Development of Phonetic Engine for Punjabi Language

Dissertation

Submitted in partial fulfillment of the requirements for the award of degree of

Masters of Technology

in

Computer Science and Applications

Submitted By
Sakshi Mittal
(Roll No. 601203023)

Under the supervision of
Dr. R. K. Sharma
Professor, SMCA

School of Mathematics and Computer Applications
Thapar University
Patiala –147004

July 2014

CERTIFICATE

I hereby certify that the work which is being presented in the dissertation entitled, "**Development of Phonetic Engine for Punjabi Language**", in the partial fulfillment of the requirements for the award of degree of Master of Technology in **Computer Science and Applications** submitted in School of Mathematics and Computer Applications of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. R.K. Sharma and refers other researcher's work which are duly listed in the reference section.

The matter presented in this dissertation has not been submitted for award of any other degree of this or any other University.

(Sakshi Mittal)

Roll No.: 601203023

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. R.K. Sharma)

Professor, SMCA

Thapar University, Patiala

Countersigned:

(Dr. Rajesh Kumar)

Head

School Of Mathematics and Computer Applications

Thapar University, Patiala

(Dr. S.K. Mohapatra)

Dean of Academic Affairs

Thapar University, Patiala

CERTIFICATE

I hereby certify that the work which is being presented in the dissertation entitled, "**Development of Phonetic Engine for Punjabi Language**", in the partial fulfillment of the requirements for the award of degree of Master of Technology in **Computer Science and Applications** submitted in School of Mathematics and Computer Applications of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Dr. R.K. Sharma and refers other researcher's work which are duly listed in the reference section.

The matter presented in this dissertation has not been submitted for award of any other degree of this or any other University.

(**Sakshi Mittal)**

Roll No.: 601203023

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. R.K. Sharma)

Professor, SMCA

Thapar University, Patiala

Countersigned:

(Dr. Rajesh Kumar)

Head

School Of Mathematics and Computer Applications

Thapar University, Patiala

(Dr. S.K. Mohapatra)

Dean of Academic Affairs

Thapar University, Patiala

ABSTRACT

Hidden Markov Models have been used to model speech in many speech processing areas. This work presents a 'Phonetic Engine' that has been developed to provide the segmentation and labeling of continuous speech signals of Punjabi speech and provides the phoneme level recognition. This system is based on acoustic phonetic Hidden Markov Models that provide statistical representation of each of the distinct sounds that makes up a word. Segmentation of continuous speech signal is done at phone level and each distinct sound is assigned a 'phoneme' label. To get the statistical representation of each distinct sound, Hidden Markov models (HMMs) are trained at phoneme level using HTK toolkit. HTK is a statistical tool which is used for building HMMs. HMMs are the continuous density Gaussian mixtures. In this work, HMMs are trained for three modes of speech: Read speech mode, Lecture speech mode and Conversational speech mode. In read speech mode, HMMs are trained for male and female speakers separately. In each mode, to check the accuracy of speaker independent Phonetic Engine, performance evaluation has been done. Each HMM model has 5 states in which first and last states are non emitting, and only three remaining states, *i.e.*, state 2, 3 and 4 will have state output distribution. For each HMM (HMMs of all phones) Gaussian mixtures are continuously incremented for example, for first time say, for monophone HMM 'i', Gaussian mixture is computed only once for each state of HMM i, next time Gaussian mixture is computed twice for each state of hmm 'i'. Here, we have taken 32 Gaussian mixtures, it means that we will keep on computing the Gaussian mixtures for each state of each HMM till 32 Gaussian mixtures are not computed for each state of each HMM. In read speech mode, phonetic engine achieved 61.48% accuracy, for lecture speech mode it achieved 46.96% accuracy and for conversational speech mode the accuracy is 22.39%. The overall accuracy of phonetic engine for male and female speakers individually in read speech mode are 61.58% and 52.01%. In read speech mode, first male speaker got an accuracy of 61.21%, second male speaker got an accuracy of 57.91%, first female speaker got an accuracy of 49.96% and second female speaker got an accuracy of 57.76% accuracy. The HTK toolkit has been used in Ubuntu-12.04 32-bit environment. This Dissertation is divided into five chapters. A brief review of these chapters is given below.

Chapter 1 includes the introduction of tools used and all the files needed to implement Phonetic Engine.

Chapter 2 includes literature survey. This chapter is divided into two sections. First section includes review of literature on existing systems that have been implemented using HTK toolkit and second section includes review of literature on existing systems that have been developed using HMMs but with different tool.

Chapter 3 includes the working of Phonetic Engine and the algorithms used to provide the training to HMMs and for checking the accuracy of engine and getting the phoneme level recognition of unknown utterances.

Chapter 4 includes the description of data collected in all three modes and the accuracy of Phonetic Engine in each mode.

Chapter 5 concludes the work done in this dissertation with an illustration on future scope for the same.

ACKNOWLEDGEMENTS

A dissertation cannot be completed without the help of many people who contribute directly or indirectly through there constructive criticism in the evolution and preparation of this work. It would not be fair on my part, if I don't say a word of thanks to all those whose sincere advice made this period a real educative, enlightening, pleasurable and memorable one.

First of all, a special debt of gratitude is owned to my supervisor **Dr. R.K. Sharma**, for his gracious efforts and keen pursuits, which has remained as a valuable asset for the successful completion of research work. His dynamism and diligent enthusiasm has been highly instrumental in keeping my spirit high. The flawless and forthright suggestions blended with an innate intelligent application have crowned my task a success.

I am equally grateful to **Dr. Rajesh Kumar**, Associate Professor and Head, School of Mathematics and Computer Applications, for his motivation and inspiration that triggered me for the dissertation work.

I also like to offer my sincere thanks to all faculty members, teaching and non-teaching staff of School of Mathematics and Computer Applications (SMCA) and staff of central library, Thapar University, Patiala for their assistance.

I would also like to thank to my parents and friends for their constant encouragement during the entire course of my work.

Above all, I owe my reverence to Almighty for the kindness who blessed me at finish of whole work.

(Sakshi Mittal)

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Expanded Form
HMM	Hidden Markov Model
LPC	Linear Predictive Coding
MFCCs	Mel Frequency Cepstral Coefficients
IPA	International Phonetic Alphabet
HSMM	Hidden Semi Markov Model
MMI	Maximum Mutual Information
GS	Gaussian Selection
CDM	Chinese Dictation Machine
LVCSR	Large Vocabulary Continuous Speech Recognition
MMIE	Maximum Mutual Information Estimation
LFPC	Log Frequency Power Coefficient
LPCC	Linear Prediction Cepstral Coefficient
EM	Expectation Maximization
PD	Pronunciation Dictionary
CAC	Command and Control Corpus
ADC	Arabic Digit Corpus
GMM	Gaussian Mixture Modeling

CMU	Carnegie Mellon University
CD-HMM	Continuous-Density Hidden Markov Model
MCE	Minimum Classification Error
CML	Conditional Maximum Likelihood
ANN	Artificial Neural Network
MFT	Missing Feature Training
DTW	Dynamic Time Warp
ML	Maximum Likelihood
FFNN	Feed Forward Neural Network
PE	Phonetic Engine

CONTENTS

ACKNOWLEDGEMENTS

A dissertation cannot be completed without the help of many people who contribute directly or indirectly through there constructive criticism in the evolution and preparation of this work. It would not be fair on my part, if I don't say a word of thanks to all those whose sincere advice made this period a real educative, enlightening, pleasurable and memorable one.

First of all, a special term of gratitude is owned to my supervisor **Dr. R.K. Sharma**, for his gracious efforts and keen pursuits, which has remained as a valuable asset for the successful completion of research work. His dynamism and diligent enthusiasm has been highly instrumental in keeping my spirit high. The flawless and forthright suggestions blended with an innate intelligent application have crowned my task a success.

I am equally grateful to **Dr. Rajesh Kumar**, Associate Professor and Head, School of Mathematics and Computer Applications, for his motivation and inspiration that triggered me for the dissertation work.

I also like to offer my sincere thanks to all faculty members, teaching and non-teaching staff of School of Mathematics and Computer Applications (SMCA) and staff of central library, Thapar University, Patiala for their assistance.

I would also like to thank to my parents and friends for their constant encouragement during the entire course of my work.

Above all, I owe my reverence to Almighty for the kindness who blessed me at finish of whole work.

Sakshi

(Sakshi Mittal)

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Expanded Form
HMM	Hidden Markov Model
LPC	Linear Predictive Coding
MFCCs	Mel Frequency Cepstral Coefficients
IPA	International Phonetic Alphabet
HSMM	Hidden Semi Markov Model
MMI	Maximum Mutual Information
GS	Gaussian Selection
CDM	Chinese Dictation Machine
LVCSR	Large Vocabulary Continuous Speech Recognition
MMIE	Maximum Mutual Information Estimation
LFPC	Log Frequency Power Coefficient
LPCC	Linear Prediction Cepstral Coefficient
EM	Expectation Maximization
PD	Pronunciation Dictionary
CAC	Command and Control Corpus
ADC	Arabic Digit Corpus
GMM	Gaussian Mixture Modeling

CMU	Carnegie Mellon University
CD-HMM	Continuous-Density Hidden Markov Model
MCE	Minimum Classification Error
CML	Conditional Maximum Likelihood
ANN	Artificial Neural Network
MFT	Missing Feature Training
DTW	Dynamic Time Warp
ML	Maximum Likelihood
FFNN	Feed Forward Neural Network
PE	Phonetic Engine

CONTENTS

CHAPTER 1

Introduction

In today's world, there is a lot of advancement in technology and this advancement is continuing for the welfare of human beings. Researchers are putting much more efforts on developing newer computer technology to meet the needs of society because once a particular system is trained for some task then for the similar tasks it can easily compute the results. Computers can solve complex tasks in a simple and easy manner. For interaction between computers and human beings, an interface is required which is provided by some hardware components of computer such as keyboard, mouse, joystick *etc.* There are some limitations in dealing with these interfaces such as mouse requires a very good hand and eye coordination and also requires people to be proficient in English. The physically disabled people find computers difficult to use. Speech which is a natural and very easy way of exchanging the information, if used as a medium to interact with the computer can solve all these problems. For doing this, some speech interfaces such as speech synthesizer and speech recognizer are required.

