence within the longer sequence that optimally fits the shorter sequence [8]. In this case assuming, the longer sequence is the given database is the reference and the lower shorter sequence is the test query.

Local weight: Additional weight vectors $(a, b, c) \in R^3$ are used to improve the alignment in vertical, horizontal and diagonal direction. The resulting accumulated matrix will be

$$D(n, m) = \min \begin{cases} D(n-1, m-1) + a.C(x_n, y_m) \\ D(n-1, m) + b.C(x_n, y_m) \\ D(n, m-1) + c.C(x_n, y_m) \end{cases}$$

Let $X := (x_1, x_2, \dots, x_N)$ and $Y := (y_1, y_2, \dots, y_M)$ be the two feature sequences, where the

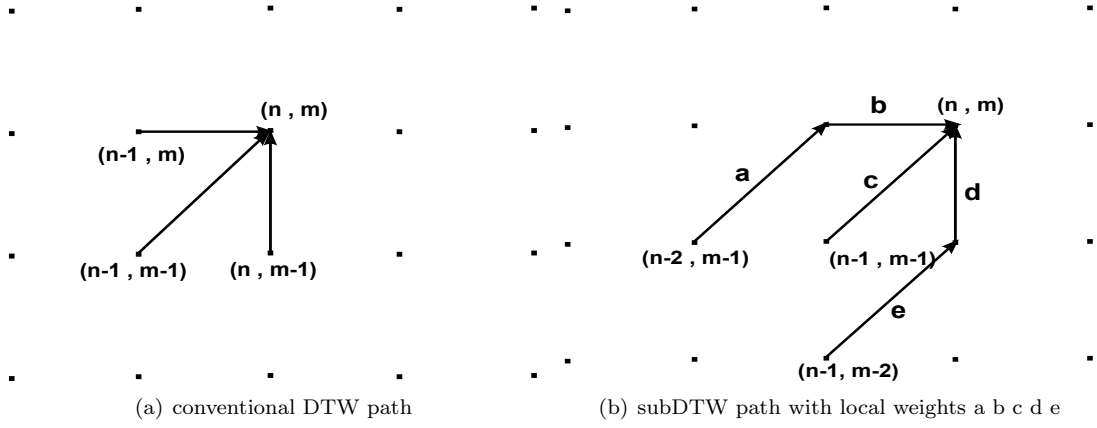


Figure 4.1: Path shown for different DTW

length M is much larger than the length N . It is the goal to find a subsequence $Y(a^* : b^*) := (y_{a^*}, y_{a^*+1}, \dots, y_{b^*})$ with $1 \leq a^* \leq b^* \leq M$ that minimizes the DTW distance to X over all possible subsequence of Y . Since the sequence X can start from anywhere in sequence Y we keep individual distances in the first row of accumulated distance matrix i.e., we do not accumulate distances along the first row as in the case of conventional DTW. The remaining values of D are defined recursively as in 4.1.1. The concept of SubDTW renders the requirement of STD system, where the best matching paths between query and reference utterances can be traced by backtracking from local minimum points along the last row of accumulated distance matrix. In Fig.4.1 (a) shows the path followed to calculate the accumulated matrix in case of classical DTW and Fig.4.1 (b) shows the path followed for the same in case of SubDTW with local weights a, b, c, d, e. These local weights can be adjusted to account for the incomparable durations of query and reference utterances.

4.2 Different Cost Measure Function

There is many different types of cost measure function, which are used to calculate the similarity matrix in DTW. But following are 3 distance measures are used in this work, namely Euclidean distance, negative logarithm of dot product and Kullback-Leibler divergence. Let T and Q are p^{th} test frame and q^{th} query frame. Then $(p, q)^{th}$ element in similarity matrix can be defined using different distance measures as

- **Euclidean Distance:** Euclidean distance measures the distance between two vectors on Euclidean space.