Speech recognition and speech synthesis both require phonetic transcription. In speech synthesis, firstly, in preprocessing stage text is assigned the phonetic transcription and then front end divides and marks the text into prosodic units (syllable boundary marking, break index marking, pitch marking *etc.*). Phonetic transcription and prosody information together make up the symbolic linguistic representation and then synthesizer converts symbolic linguistic representation into sound. In speech recognition, speech is provided as an input to system and then corresponding phonetic transcription is generated by the system as output.

Phonetic Engine (PE) is such a module that uses the acoustic phonetic information present in the speech signal for converting the speech signal into symbolic form. This symbolic form is nothing but the basic sound units present in the spoken utterances of speech signal. These basic sound units can be represented in symbolic form using International Phonetic Alphabet (IPA) transcription standard. Acoustic phonetic information means that the PE will use the sounds of phones of spoken utterances and these sounds are represented in the symbolic form.

Phonetic Engine has been developed in this work using the HTK toolkit. HTK is a statistical toolkit to build Hidden Markov Models (HMMs).

1.1 Relevant Definitions

1.1.1 Phoneme

A phoneme is the smallest basic unit of sound system of a language which when combined with other phonemes makes meaningful units such as words. There is a distinction between phone and phoneme. A phone is only just a sound and are infinite in number, not necessary that on combining different phones they would produce some meaningful unit, can be simply a noise, word, animal cry *etc.* but phoneme always produce a meaningful unit. For example, words 'madder' and 'matter', both composed of different phonemes but in American English when pronounced both words sounds same, *i.e.,* both words have same phones. If a particular person makes different sounds, phones may be same in all languages but are written differently from one language to another and for that different phonemes are used.

1.1.2 Acoustic Model

An acoustic model contains the statistical representation (HMMs) of different sounds which when combined makes a meaningful unit such as word. Phonemes are assigned to each statistical representation of sound. Acoustic model is created by taking the speech database and their transcriptions which are given as input to some software which in return provides the statistical representation of different sounds.

1.1.3 Language Model

This model involves the grammar and dictionary used by the software which helps in recognizing the phoneme of unknown utterances.

1.2 Transcription Using IPA

The visual representation of speech sounds (phones) is called the phonetic transcription. Phonetic alphabet, *e.g.,* IPA is the most common type of phonetic transcription. Suppose there is a Punjabi word 'ਇਸਨੂੰ', this can be transcribed as 'isanū'. Phonetic transcription and orthography both provide different functionality. An orthography includes rule of spelling. It is a standardized system for using a particular writing system to write a particular language. Phonetic transcription deals with the sound of phones used in words, *i.e.,* it tells us the pronunciation of words. For

example, transcription is essential in English dictionary because most of the words in English are not pronounced in the same way as they are spelled.

There is one more advantage of transcribing the words using IPA chart. This advantage is that if we want a computer to understand spoken utterances of any language, transcribe all the utterances used in that language using IPA chart so that computer becomes language independent.

Transcription can be done at phoneme level, word level or at syllable level. In this work, main focus has been the transcription at phoneme level, *i.e.*, phonetic engine will work at phoneme level.

Some of the examples of phonetic transcription are given below.

Table 1.1: Symbolic Form Representation of Spoken Utterances

Punjabi	Transcription
ਆਪਣਾ ਆਪਣਾ ਹਿੱਸਾ ਵਰਿਆਮ ਸੰਧੂ	əpɳa əpɳa hɪsa vrjam səndʊ
ਗੁਰੂ ਜੀ ਸੁਲਤਾਨਪੁਰ ਵਿਚ	gʊrʊ ʤɪ sʊltanpʊr vɪʧ
ਇਸ ਲਈ ਤਾਂ ਉਨਾ ਦੇ ਪਿੰਡ ਦੇ ਬਾਬੇ	ɪs leɪ ta_ʊna de pinɖ de babe
ਜੇ ਮਗਰੋ ਸਲਾਹ ਕਰਕੇ ਗੁਰੂ ਜੀ ਨੇ ਉਨਾਂ ਦੇ	ʤe magro səla krke gʊrʊ ʤɪ ne una de
ਅਪਣੇ ਲੰਗਰ ਵਿਚੋ ਥੋੜਾ ਭੋਜਨ ਤਾਂ ਮੰਗੋ	əpɳe ləngr vɪʧo tʰoɽa poʤn ta məngəo

The IPA chart that has been used in this work for transcription purpose is given in Figure 1.1.

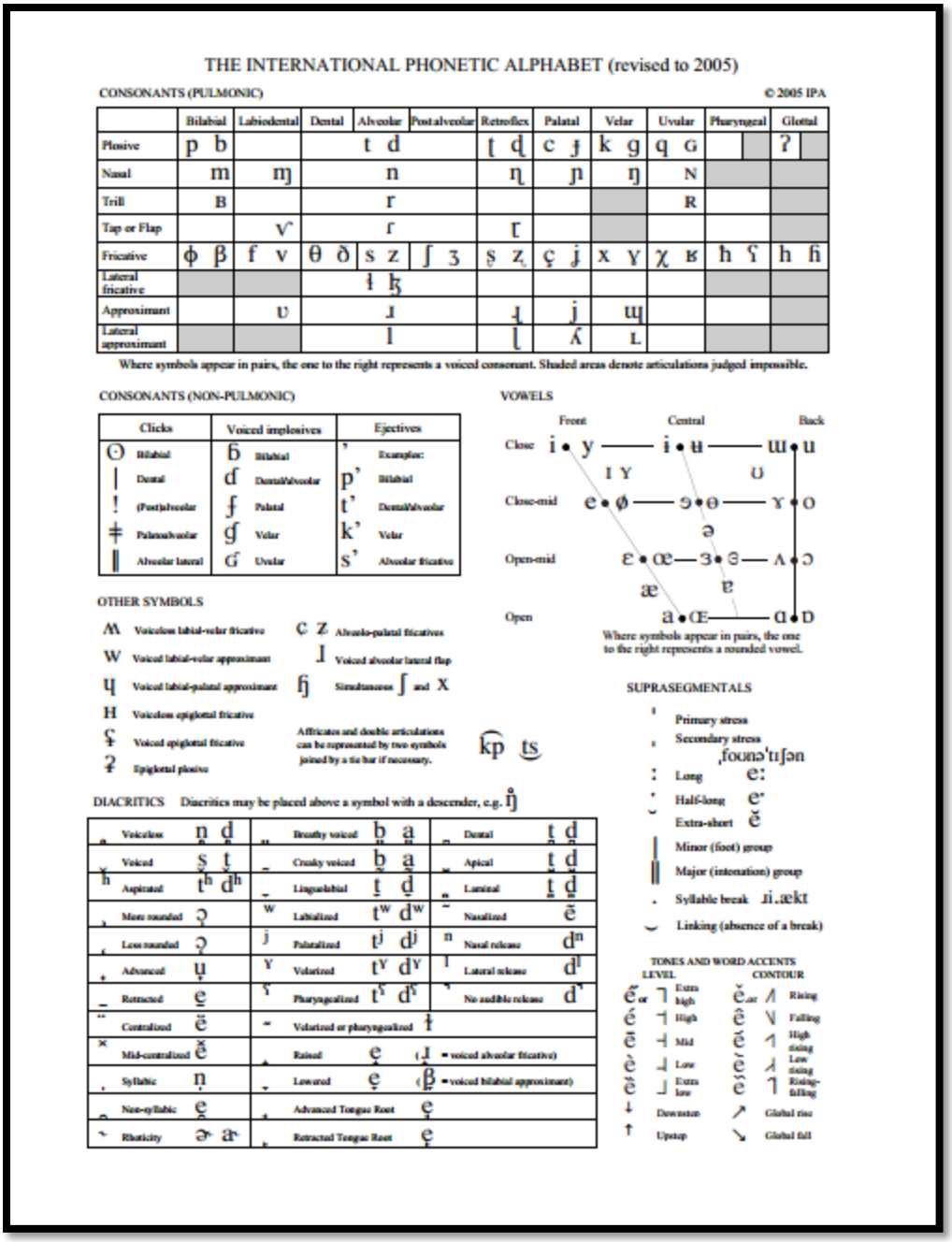


Figure 1.1: IPA Chart

1.3 Hidden Markov Model (HMM)

HMM is a Markov process that is divided into two components: the observable states and unobservable states (hidden states). It is a doubly stochastic process such that underlying stochastic process is hidden (not observable) but through another set of stochastic process (that generated the sequence of observed symbols) can be observed. For example, suppose in a room

there is a barrier (say, curtains) between two persons and one person cannot see what the other side happening. First person on one side of curtain is flipping a single coin (or multiple coins) and telling the results to second person on the other side of curtain but not exactly telling what he is doing. Thus, second person can only observe the results without knowing what other side exactly going on.

A HMM '*M'* can be defined by a set of '*N*' states, '*P*' observational symbols and three probabilistic matrices *A*, *B* and π. As such,

$$M = (A, B, \pi)$$

where, π denotes initial state probabilities, A denotes state transition probabilities and B denotes observation probabilities.

HMMs can be classified into two categories, namely, Discrete Observation HMM and Continuous Observation HMM.

1.3.1 Discrete Observation HMM

HMMs are required for recognition purpose because words are made up of distinct sequence of elements and these distinct elements in sequence are the phonemes. Suppose there is a word say, 'ਇਸਨੂੰ' (isanū) needs to be recognized, we need some learning which is expressive enough to capture the sounds such as first it sounds like iii for a longer amount of time, then it is 's' for longer amount of time, then it is 'a' for a short moment, then it is 'n' for a short while and then, it is 'u' for a longer amount of time. HMMs are good enough to capture these sounds that can be represented using states, probability transition matrix and distribution associated with each state from which observations are drawn. States are the phonemes, transition matrix indicated that we move through the word form first phoneme to last phoneme, staying a variable amount of time in each phoneme and the distribution associated with each state denotes how each phoneme translates into acoustic feature.

In real world speech recognition, the phoneme themselves are modeled as left-to-right HMMs (*e.g.*, to model separately the transition part at the begin of the phoneme, then the stationary part

and finally the transition at the end). Words are then represented by large HMMs made of concatenation of smaller phonetic HMMs.

In Discrete Observation HMM, observation sequences are drawn from discrete distribution associated with each state and in Continuous Observation HMM, observation sequences are drawn from continuous distribution associated with each state and these observations are scalars or vectors.

For discrete observation HMM, following notations are used.

$M =$ number of observation symbols.

$Q = \{q_1,\ q_{2,} \dots, q_N\}$ are the states.

$O = \{o_1,\ o_{2,} \dots, o_p\}$ discrete set of possible symbols observations.

$A = \{a_{ij}\}$, $a_{ij} = P(q_j$ at t+1$|\ q_i$ at t), state transition probability distribution.

$B = \{b_j(o_p)\}$, $b_j(o_p) = P(o_p$ at t$|q_j$ at t), observation probability distribution in state j.

$\pi = \{\pi_i\}$, $P(q_i$ at t = 1), initial state distribution.

1.3.2 Continuous Observation HMM

In Continuous Observation HMM, all the notations used in discrete observation HMM denote the similar meaning except the observation sequences. In this, $b_j(o_p)$'s are computed as some probability density functions or mixtures of them. Some restrictions must be placed in order to re-estimate the parameters of probability density function (*pdf*). The restriction is that the *pdf* can be only log-concave or elliptically symmetric density. The mostly used log-concave or elliptically symmetric density is the Gaussian density (Nilsson *et al.*, 2002). The most general representation of *pdf* is a finite mixture of given form.

$$b_j(o_p) = \sum_{m=1}^{M} c_{jm}\ b_{jm}(o_p),\ j = 1, 2, \dots, N \qquad \dots (1.1)$$

where, M is the number of mixtures, mixture weight, $\sum_{m=1}^{M} c_{jm} = 1,\ j = 1, 2, \dots, N$.

$b_{jm}(o_p)$ is a d-dimensional log-concave or elliptically symmetric density with mean vector μ_{jm} and covariance matrix $\sum jm$.

$$b_{jm}(o_p) = N(o_p, \mu_{jm}, \sum jm)$$

The Gaussian density can be computed using the given formula.

$$b_{jm}(o_p) = N(o_p, \mu_{jm}, \sum jm) = \left\{1/(2\pi)^{d/2} \mid \sum jm \mid^{1/2}\right\} e^{-1/2(o_p-\mu_{jm})^T \sum_{jm}^{-1}(o_p-\mu_{jm})} \quad \text{... (1.2)}$$

As the length of feature vector increases, the size of covariance matrices increases in square proportional to the vector dimension. The diagonality provides a simpler and faster implementation for the probability computation.

$$b_{jm}(o_p) = N(o_p, \mu_{jm}, \sum jm) = \left\{1/(2\pi)^{d/2} \mid \sum jm \mid^{1/2}\right\} e^{-1/2(o_p-\mu_{jm})^T \sum_{jm}^{-1}(o_p-\mu_{jm})}$$

$$= \left\{1/(2\pi)^{d/2} \left(\prod_{l=1}^{d} \sigma_{jml}\right)^{1/2}\right\} e^{-\sum_{l=1}^{d} \frac{(o_{pl}-\mu_{jml})^2}{2\sigma^2_{jml}}} \quad \text{... (1.3)}$$

where, $\sigma_{jm1}, \sigma_{jm2}, \ldots, \sigma_{jmd}$ are the diagonal elements of covariance matrix $\sum jm$.

1.3.3 Classification of HMM Structure

HMM structures are classified based on the network topologies that they inherently possess. These topologies are given in the following subsections.

1.3.3.1 Ergodic Topology

In this topology, any state can be reached from any other state. But this topology cannot be used for speech recognition because speech includes an ordered sequence of sounds.

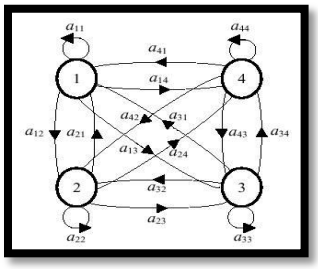


Figure 1.2: Ergodic Topology

1.3.3.2 Left-to-Right Topology (or Bakis Model)

In this, states are reached in an ordered sequence and once any state is left then, it cannot be reached again, *i.e.*, states are reached only in one direction from left to right. This topology is generally used for speech recognition.

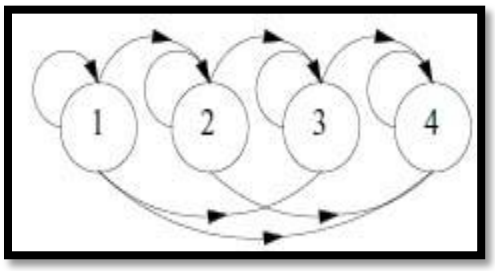


Figure 1.3: General Left-to-Right topology

The transition probability matrix for this topology can be represented as given below.

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ 0 & a_{22} & a_{23} & 0 \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

1.3.4 Basic Issues in HMM

Once we have an HMM, there are three problems of interest. Following three problems arise in applying the model for recognition task.

- **Learning Problem**

  In this, general structure of HMM, *i.e.*, number of hidden and visible states and some training observation sequences $O = o_1, o_2, \dots, o_p$ are given, and HMM parameters $M = (A, B, \pi)$ are to be determined that best fit training data. To resolve this issue, Balum-Welch algorithm is used.

- **Decoding Problem**

  HMM $M = (A, B, \pi)$ and observation sequences $O = o_1, o_2 \dots, o_p$ are given and most likely sequence of hidden states are to be computed which produced this observed sequence. To resolve this issue Viterbi algorithm is used.

- **Evaluation Problem**

  HMM $M = (A, B, \pi)$ and observation sequences $O = o_1, o_2, \dots, o_p$ are given and the probability that model *M* has produced this sequence is to be computed. To resolve this issue forward-backward algorithm is used.

1.3.5 Use of HMM

One of the most common usage of HMM is for speech recognition where the speech audio waveform is the observed data and the spoken text is the hidden state, *i.e.*, HMMs are used to recognize spoken utterances in the speech given observation sequence $O = o_1 o_2 \dots . o_p$. Utterances may be a word, phoneme or a sentence.

In Speech Recognition, given a sequence of observations, the most-likely corresponding sequence of states would be estimated using Viterbi algorithm, the probability of the sequence of observations would be computed using forward algorithm, and the Baum–Welch algorithm would estimate the initial probabilities, transition probabilities, and the observation function of a HMM, *i.e.*, parameters (A, B, π).

1.3.5.1 Principle of HMM

This subsection consists of how recognizer recognizes the spoken utterances. According to Young *et al.* (2009), speech signal is considered to be a message which is encoded as a sequence of symbols by the speech recognition system, *i.e.*, spoken utterances of speech signal are a sequence of some symbols. At the time of recognition, *i.e.*, spoken utterances are given and sequence of symbols corresponding to spoken utterances are to be determined, the continuous speech signal is parameterized into equally spaced discrete parameter vectors. It is assumed that this sequence of parameter vectors is an accurate representation of speech signal. Firstly, speech signal is divide into overlapping frames with each frame of an approximate length of 10ms, and then, from each frame parameter vectors are extracted. Framing is done because it is assumed that for a short duration in milliseconds signal is stationary though it is not strictly true and is just a reasonable approximation. These parameter vectors can be Mel-Frequency Cepstral Coefficients (MFCCs), Linear prediction coefficients (LPCs) *etc.* Recognizer would do a mapping between these parameter vectors and underlying symbol sequences.