$$C_{ED}(p, q) = \sqrt{\sum_{k=1}^M (T(k) - Q(k))^2} \quad (4.4)$$

- **Negative Logarithm of Dot Product:** Geometrically, dot product of two vectors can be defined as the angle between them, mathematically

$$C_{DP}(p, q) = \sum_{k=1}^M T(k)Q(k) \quad (4.5)$$

Dot product is a similarity measure. To use it as a distance measure, we are applying negative logarithm on it. So, $(p, q)^{th}$ element can be calculated as

$$C_{LDP}(p, q) = -\log(D_{DP}(p, q)) \quad (4.6)$$

- **Kullback-Leibler Divergence:** Kullback-Leibler Divergence (KL Divergence) is a measure of distance between two probability distributions. In mathematical terms, KL Divergence from Q to T is defined to be

$$C_{KL}(T||Q) = \sum_{k=1}^M \ln \left(\frac{T(k)}{Q(k)} \right) T(k) \quad (4.7)$$

This distance metric is not symmetric. To make it symmetric we took sum of KL divergence between T, Q and Q, T as our distance measure. So, the modified KL divergence is

$$C_{KL}(p, q) = \sum_{k=1}^M \ln \left(\frac{T(k)}{Q(k)} \right) T(k) + \sum_{k=1}^M \ln \left(\frac{Q(k)}{T(k)} \right) Q(k) \quad (4.8)$$

We can interpret each vector in posteriorgram as a probability distribution. So, we can use this metric as our local distance measure. If any element in posterior vector P is zero then KL divergence with any other vector is infinity, which is not true. To avoid it, we use smoothing method suggested by [9]. So, the new posterior vector P_{new} is

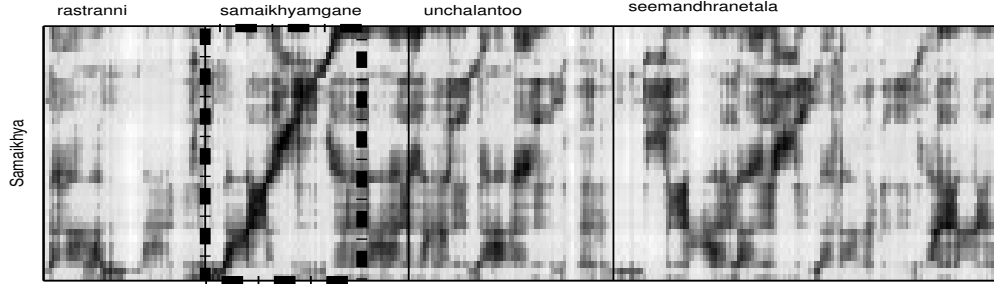


Figure 4.2: Similarity matrices using MFCC as feature vector. Query spoken by same speaker

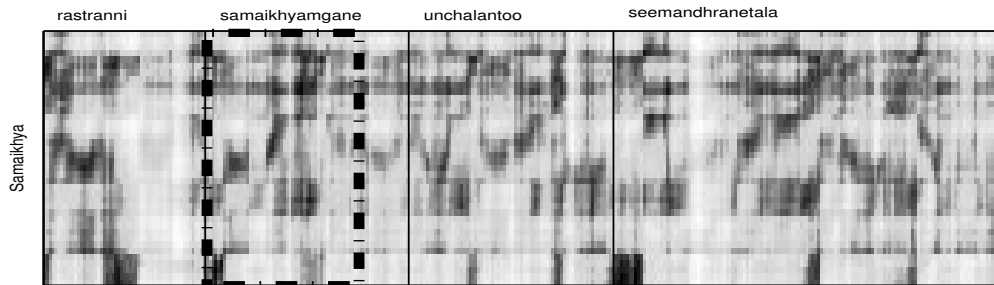


Figure 4.3: Similarity matrices using MFCC as feature vector. Query spoken by different speaker

$$P_{new} = (1 - \lambda)P + \lambda\mathcal{U} \quad (4.9)$$

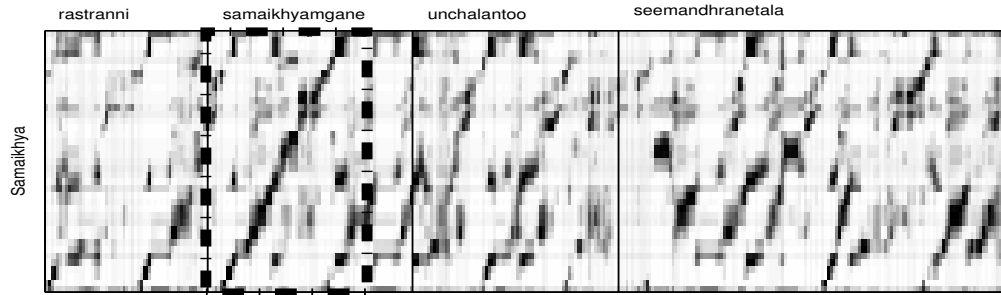


Figure 4.4: Similarity matrices using posterior probability as feature vector. Query spoken by same speaker