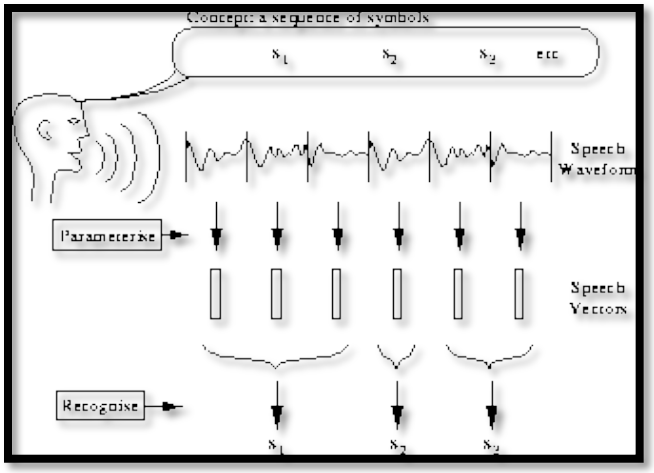


Figure 1.4: Speech recognizer

1.3.5.2 Elements of HMM

- There are finite number of states '*N*' such that signal possesses distinct properties within each state.
- Based on transition probability distribution depending only on the previous state, at each clock time '*t*', a new state is entered.
- Based on probability distribution depending only on the current state, an observation output symbol is generated after each transition is made. Thus, there are '*N*' observational probability distributions.

1.4 HTK Toolkit

HTK toolkit is a statistical tool to build HMM models. The main objective for designing this toolkit is to build HMM-based Speech Processing tools, specifically recognizers. It is mainly concerned with HMMs of which each observation probability distribution is represented by Gaussian mixture density (Young *et al.*, 2009). It consists of two major processing stages.

i. Using training utterances (recorded speech used for training purpose) and their corresponding transcriptions, the HTK training tools compute the parameters of set of HMMs.

ii. Unknown utterances (recorded speech whose transcription is to be done) are transcribed using the HTK recognition tools.

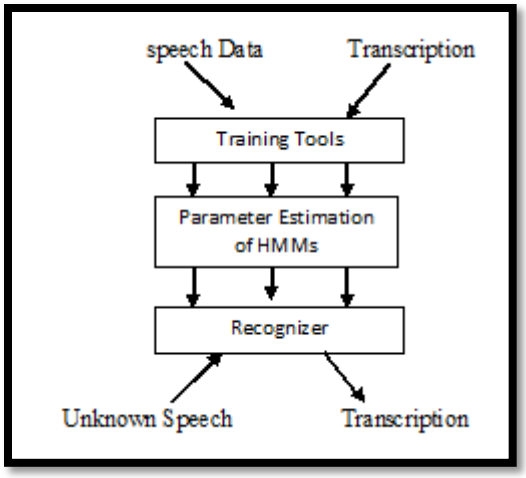


Figure 1.5: Working of Phonetic Engine

1.4.1 Training Using HTK Toolkit

At the time of training of system, HTK uses Baulm -Welch algorithm which uses the forward-backward algorithm and at the time of recognition HTK uses Viterbi algorithm.

Given below are the tools used by HTK toolkit to provide training to HMMs.

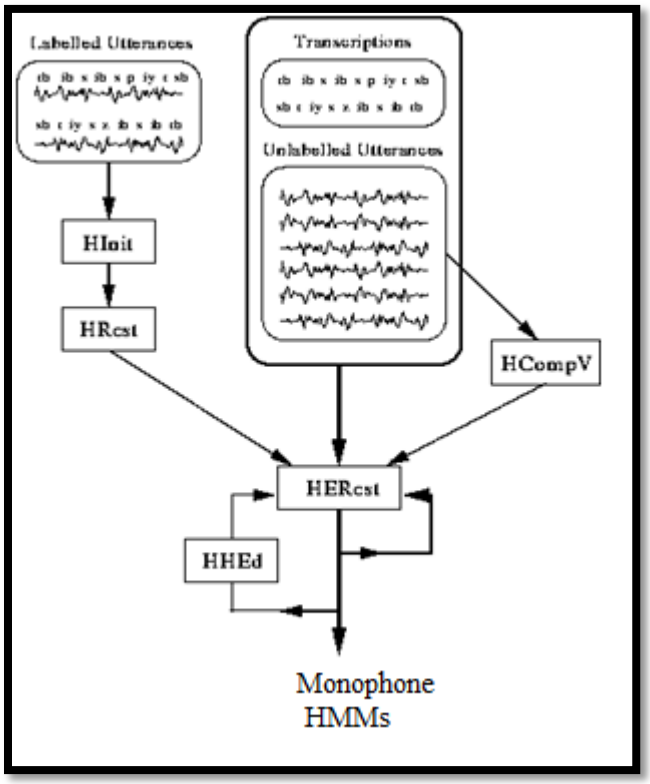


Figure 1.6: Training HMMs

HCOMPV

This is basically called flat start training scheme in which HCOMPV tool is used such that identical initialization is done to all the phoneme models and each phoneme model have state means and variances equal to global speech means and variances. It computes the global means and variances of a set of training data (Young *et al.*, 2009).

When there is a limited data for providing training to large model set, setting a floor becomes indispensable to avoid variances from being badly underestimated through limited data. One method of doing this is to define a variance macro. To generate these variance floor macros, HCOMPV can be used, with values equal to specified fraction of the global variance.

HMM Editor- HHED and Re-estimation Tool- HERest

According to Young *et al.* (2009), the HMM editor HHED takes as input a set of HMM definitions and in output provides a new modified set of HMM definitions, usually to a new directory . The general syntax of HHED is given below.

HHEd -H MMF1 -H MMF2 ... -M new_dir cmnds.hed Hmmlist

This command would read all the models listed in HMM list and defined by files MMF1, MMF2, in our case it is a single file namely, hmmdef, which defines all the monophone HMM models, then carry out the editing operations as mentioned in the cmds.hed and then writes out the result to the new directory 'new_dir'. cmnds.hed is an edit script consisting of list of edit commands such that each command begins with two letter command name and written on a separate line. In our case, command 'MU' is used. MU command is responsible for the conversion from the single Gaussian HMMs to multiple mixture component HMMs. For example, if we are generating the mixtures for HMM2 which means that for each state of each model, two times mixtures need to be computed which is defined in a file having extension hed say, mix_2.hed. Figure 1.7 shows the commands defined in mix_2.hed.

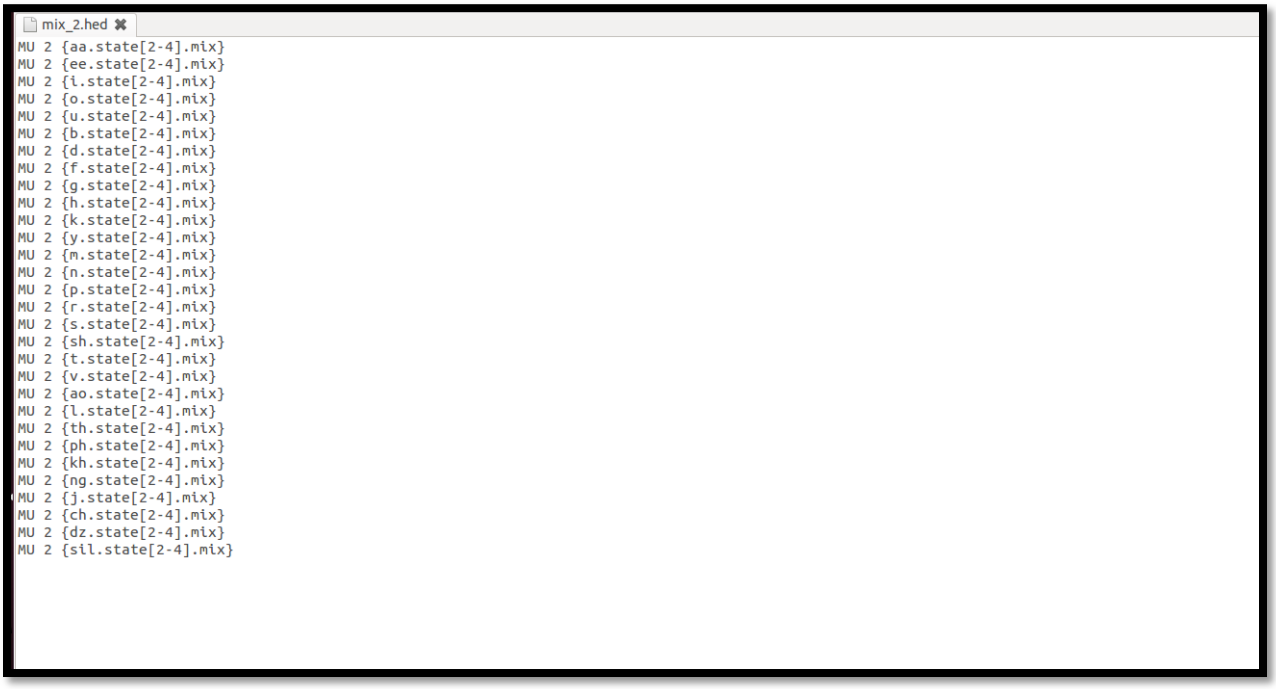


Figure 1.7: HMM Editor- HHED : mix_2.hed

In the state output distribution, commands defined in Figure 1.7 would increment the number of Gaussian mixture components for state 2,3 and 4 of all the defined models.

Each execution of HHED is followed by re-estimation using HERest. HERest is a core HTK training tool. This tool perform single re-estimation of parameters of whole set of HMMs simultaneously using Baum-Welch Re-estimation algorithm (in-built). Re-estimation means refining the parameters of existing HMM. HERest operates in two stages.

Stage 1:- The corresponding phoneme models for each training utterance are concatenated for accumulating the statistics of state occupation, means, variances *etc*. for each HMM in the sequence using forward backward algorithm.

Stage 2:- The accumulated statistics are used to re-estimate the HMM parameters when all of the training utterances are processed.

HINIT

Using this, more detailed initialization of parameters of HMM can be provided and computed using viterbi style of estimation over HCOMPV.

HREST

HEREST and HREST are used to improve the existing HMM parameters. HREST perform isolated-unit training and HEREST operate on whole model sets and does embedded-unit training (Young *et al.*, 2009).

1.4.2 Testing Using HTK Toolkit

After providing training to HMMs, to check the accuracy of trained system testing is done. For testing HTK uses some of the tools which are given below.

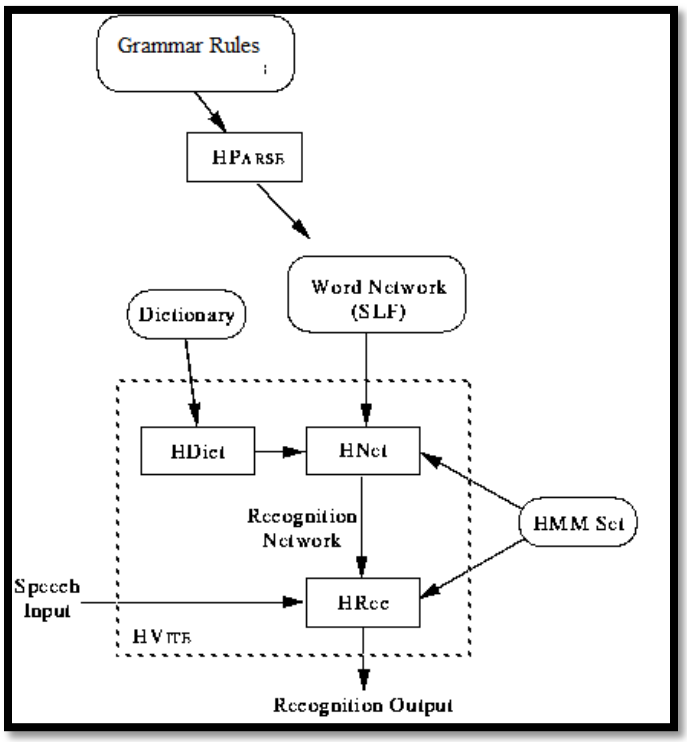


Figure 1.8: Recognition

HVITE

HTK provides a tool HVITE which is responsible for recognition, *i.e.,* it provides the transcription of unlabelled utterances. HVITE provides the combined functionality of HDiet, HNet and HRec (Young *et al.*, 2009). It performs evaluation on trained HMMs using testing speech data. As an input, it takes dictionary, word network and HMM models and generates a recognition network and then recognize each input utterance.

HResults

This is the analysis tool of HTK responsible for computing the actual performance of trained system by comparing desired output with the actual output of system and provides the comparison result in the form of confusion matrix (Young *et al.*, 2009).

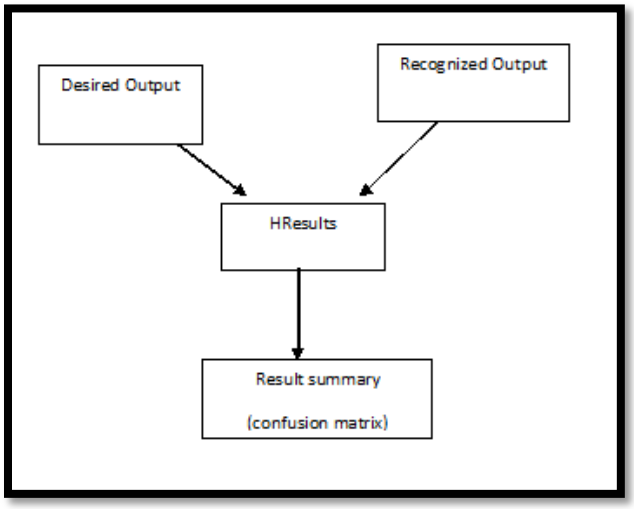


Figure 1.9: Working of HResults

1.5 Feature Extraction Technique

It is possible that speech can be recognized directly from digital waveform but theoretically, practically it is too complicated task because of variability present in the speech signal. In order to simplify the task, some features are extracted from the speech signal that reduces the variability. For extracting the features from speech signal, firstly speech signal is divided into overlapping frames then on each frame windowing is done. Within each frame it is assumed that signal is stationary though it is not. Various windows are used for feature extraction (Paul *et al.*, 2011).

- Rectangular window: $w(n) = 1$
- Hanning window : $w(n) = 0.5\,(1 - cos\,(2 * pi * n\,/\,(N-1)))$
- Hamming window : $w(n) = 0.54 - 0.46\,cos\,(2 * pi * n\,/\,(N-1)))$
- Cosine window : $w(n) = cos\,(\,(pi * n/n - 1) - (pi/2)\,)$

Most common window is hamming window.

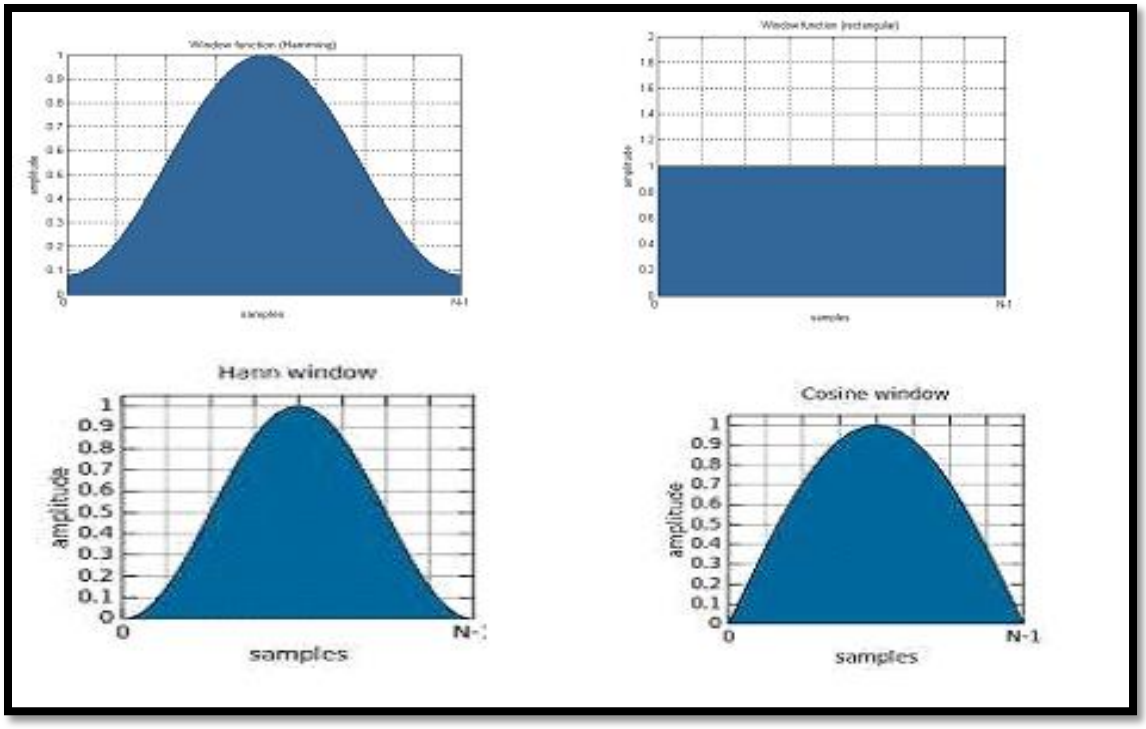


Figure 1.10: Window Functions

Human beings have the ability that they can differentiate among sounds even though different people speaks same word they can recognize that it is the same word. The same methodology is needed to be used by the speech recognition system as the human beings use to identify and differentiate sounds. Though the same word is spoken by different people in a different manner but that spoken utterance has some common features which help the human beings to recognize them and those features are need to be extracted that can help the system in recognition.

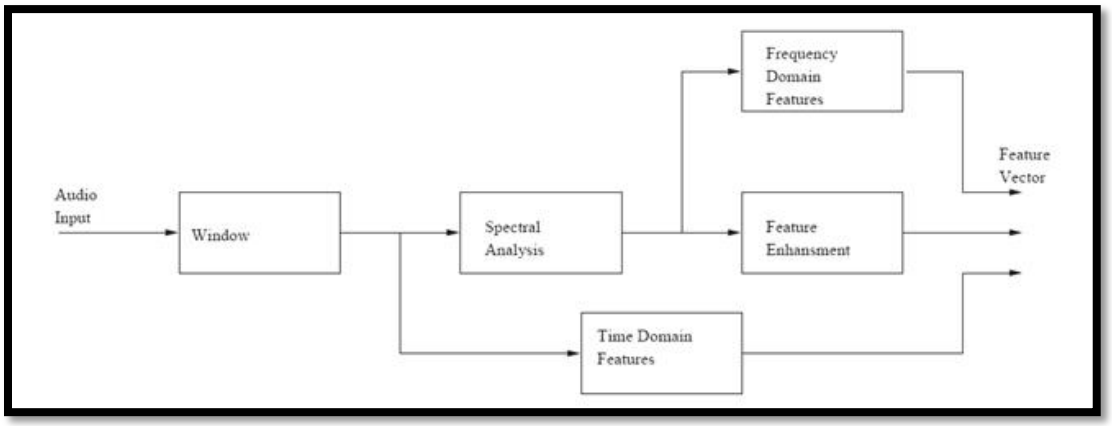


Figure 1.11: Feature Extraction

1.5.1 Mel Frequency Cepstral Coefficients (MFCCs)

It is one of the standard methods for extracting features. Due to its dependency on spectral form it is more sensitive to noise. In automatic speech recognition, using 20 MFCC coefficients is very common but 10-12 coefficients are considered to be enough for encoding speech.