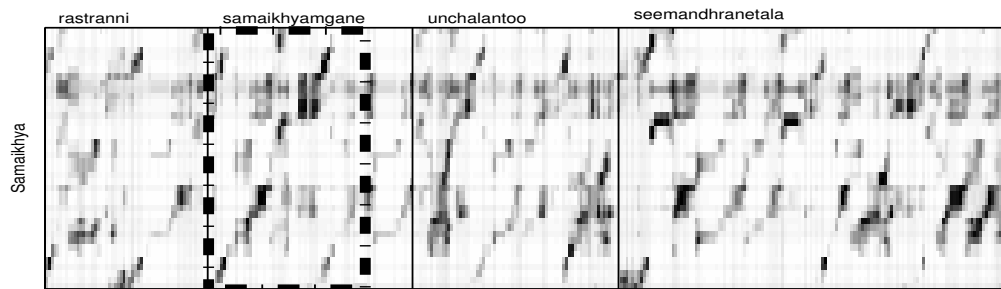


Figure 4.5: Similarity matrices using posterior probability as feature vector. Query spoken by different speaker

where \mathcal{U} is uniform distribution. This symmetric KL-divergence has been successfully applied to calculate the distance between two probability distribution functions [44]. Also the author of [45], shown that the KL Divergence provides better performance compare to other cost measure function in case of STD application.

Figure 4.2 illustrates the similarity matrix based on MFCC features obtained between reference and query utterances spoken by same speaker and Figure 4.3 illustrates the same for different speakers. Black color indicates more similarity and white indicates less similarity. It can be seen from Figure 4.2 that the match between the query and the segment of reference utterance is clearly visible. However in Figure 4.3 this match is not noticeable because of the speaker variability.

Similarity matrices using posterior probabilities as input vectors are shown in Figure 4.4, in cases when query and reference utterances spoken by same and different speakers. An unambiguous DTW path can be observed in both Figure 4.4 and Figure 4.5 at the marked area. This depicts the effectiveness of posterior features in bringing out the speaker independent acoustic information from utterances, thereby delivering better STD performance.

This chapter provides a detailed explanation about the sequence matching techniques and its variants. After getting the system specification the next chapter discusses the experimental evaluation.

Chapter 5

Experimental Evaluation

The speech signal is quasi-periodic in nature. Thus it is invariant over a short range of 10-30ms period. Speech signal is sampled at 16KHz sampling frequency. Here, each utterance is segmented into frames of 25 ms duration with a shift of 10ms and 39 dimensional MFCC vectors (13 cepstra + 13 Δ + 13 $\Delta\Delta$) are extracted from each frame. Training of MLP is carried out using supervised method. 39 dimensional MFCC vectors (13 cepstra + 13 Δ + 13 $\Delta\Delta$) are extracted from each frame. The architecture used for training an phone MLP is 39L 120N 13L 120N 25S. The integer values in the MLP architecture indicate the number of nodes, and L (linear), N (non-linear) represent the activation functions in each of the layers. We use 39 dimensional acoustic parameters as the input for the phone MLPs. In the MLP 25 phoneme classes are considered on basics of speech production mechanism. The posterior features are extracted taking softmax function at the output layer and the dimension of the posterior feature depend upon the number of phoneme classes used for classification. HTK toolkit [46] and quicknets [47] are used in this work.

5.1 Creation of the Database

The system is evaluated on 10 indian languages. The total available data for every individual language is divided into two subsets, training set and testing set. The training dataset includes 3500 utterance of $3\frac{1}{2}$ hours read by multiple speakers. The testing dataset consists of 1 hour news data from different speakers. The training and testing dataset are constituted by distinct speakers. This STD system is tested on 10 indian languages database and evaluated on GMM,MLP as well as on Hybrid MLP-GMM. There is significant improvement in the supervised method compared to unsupervised method and semi-supervised method.It is observed that the semi supervised KWS performance of low-resource languages had improved up to 10 percent in comparison with unsupervised learning, while the performance of semi supervised KWS with respect high-resource languages did not improve in comparison to supervised learning.