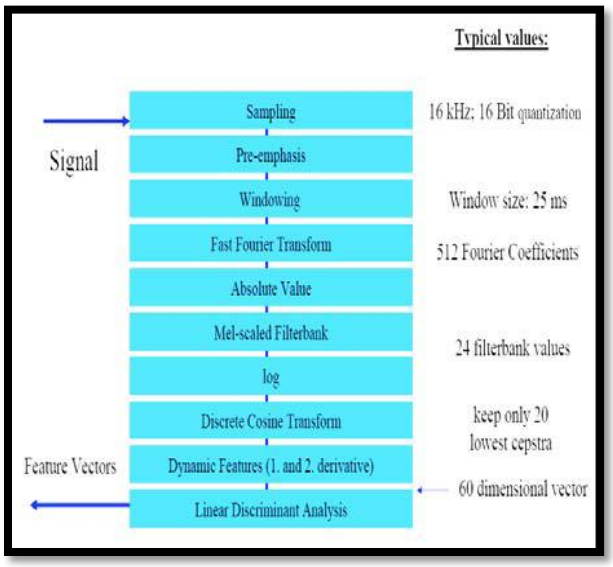


Figure 1.12: Complete pipeline of MFCC

MFCC can be computed using the given equation.

$$Mel(f) = 2595 * log10(1 + f/700) \quad \text{... (1.4)}$$

MFCCs are the possible approximation that are considered to be closest to human ear. These are generated by passing the speech signal through the high band pass filters. This results in distinction between higher frequency and lower frequency (Paul *et al.*, 2011).

CHAPTER 2

Literature Survey

In this dissertation, literature survey is divided into two parts. The first part deals with existing recognition systems that have been developed using HTK toolkit and second part deals with those existing recognition systems that have been developed using HMMs but use some other tools.

2.1 Literature Survey on the Use of HTK in Speech

Woodland *et al.* (1995) proposed speech recognition system based on HTK toolkit. This system was based on tied-state context dependent continuous density HMMs, *i.e.*, Gaussian mixture HMMs. MFCCs was used as a feature extraction technique which included 12 cepstrums, normalized log energy, delta and acceleration coefficients, *i.e.*, first and second order derivatives of parameters. In this work, vocabulary consisting of 65464 words, 4-gram language model was used.

Azmi *et al.* (2008) proposed automatic speech recognition for Arabic speech using syllables and did a comparison of monophone, triphone and word based recognition with syllable based recognition. Proposed system was based on HMMs and was implemented using HTK toolkit. For training and testing purpose, data was collected from 44 speakers. MFCCs was used as feature extraction technique. It was observed that syllable based recognition overcome the performance of other recognizers as recognition rate of monophone based recognizer, triphone based recognizer, word based recognizer was 90.75%, 92.24%, 91.64% and of syllable based recognizer, it was 93.43%.

Al-Qata *et al.* (2010) proposed the automatic speech recognition engine for Arabic language which was HMM based and implemented using HTK toolkit. This engine could recognize both isolated words and continuous speech. MFCC was used as the feature extraction technique. To compute HMM parameters training was based on tri-phones. Data was collected from thirteen Arabian native speakers which further divided into ten speaker dependant and three speaker independent data for evaluating the performance of the system. It was observed that the system could recognize the speaker dependant and speaker independent for both continuous speech and

isolated words. The overall performance of system was 90.62% for sentence correction, 98.01% for word correction and 97.99% word accuracy.

Kumar *et al.* (2011) proposed Speech Recognition System for isolated words of Hindi language. MFCC was used as a feature extraction technique. This system was based on HMMs and developed using HTK toolkit on Linux platform. A vocabulary of size of 30 words was used for training purpose. This training data was collected from eight speakers. At the time of performance evaluation which was conducted in room environment, the overall accuracy of the system that had been observed was 94.63% and word error rate was 5.37%.

Dua *et al.* (2012) presented automatic speech recognition system for isolated words for Punjabi language. This was based on a statistical approach 'HMMs' using HTK toolkit. From eight speakers data was collected for training of 115 distinct Punjabi words and some samples were collected from six speakers in real time environment for analyzing the performance of system. A GUI was also implemented using java language to make system more interactive. Data was recorded using audacity recording tool. For feature extraction, MFCC technique was used. The performance of system was analyzed in two cases: same speakers involved in both training and testing and speakers involved in testing only. It was observed that the system showed its average performance in range of 94% to 96%.

Kumar *et al.* (2012) proposed connected-words speech recognition system for Hindi language based HMMs using HTK toolkit. Training data was collected from 12 speakers including both males and females and 5 speakers data was collected for testing purpose. The system was trained for recognizing any sequence of words from a vocabulary of size 102 words. MFCCs features extracted from the speech signal and then the system was trained for computing HMM parameters using word level acoustic models. Many experiments were done in different environments such as open space, lab room, room environment, classroom and in market. It was observed that as the noise level increases, the performance of the system degraded and the accuracy of system was 87.01%.

Choudhary *et al.* (2013) proposed automatic speech recognition for Hindi language using HMMs. Proposed system was developed for recognizing the isolated and connected words of Hindi speech and implemented using HTK toolkit. System was trained for 100 different isolated

words and each word was uttered ten times. After evaluation the performance of system, recognition rate was 95% and word error rate was 5%.

Saini *et al.* (2013) proposed Hindi Speech Recognition System based on HMMs using HTK toolkit. This system recognizes 113 Hindi isolated words and three different states (6, 8, 10) HMM topology was used. For recognizing the speech, word model was used. For data parameterization, MFCC features were extracted. For training purpose, from six speakers data was collected. This system was developed on Linux environment. In this system, recognition involves two cases: recognition by speakers involved in both training and testing using three different states in HMM topology and recognition by speakers involved only in testing using three different states in HMM topology. In first case, with 6 states in HMM topology accuracy was 93.96%, with 8 states in HMM topology accuracy was 91.66%, with 10 states in HMM topology accuracy was 96.61%. In second case, with 6 states in HMM topology accuracy was 92.68%, with 8 states in HMM topology accuracy was 91.17%, with 10 states in HMM topology accuracy was 95.49%.

Sarma *et al*. (2013) proposed Phonetic engine development for Assamese language and discussed some issues related to it. Proposed work was implemented using HTK toolkit and based on HMMs. It was a phoneme based recognizer. In three different modes speech data was recorded: read speech mode, lecture speech mode and conversational speech mode. Read speech data was used to train HMMs and performance accuracy that was achieved in all three modes was 47.31% in read speech mode, 45.30% in lecture speech mode and 36.13% in conversation speech mode.

Thakuria *et al.* (2013) proposed a speech recognition system for BODO language based on left-to-right five state HMMs using HTK toolkit in Linux environment. This system was trained for continuous BODO speech signal. From the speech signal MFCCs and excitation parameters (Fundamental frequency F0) were extracted. This system was trained for 38 context dependant BODO phonemes. From the results, it was observed that the system was sensitive to changing scenarios and changing spoken methods. For making system more fast noise reduction techniques might be applied to it.

Tripathy *et al.* (2013) proposed Hindi Speech Recognition System using different feature extraction techniques MFCCs and linear predictive coding (LPC) and afterwards comparison has been made between both techniques. In this work, HMM was used as a classifier and were implemented through HTK toolkit. Proposed system has been tested on both environments speaker dependant and speaker independent . In this work, a vocabulary of size of 35 Hindi words was prepared and five speakers (2 males and 3 females) were used for recording the Hindi speech. Data was prepared using audacity. After preparing all data, speech recognition system was prepared by applying MFCC and LPC as feature extraction techniques using HTK toolkit. Then, performance of system was analyzed under four cases: Speaker dependant environment and MFCC as feature extraction technique, speaker independent environment and MFCC as feature extraction technique, speaker dependant environment and LPC as feature extraction technique and speaker independent environment and LPC as feature extraction technique. It was observed that in all four cases as the number of speakers were increasing performance of system degrading. In speaker independent environment system performs badly and LPC giving poor results in all four cases. MFCCS work better than LPC in all four cases.

Mankala *et al.* (2014) proposed automatic speech recognizer for Telugu language using HTK toolkit. This was developed for recognizing isolated words using acoustic word model. Data was collected from nine Telugu speakers for training purpose and system was trained using 113 isolated Telugu words. The overall accuracy of the system that was observed was in the range of 95.46% and 96.64%.

2.2 Literature Survey on the Use of HMM in Speech

Lee *et al.* (1989) proposed speech recognition system using SPHINX tool. This tool is based on discrete HMMs and LPC as feature vectors. To train the 48 context independent phonetic HMMs, 4200 sentences were used spoken by 105 speakers. Phone HMMs were concatenated to build word HMMs which further concatenated to build large sentence HMM. It was observed that when words occur in cluster it becomes difficult to recognize them. Training was done in two stages: in first stage 48 context independent phonetic HMMs were trained and in second stage trained models from first stage initialized context dependent phone models. This system was evaluated on 150 sentences spoken by 15 speakers. With word-pair grammar, word

recognition accuracy of 96% was obtained and with null grammar word recognition accuracy was 82%.