5.2 Based on Unsupervised GMM Method

GMM posterior features are obtained by providing MFCC feature vectors as input to the trained GMM. The number of mixtures can be chosen accordingly based on the amount of training data.

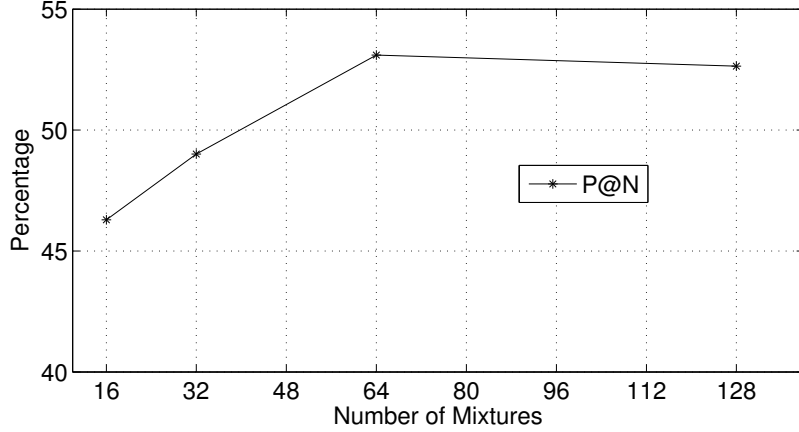


Figure 5.1: Experiment with number of mixtures for telugu language

Table 5.1: Results of query spotting for telugu language using MFCC and using GMM posteriorgrams with 64-mixtures GMM

Metric	MFCC		GMM-64	
	Euclidean Distance (%)	Euclidean Distance (%)	Dot Product (%)	KL Divergence (%)
P@N	45.68	42.89	51.89	53.10
P@2N	54.91	50.68	63.08	63.69
P@3N	60.81	55.67	67.77	67.47
P@4N	63.23	58.54	71.55	70.34
P@5N	64.29	60.96	73.67	73.67

Effect of number of mixtures in a GMM on query word spotting is analyzed for one language i.e telugu in Figure 5.1 with KL divergence as similarity measure. Low P@N for 16-mixtures GMM can be justified as high sensitivity of posteriorgrams to small changes in data. As the number of mixtures increased the performance of the system improved. We adopted 64-mixtures GMM for all experiments as P@N is high. Decrement in P@N for 128-mixtures GMM can be attributed to association of each phoneme class with more than one mixture.

Importance of local distance measure used in subDTW for different input features is tabulated in Table 5.1. For this experiment 64-mixtures are used. It is observed that for euclidean distance measure, MFCC features are better than GMM posteriors. However, when negative logarithm of dot product and KL divergence are used as local distance measures combined with GMM posteriorgrams, our query spotting system performance is improved by 9% and 10.21% respectively compared to euclidean distance measure. This concludes KL divergence is the better method to calculate the local cost measure for STD.

Table 5.1 also shows results for different values of k . As k increases, more number of query words can be retrieved. We can recover all instances of a query using GMM posteriorgrams for larger value of k .

5.3 Based on Supervised MLP Method

MLP posterior features are extracted from the trained MLP. Once the parameter of the MLP are drawn, then posterior features can be easily extracted by providing MFCC feature vectors as the input to the MLP. The next it is discussed about how STD performance affected by taking different

Table 5.2: Different phoneme classes used in this work

Class	Phoneme(IPA) (%)
Vowels	a, a:, e, e:, o, o:, i, i:, u, u:
Stop consonants	k, k ^h , g, g ^h , t, t ^h , d, d ^h , p, p ^h , b, b ^h , d, d ^h
Fricatives	h, s, s, f, f
Nasals	m, n
Affricates	z, c, c ^h , j ^h , J ^h , ʒ, ʃ
Glides	j, v
Trill	r
Liquid	l
Silence	sil

Table 5.3: Grouping of the phonemes into different classes

45 classes	a	a:	i	i:	j	u	u:	e	e:	o	o:	v	f	s	h	f	m	n	k	k ^h	g	g ^h	c	c ^h	z	j ^h	j ^h	ʒ	ʃ	t	t ^h	d	d ^h	t	t ^h	d	d ^h	p	p ^h	b	b ^h	r	l	sil
25 classes	a	i	j	u	e	o	v	s	h	f	m	n	k	g	c	j	t	d	t	d	p	b	r	l	sil																			
15 classes	a	i	u	e	o	F(Fricatives)		N(Nasal)		G(Glottal)		P(Palatal)		R(Retroflex)		D(Dental)		B(Bilabial)		r	l	sil																						
6 classes	V(Vowel)						F(Fricatives)		N(Nasal)		C(Consonants)												T(Trill and Liquid)		sil(Silence)																			

number of phoneme classes.

5.3.1 Study on Different Size of Phoneme Classes

The number of phonemes varies from 20 to 50 for most of the languages. For training this network, we investigated the usage of different number of phonemes for one language i.e telugu at the output layer. Initially, 45 classes were used for training MLP as shown in Table 5.2. As the phoneme recognition varies over different grouping of phoneme classes, the keyword performance also varies over the different phoneme classes. For comparative study, these 45 classes are quantized depending upon the manner and the place of articulation into 25 classes, 15 classes and 6 classes as shown in Table 5.3.

For phoneme recognition we used both HMM and MLP and the results are shown in Table 5.4. Corresponding to an increase in number of phoneme classes, the nodes at the output layer of MLP increases and this results in a notable drop in phoneme recognition accuracy. Table 5.5 shows P@N by considering different phoneme classes for telugu language.

- *45 phoneme classes* : The 45-phoneme class is shown first row of the Table 5.3. Here all the available phonemes are considered as a different class. As the output dimension of the MLP in classification problem is more compared other phoneme classes, it resulted with less phoneme accuracy.
- *25 phoneme classes* : The 25-phoneme class is shown second row of the Table 5.3. The grouping

Table 5.4: Recognition accuracy for using different number of phoneme classes

Phoneme Classes	HMM	ANN
6	77.88	81.87
15	70.6	76.02
25	69.51	74.24
45	62.68	69.11

Table 5.5: Average performance using subsequence DTW by taking different phoneme classes

Performance measure	6 classes	15 classes	25 classes	45 classes	raw MFCCs
P@N	44.05	77.65	80.13	72.36	45.68
P@2N	55.68	86.50	89.13	80.57	54.91
P@3N	59.17	88.10	90.75	82.39	60.81
P@4N	61.19	88.71	91.28	83.10	63.23
P@5N	62.13	88.99	91.61	83.41	64.29

are done using the basics of the speech production mechanism. The phoneme accuracy is improved compared to 45 phoneme classes.

- *15 phoneme classes* : The 25-phoneme class is shown third row of the Table 5.3. In this case phoneme classes are formed combining a broad phonemes like all the fricatives are grouped into single fricative. The phoneme accuracy is improved compared to 25 and 45 phoneme classes.
- *6 phoneme classes* : The 6-phoneme class is shown fourth row of the Table 5.3. This is the smallest phoneme class size we made by taking all vowels as a single vowel, all the consonants into a single consonant, trills and liquids into single class trill. Here vowel, consonants trills, fricative, nasals and sil are the six phonemes used in this class. The phoneme accuracy is improved compared to other phoneme classes.

We observed that there is significant decrease in P@N from 15 to 6 classes. As more information lies in vowels compared to consonants and in case of 6 classes we grouped all the vowels into one class caused this reduction in performance. Segregation of the different phoneme classes into further sub-classes helps to bring up phoneme accuracy index. The number of these sub-classes can be anything from 2 to 45. However even by dividing the phoneme classes into merely 2 sub-classes namely voiced and unvoiced, does not assure an increase in STD performance. Concluding 25 phoneme classes giving better results, further analysis was carried out only for this class for all remaining languages.