Lee *et al.* (1989) proposed speaker independent phone recognition system based on discrete HMMs. This system was evaluated on TIMIT database consisting of 6300 sentences from 630 speakers. In this system, HMMs are trained using TIMIT sentences from 357 speakers and was evaluated on 160 TIMIT sentences from 20 speakers. LPC was used as a feature extraction technique. A new novel smoothing algorithm which smooth out the HMM output parameters was also introduced in this work. For 39 English phones, 64.07% was the recognition rate with context-independent phone models and with right context dependent phone models recognition rate was 73.80%.

Rabiner *et al.* (1989) proposed connected digit recognition system based on HMMs. Proposed system was trained and tested in three modes: speaker trained, multi-speaker and speaker independent. Evaluation was done on three databases: database widely distributed through National Bureau of Standards, 225 adult talker and 50 talker connected digit database. 0.78, 2.85 and 2.94 were the string error rate that were observed for all 3 modes.

Lamel *et al.* (1992) proposed speaker independent phoneme recognition system for continuous speech of French language. Data was collected from 43 speakers to provide training to HMMs and for testing from new 19 speakers data was collected using large read speech corpus BREF. For 35 context independent phoneme models, 60% of phone accuracy was obtained and with 428 context dependent models, 68.6% of phone accuracy was obtained.

Ratnayake *et al.* (1992) proposed speaker independent phoneme recognition based on hidden semi Markov models (HSMMs). In this work, HSMMs were used to overcome the limitation of HMMs. Instead of parametric distributions, non parametric distributions were used. LPC was used as the feature extraction technique. As a result, it was observed that with HSMMs phoneme recognition accuracy was 53.7% and with HMMs it was 48.4%. One drawback that was observed with HSMMs was that it has high computational complexity as compared to HMMs.

Brugnara *et al.* (1993) proposed automation of segmentation and labeling of speech of Italian language using HMMs. Training and performance evaluation both was done on TIMIT database. For training purpose 64 speakers were selected, eight sentences from each speaker. Performance

was evaluated on 24 different speakers, each speaker was asked to utter eight sentences. It was observed that manual segmentation done by expertise in phonetics provided 93.5% accuracy for locating correct positioned boundary which was not so far from 86.9% obtained by automatic segmentation system.

Kapadia *et al.* (1993) proposed phoneme recognition system based on continuous density monophone HMMs. HMMs were trained using Maximum Mutual Information (MMI) algorithm. In this work, comparison was made between ML and MMI training algorithms for both type of models, diagonal and full covariance models. Performance and implementation issues related to MMI training were also discussed. As a result, it was observed that as the complexity of models increases performance of MMI trained recognition system improves but the performance of ML trained recognition system decreases.

Angelini *et al.* (1994) proposed speaker independent speech recognition system for Italian language using continuous density HMMs. Using APASCI corpus, propose system was trained and tested. In this work, a set consisting of 38 context independent units was evaluated and two sets of other context dependent units were also considered which performed differently. Vocabulary of size of 3900 words read by 88 males and 88 female speakers was used consisting of most frequent Italian words. Performance was evaluated in terms of phone loop recognition accuracy and word loop recognition accuracy.

Leggetter *et al.* (1995) proposed maximum likelihood linear regression technique for speaker adaptation of continuous density, *i.e.*, Gaussian mixtures HMMs. Modeling of new speaker was improved using an initial speaker independent system by updating the HMM parameters. Experiments were performed on ARPA RM1 database using HMMs with continuous density mixtures output distribution and cross word triphones. It was observed that with supervised adaptation 37% error reduction was achieved and with unsupervised adaptation 32% of error reduction was achieved using 40 adaptation utterances.

Knill *et al.* (1996) investigated the use of Gaussian selection (GS) in Speech recognition systems using HMMs. In this work, an investigation was done to got the trade-off required between low computation and achieving good state likelihoods. Also, problem related to limited performance when beam search is applied was also addressed. It was observed that the GS introduces error

because of two reasons: firstly, the exclusion of significant components from the cluster and secondly, state flooring.

Mari *et al.* (1996) proposed a second order HMM for word and phone based continuous speech recognition. In this work, it was shown that second order HMM yield better performance than first order HMM. Data was collected from speech telephone corpus and experiments were done on spelled names over telephone. More than 4000 people were asked to spell their first and last name with and without pauses over telephone and their voice were recorded. This was the speaker independent system. For training purpose 1200 calls and for testing purpose 491 calls were selected. It was observed that the second order HMMs can achieve more than 69% of accuracy.

Ming *et al.* (1998) proposed phone recognition system for continuous speech signal based on Bayesian triphone models. In this work, a new statistical framework was introduced for building triphone models using models of less context dependency. This method somewhat was different from the previous models as it was based on the Bayesian principle not on the heuristic method. This system was used for the recognition of the 39 phones on the TIMIT database. Performance of proposed system was tested on two test set: core test set and complete test set. With core test case accuracy was 74.4% and with complete test case accuracy was 75.6%.

Sun *et al.* (1998) proposed a genetic algorithm to train HMMs for Speech recognition system and a comparison is made between the proposed algorithm and traditional training HMM algorithm (Baulm-Welch algorithm). This proposed algorithm was tested on recognition of isolated words. For training purpose 3000 words and for testing purpose 500 words were used. At the time of evaluation of recognition system, it was observed that with genetic algorithm accuracy of the system was 96.2% and with traditional training algorithm accuracy was 94.0% under the same conditions.

Zheng *et al.* (1999) proposed an Easytalk application, *i.e.,* Chinese dictation machine (CDM) . This application was developed for recognizing the large vocabulary speaker-independent continuous Chinese speech. CDM engine included automation of merging based syllable detection, frame synchronous search algorithms based on statistical knowledge, methods for rejecting and accepting the decisions, critical area percentage and syllable synchronous network

search. LPC was used as a feature extraction technique. In this application, it was observed that cepstrum was not the best feature. CMD achieved 98% accuracy for in-vocabulary commands and 95% accuracy for out-of-vocabulary commands.

Young (1999) presented acoustic modeling for large vocabulary continuous speech recognition (LVCSR). The objective of LVCSR was to transcribe input speech into an orthographic transcription. It was assumed that input speech consisted of sequence of words and using the language model probability of any specific word sequence could be determined. This was a N-gram model. MFCCs was used as a feature extraction technique. It was observed that to get the good phonetic discrimination, for each different context HMMs required to be trained and the most common context was tri-phone. Cross-word triphones provided the best modeling accuracy but too many parameters required to be computed. To overcome that, state-tying context was used. This involved the concept of mixture splitting.

Rao (2000) proposed a framework based on discrete HMMs. This framework was the speaker independent isolated digit voice recognition. Telephone quality speech data was recorded using modem interface and data was collected from 160 speakers for training purpose. To improve the proposed system performance fine tuning methods were also shown. The basic speech recognition system showed 92.16% overall accuracy.

Pruthi *et al.* (2000) proposed the implementation of speaker dependent real time isolated word recognizer for Hindi. This recognizer was named as swaranjali. This system was based on HMMs. Data was collected from two male speakers who were asked to utter Hindi digits from shoonya (0) to nau (9) two times means using total 20 tokens HMMs were trained. After training, evaluation was performed on the proposed system to check the accuracy. Because of presence of plosives at the beginning and end of some of the words some errors were also recognized. On an average 84.49% was the accuracy of system for speaker 1 and for speaker 2 it was 84.27%. Woodland *et al.* (2002) proposed a framework based on continuous density HMMs for providing the discriminative training to the large vocabulary speech recognition systems. To train HMMs the maximum mutual information estimation (MMIE) method was used. 265 hours of data was used as training data for conversational telephone speech transcription. In this, tri-phone and quinphone HMM parameters were estimated which led to reduction in word error rate for the transcription of conversational telephone speech. Also, a scheme which reduced the danger of

over training was also shown. This scheme was based on linear interpolation of MMIE and MLE objective functions.

Nwe *et al.* (2003) proposed a text independent method for emotion classification of the speech. This was based on discrete HMMs which was used as classifier and log frequency power coefficients (LFPC) was used to represent the speech signals. In this system, emotions were classified into six categories anger, disgust, fear, joy, sadness and surprise. Data was collected from twelve speakers, each of which was asked to utter 60 emotional utterances. A comparison of LFPC feature parameters was made with Linear Prediction Cepstral Coefficients (LPCC) and MFCC feature parameters. It was observed that LFPC as feature parameter showed better performance than other traditional feature parameters. Proposed system showed 78% average accuracy on evaluation.

Sheh *et al.* (2003) proposed a system for automating the chord segmentation and recognition based on Expectation Maximization (EM)-trained HMMs. In this work, using speech recognition tool such as HMMs automated chord transcription system was built. HMMs were used in this system for sequence recognition and were trained using EM algorithm. As training examples, only the chord sequences were given as input without requiring the precise timings of the chord changes which were computed automatically at the time of training only. Proposed system showed about 75% accuracy when evaluated on a small set of 20 Beatles songs.

Hasan *et al.* (2004) proposed speaker recognition system, a security system based on speaker identification. MFCCs was used as a feature extraction technique. This system was implemented using Matlab6.1 in windowsXP environment. Data corpus prepared for this particular system consisted of 21 speakers (13 male speakers and 8 female speakers). To evaluate the performance of the system identification rate was used as a measure which is the ratio of number of identified speakers to the total numbers of speakers tested. Identification rate was measured using three windows triangular, rectangular and hamming on two scales one was linear frequency scale and another was Mel-frequency scale. When linear frequency scale was used it was observed that identification rate was directly proportional to codebook size, as the codebook size increased identification rate was also increased and when the codebook size was 16 identification rate was 100% for both triangular and hamming windows. When Mel-frequency scale was used, relation between identification rate and codebook size was same as it was in the case of linear frequency

scale but in this case 100% identification rate was observed when size of codebook was 4 and hamming window was used.