5.4 Based on Semisupervised MLP-GMM Method

We use a three step process to generate the features in this method for STD: (a) Extracting speech parameters such as MFCC (b) Train a phone MLP and extract the bottle-neck features for each of the speech parameters, and (c) Compute Gaussian posteriors using speech parameters in combination with the derived BN features. In this method first we trained phone MLP by using 3 multilingual languages those are Gujarati, Marathi and Manipuri which are having labelled data and then we used that model for remaining 7 low resource languages. The results for all languages using these supervised, Unsupervised and Semisupervised methods are shown in the Table 5.7. Phoneme recognition accuracy for 10 languages using HMM and ANN are shown in Table 5.6. Here mean precision (P@kN) is used as evaluation metric, where N is number of instances of query word occurring in test utterance and k is a positive integer. P@N defined as the fraction of queries recovered from test data in top N hits. Evaluation was done using P@N, where P@N is average precision for top N hits, where N is the number of times query occurs in reference utterance. It is considered to be a hit, if the system proposed location matches with reference location by more than 50% [45]. The system which produces higher value of P@N, is considered as best system.

language	HMM	ANN
Bengali	50.66	59.35
Gujarai	64.10	66.93
Hindi	50.43	55.30
Malayalam	33.48	38.19
Manipuri	63.37	69.18
Marathi	51.97	60.27
Odia	67.79	75.78
Punjabi	72.46	73.89
Telugu	65.31	70.13
Urdu	56.40	60.47

Table 5.6: Recognition accuracy for 10 languages using HMM and ANN

Language	Techniques	P@N	P@2N	P@3N	P@4N
Bengali	Unsupervised	27.88	29.65	31.42	32.30
	Semisupervised	36.28	36.73	39.82	40.71
	Supervised	69.03	76.55	80.09	83.63
Gujarathi	Unsupervised	43.03	51.76	58.32	64.03
	Semisupervised	44.71	56.64	62.18	66.39
	Supervised	68.91	76.97	81.01	83.70
Hindi	Unsupervised	37.82	51.92	57.69	61.54
	Semisupervised	49.36	62.18	66.03	66.67
	Supervised	60.66	63.94	66.29	69.11
Malayalam	Unsupervised	14.78	26.10	32.79	36.95
	Semisupervised	21.25	34.41	42.03	48.96
	Supervised	57.66	67.79	71.85	72.52
Manipuri	Unsupervised	21.67	29.44	33.33	38.33
	Semisupervised	25.56	34.44	43.33	50.56
	Supervised	51.45	61.85	67.63	72.83
Marathi	Unsupervised	36.60	54.95	65.26	73.51
	Semisupervised	44.74	58.14	68.45	75.98
	Supervised	76.17	81.78	83.09	84.04
Odia	Unsupervised	48.58	64.52	70.97	77.80
	Semisupervised	53.32	68.31	76.09	82.35
	Supervised	74.95	87.90	90.45	93.42
Punjabi	Unsupervised	58.70	70.65	77.17	81.52
	Semisupervised	60.87	70.65	73.91	75.00
	Supervised	68.26	76.96	80.22	82.39
telugu	Unsupervised	44.07	59.27	63.83	68.69
	Semisupervised	53.80	67.78	73.56	77.81
	Supervised	72.34	88.15	91.79	93.31
Urdu	Unsupervised	57.59	67.78	72.04	74.44
	Semisupervised	58.15	69.07	73.89	76.67
	Supervised	64.89	75.22	79.02	82.28

Table 5.7: Average performance for all languages using three different approaches

Chapter 6

Conclusion

This thesis deals with spoken term detection. Several STD techniques are explained and tested on the 10 different Indian language databases. This thesis aimed at the improvement of the performance accuracy compared to the existing methods.

An analysis of supervised learning technique based on a discriminative classifier- the multilayer perceptron (MLP) for the task of spoken term detection (STD) is presented here. A study on the use of stabilized features containing acoustic variability across sounds was carried out and it was noted that conventional MFCCs contain speaker variabilities together with acoustic variabilities. Hence the posterior features obtained by training MLP were chosen, as they were performing well in nullifying speaker variabilities for the task of STD. Also it was noted that in comparison to HMM, phoneme recognition accuracy was also improved using MLP.

The posterior features for query and reference utterances were matched for STD using subDTW, as they outperformed classical DTW in terms of computational as well as time complexity. Experimental evaluation of this method was carried out with respect to Telugu News bulletin database. An idea is given how the cost measure function affects the subDTW performance. GMM provides an increment in accuracy of 7.42% compared to the conventional MFCC features. In comparison to unsupervised methods like GMM, the supervised method HMM performed better, resulting in a 27% improvement. This subDTW method had delivered improved STD accuracy in comparison with conventional MFCCs.

The semi-supervised method combines the capabilities of supervised MLP and unsupervised GMM learning strategies. This setting is optimal for low-resource languages, provided the availability of databases from high-resource languages. In this work, we have included labeled databases from 3 high-resource Indian languages to build the MLP network and extract bottleneck features, and evaluate KWS performance for 7 low-resource Indian languages. It is observed that the semi-supervised KWS performance of low-resource languages had improved up to 10% in comparison with unsupervised learning, while the performance of semi-supervised KWS with respect to high-resource languages did not improve in comparison to supervised learning.

References

- [1] J. Foote. An overview of audio information retrieval. *Multimedia Systems* 7, (1999) 2–10.
- [2] H. Bahi and N. Benati. A new keyword spotting approach. In *Multimedia Computing and Systems, 2009. ICMCS '09. International Conference on. 2009* 77–80.
- [3] D. Jurafsky and H. James. *Speech and language processing an introduction to natural language processing, computational linguistics, and speech* .
- [4] M. R. Azimi-Sadjadi, S. Citrin, and S. Sheedvash. Supervised learning process of multi-layer perceptron neural networks using fast least squares. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on. IEEE, 1990* 1381–1384.
- [5] Y. Arriola and R. Carrasco. Integration of multi-layer perceptron and Markov models for automatic speech recognition. In *UK IT 1990 Conference. IET, 1990* 413–420.
- [6] D. Jurafsky and J. H. Martin. *An introduction to natural language processing, computational linguistics, and speech recognition 2000*.
- [7] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on* 26, (1978) 43–49.
- [8] M. Müller. *Information retrieval for music and motion, volume 2*. Springer, 2007.
- [9] T. J. Hazen, W. Shen, and C. White. Query-by-example spoken term detection using phonetic posteriorgram templates. In *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on. IEEE, 2009* 421–426.
- [10] T. Hazen, W. Shen, and C. White. Query-by-example spoken term detection using phonetic posteriorgram templates. In *Automatic Speech Recognition Understanding, 2009. ASRU 2009. IEEE Workshop on. 2009* 421–426.
- [11] Y. Zhang and J. Glass. Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams. In *Automatic Speech Recognition Understanding, 2009. ASRU 2009. IEEE Workshop on. 2009* 398–403.
- [12] H. Wang, T. Lee, and C.-C. Leung. Unsupervised spoken term detection with acoustic segment model. In *Speech Database and Assessments (Oriental COCODA), 2011 International Conference on. 2011* 106–111.

- [13] I. Szöke, P. Schwarz, P. Matejka, L. Burget, M. Karafiát, M. Fapso, and J. Cernocký. Comparison of keyword spotting approaches for informal continuous speech. In *Interspeech. 2005* 633–636.
- [14] D. A. James and S. J. Young. A fast lattice-based approach to vocabulary independent wordspotting. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, volume 1. IEEE, 1994 1–377.
- [15] S. J. Young, M. Brown, J. T. Foote, G. J. Jones, and K. Sparck Jones. Acoustic indexing for multimedia retrieval and browsing. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 1. IEEE, 1997 199–202.
- [16] K. Thambiratnam and S. Sridharan. Dynamic Match Phone-Lattice Searches For Very Fast And Accurate Unrestricted Vocabulary Keyword Spotting. In *ICASSP (1). 2005* 465–468.
- [17] R. C. Rose and D. B. Paul. A hidden Markov model based keyword recognition system. In *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE, 1990 129–132.
- [18] M. E. Dunnachie, P. W. Shields, D. H. Crawford, and M. Davies. Filler Models for Automatic Speech Recognition Created from Hidden Markov Models Using the K-Means Algorithm. *Proceedings EUSIPCO 2009, Glasgow, UK, August 24-28* .
- [19] J. S. Bridle. An efficient elastic-template method for detecting given words in running speech. In *Brit. Acoust. Soc. Meeting*. 1973 1–4.
- [20] A. Higgins and R. Wohlford. Keyword recognition using template concatenation. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85., volume 10*. 1985 1233–1236.
- [21] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish. Continuous hidden Markov modeling for speaker-independent word spotting. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*. IEEE, 1989 627–630.
- [22] H. Bourlard and N. Morgan. A continuous speech recognition system embedding MLP into HMM. In *Advances in neural information processing systems*. 1990 186–193.
- [23] S. Furui. *Digital speech processing: synthesis, and recognition*. CRC Press, 2000.
- [24] X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov models for speech recognition*, volume 2004. Edinburgh university press Edinburgh, 1990.
- [25] L. R. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*, volume 14. PTR Prentice Hall Englewood Cliffs, 1993.
- [26] C. M. Bishop. *Pattern recognition and machine learning (information science and statistics)*. Secaucus 2006.
- [27] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77, (1989) 257–286.

- [28] G. Aradilla, J. Vepa, and H. Bourlard. Using posterior-based features in template matching for speech recognition. In INTERSPEECH. 2006 .
- [29] R. O. Duda, P. E. Hart, and D. G. Stork. Pattern classification. John Wiley & Sons, 2012.
- [30] F. Zahedi. An introduction to neural networks and a comparison with artificial intelligence and expert systems. *Interfaces* 21, (1991) 25–38.
- [31] S. Hayman. The McCulloch-Pitts model. In Neural Networks, 1999. IJCNN '99. International Joint Conference on, volume 6. 1999 4438–4439 vol.6.
- [32] S. Haykin. Neural networks: a comprehensive foundation 2nd edition. *Upper Saddle River, NJ, the US: Prentice Hall* .
- [33] R. Lippmann. An introduction to computing with neural nets. *ASSP Magazine, IEEE* 4, (1987) 4–22.
- [34] R. Rojas. Neural networks: a systematic introduction. Springer, 1996.
- [35] M. K. S. Alsmadi, K. B. Omar, S. A. Noah et al. Back propagation algorithm: the best algorithm among the multi-layer perceptron algorithm. *International Journal of Computer Science and Network Security* 9, (2009) 378–383.
- [36] X. Yu, M. O. Efe, and O. Kaynak. A general backpropagation algorithm for feedforward neural networks learning. *Neural Networks, IEEE Transactions on* 13, (2002) 251–254.
- [37] O. Awodele and O. Jegede. Neural networks and its application in engineering. In Proceedings of Informing Science & IT Education Conference (InSITE), Macon State College, Macon, Georgia, USA. 2009 12–15.
- [38] Z. Rao and F. Alvarruiz. Use of an artificial neural network to capture the domain knowledge of a conventional hydraulic simulation model. *Journal of Hydroinformatics* 9, (2007) 15–24.
- [39] A. S. Pandya and R. B. Macy. Pattern recognition with neural networks in C++. CRC press, 1995.
- [40] R. D. Reed and R. J. Marks. Neural smithing: supervised learning in feedforward artificial neural networks. Mit Press, 1998.
- [41] Y. LeCun. Efficient learning and second order methods. In Tutorial presented at Neural Information Processing Systems, volume 5. 1993 49.
- [42] H. Ney. The use of a one-stage dynamic programming algorithm for connected word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on* 32, (1984) 263–271.
- [43] K. Ng and V. W. Zue. Subword-based approaches for spoken document retrieval. *Speech Communication* 32, (2000) 157–186.
- [44] T. M. Cover and J. A. Thomas. Elements of information theory. John Wiley & Sons, 2012.

- [45] P. R. Reddy, K. Rout, and K. S. R. Murty. Query word retrieval from continuous speech using GMM posteriorgrams. In International Conference on Signal Processing and Communications-2014 (SPCOM 2014), Indian Institute of Science, Bangalore, India. 2014 .
- [46] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey et al. The HTK book, volume 2. Entropic Cambridge Research Laboratory Cambridge, 1997.
- [47] D. Johnson et al. ICSI quicknet software package. *h ttp://www. icsi. berkeley. edu/Speech/qn.html* .
- [48] G. Manthana, K. Prahallad. Use of articulatory bottle-neck features for query-by-example spoken term detection in low resource scenarios. In Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, 7128–7132.