Hyassat *et al.* (2006) proposed Arabic Speech Recognition. In this work, first SPHINX-IV based Arabic recognizer was introduced and an automatic toolkit was proposed capable of producing pronunciation dictionary (PD) for both Holly Qura'an and standard Arabic language. In this work, three corpus were developed completely : Holly Qura'an Corpus of about 18.5 hrs, command and control corpus (CAC-1) of about 1.5 hrs and Arabic digit corpus (ADC) of about less than one hour. For each corpus, three acoustic models were developed by providing the training to SPHINX-IV engine. Training was based on HMM model.

Sha *et al.* (2006) proposed a framework for phonetic recognition and classification using large margin Gaussian Mixture modeling (GMM). In large margin GMM each class was modeled by one or more ellipsoid. On both tasks, *i.e.*, phonetic classification and phonetic recognition a significant improvement was observed as compared to systems that were trained by maximum-likelihood estimation.

Satori *et al.* (2007) presented an approach for building an automated Speech recognition System for Arabic language was proposed. For building this system, utilities of Sphinx-4 engine were used which is the open source from Carnegie Mellon University (CMU). This tool is based on discrete HMMs. Difficulties that were faced in developing this system for Arabic language was that non-diacritized content was in larger amount, huge variety of dialectal and lastly, morphological complexity. This system was designed for recognizing the ten Arabic digits and this system was named as Hello_Arabic_Digit application. Data corpus prepared particularly for this system consisted of six male speakers who were asked to utter all ten digits five times. For training purpose, all 300 utterances were used. For checking the performance of trained system three different male speakers were asked to utter all ten Arabic digits. Mean recognition ratio for each one was computed for speaker 1 it was 86.66%, for speaker 2 it was 86.66% and for speaker 3 it was 83.33%.

Sha *et al.* (2007) proposed a new approach for providing discriminative training to continuous-density hidden Markov models (CD-HMM). In this work, two popular approaches (based on minimum classification error (MCE) and conditional maximum likelihood (CML)) were

compared to a new approach which was based on margin maximization. This new approach removed the problem of spurious local minima which was observed in other approaches as this approach lead to convex optimization over the parameter space of CD-HMMs. On TIMIT speech data corpus, phonetic recognizers were built using trained CD-HMMs from all three approaches. It was observed that new proposed approach was better than other two approaches as there was less phonetic error rate as compared to others.

Bhuriyakorn *et al.* (2008) presented phoneme recognition of continuous speech of Thai language. In this work, an approach of estimating HMM topology was proposed, whole process was divided into two stages: by combining different objective functions and topology generation methods a set of suitable topologies were constructed and a genetic algorithm was used as the topology selection algorithm considering global fitness. As a result, about 4.36% of error reduction in well-trained topologies was observed over already defined left-to-right HMM models.

Elshafei *et al.* (2008) proposed speaker independent natural Arabic speech recognition system. This system was based on HMMs and was developed using Sphinx tools. This system was tri-phone based acoustic model using five states HMMs in which first and the last state was non-emitting and others were emitting states. It used continuous density of eight Gaussian mixture distributions. Total 5.4 hrs of data was used for training and testing purpose out of which 4.3 hrs of data was used for training purpose and the remaining data 1.1 hrs was used for testing purpose. In pronunciation dictionary 14,232 words were defined and language model contained both bi-grams and tri-grams. After testing the system, word error rate was observed to be 9.0%.

Alotaibi (2008) proposed Arabic digit recognition system and did a comparative study of HMM and artificial neural network (ANN). Proposed system was implemented using HMM and was isolated word phoneme based recognizer. After evaluating the performances of both recognizers it was observed that ANN based recognizer obtained 99.5% accuracy in multi-speaker mode and 94.5% in speaker independent mode while HMM based recognizer obtained 98.1% accuracy in multi-speaker mode and 94.8% in speaker independent mode.

Jancovic *et al.* (2009) proposed a model to incorporate voicing information into a speech recognition system in noisy environment. By employing the Bernoulli distribution the voicing

information was modeled. This model was obtained for each HMM state and mixture using Viterbi-style training procedure. This model was evaluated within the standard model and other two models which had compensated for the noise effect (multi-conditional and missing feature training model). After incorporating the voicing information some performance improvement was achieved within standard model and noise compensated models as the SNR observed was 24.56% for standard model, 27.08% for missing feature training (MFT) model and 21.35% for multi-conditional training model. MFCC was used as a feature extraction technique.

Satori *et al.* (2009) a novel approach for building an automated Speech recognition System for Arabic language using Arabic environment was proposed. For building this system, utilities of Sphinx-4 engine were used which is the open source from Carnegie Mellon University (CMU). This tool is based on discrete HMMs. Difficulties that were faced in developing this system for Arabic language was that non-diacritized content was in larger amount, huge variety of dialectal and lastly, morphological complexity. Data corpus prepared particularly for this system consisted of 35 male speakers and 25 female speakers who were asked to utter all ten digits five times. For training purpose, all 3000 utterances were used. MFCCs was used as a feature extraction technique. For checking the performance of trained system three different male speakers and three different female speakers were asked to utter all ten Arabic digits. Mean recognition ratio for each one was computed for male speaker 1 it was 96.67%, for male speaker 2 it was 93.33% , for male speaker 3 it was 93.33%, for female speaker 1 it was 86.66%, for female speaker 2 it was 83.33% and for female speaker 3 it was 90.00%.

Kumar *et. al* (2010) proposed comparison between HMM and Dynamic Time warp (DTW) technique for speaker dependant isolated word recognition of Punjabi language. The DTW approach, the time warping technique was combined with linear predictive coding analysis and in HMM approach, Hidden Markov Modeling was combined with linear predictive coding analysis. The DTW used Nearest-Neighbor as the decision rule and HMM used the Maximum Likelihood (ML) as the decision rule. For implementing this system Visual C++ with multimedia API was used on widows platform. Data was collected from one male speaker. For the comparison between both techniques, codebook of size of 256 words was used. After making comparison, it was observed that DTW based recognizers showed better performance than HMM based recognizers because of the insufficiency of the training data but the time and space

complexity of HMM based approach was less as compared to DTW based approach. The overall accuracy of DTW recognizer was 92.3% and of HMM recognizer was 87.5% for Punjabi language numerals.

Abushariah *et. al* (2010) proposed English Digits Speech Recognition System based on HMMs. This system was developed using matlab. MFCC is used as a feature extraction technique. This system focused on all English digits from zero to nine. Two modules were implemented : isolated word speech recognition and the continuous speech recognition and both modules were tested in clean and noisy environment. In isolated word speech recognition tested in clean environment, multi-speaker mode and speaker independent mode achieved 99.5% and 79.5% accuracy and in noisy environment, multi-speaker mode and speaker independent mode achieved 88% and 67% accuracy. In continuous speech recognition tested in clean environment, multi-speaker mode and speaker independent mode achieved 72.5% and 56.25% accuracy and in noisy environment, multi-speaker mode and speaker independent mode achieved 82.5% and 76.67%. From this, it was observed that the in both environments multi-speaker mode performed better than the speaker independent mode.

Ghai *et al.* (2012) proposed automatic speech recognition system analysis for Indo-Aryan languages. For most of these languages, many of the researchers had worked for developing automatic speech recognition system except Punjabi language for which not enough work had been done in the same domain. In this work, analysis of recognizers of various Indo-Aryan languages was done and was discussed how it could be applicable to Punjabi language so that some work could be initiated.

Vimala *et. al* (2012) presented speaker independent speech recognition system for Tamil language. This system was developed for recognizing the isolated words and it was based on HMMs. MFCC was used as a feature extraction technique. This system was developed using sphinx-4 tool. The performance of system was measured in terms of word error rate. Data was collected from ten speakers out of which four speakers data were used for performance evaluation. The vocabulary size used for this work was 2,500 words and 88% of accuracy was observed with minimum word error rate of 0.88.

Manjunath *et al.* (2013) proposed two separate speaker dependent phonetic engines for two Indian languages: Bengali and Oriya. Two phonetic engines were developed for a set of 35 phones of Bengali language and for a set of 32 phones of Oriya language. Phonetic Engines were developed using HMMs and Feed Forward Neural Networks (FFNNs). For Bengali language, overall accuracy of phonetic engine was 41.65% using HMMs and 53.87% using FNNs and for Oriya language, it was 46.18% using HMMs and 59.88% using FFNNs.

CHAPTER 3

Phonetic Engine Development for Punjabi Language

Phonetic engine involves acoustic modeling that contains the statistical representation of distinct sounds and language modeling. Each of these statistical representation is assigned a label called 'phoneme'. **In this work, we have used 29 unique phonemes** excluding silence.

3.1 Requirements for System Implementation

Before proceeding for providing the training to monophone HMMs, all the speech data in the form of audio and their corresponding transcriptions must be prepared both for training and for testing purpose. All the recorded speech must not be more than 5sec to get the good accuracy.

3.1.1 List of Models (HMM List)

Prepare a list of all those phonemes whose HMM models are to be built. We have used unique 30 phonemes including silence. Figure 3.1. shows different phonemes that have been used.

Figure 3.1: HMM List

In HMM list, every defined HMM has two names: a physical name and a logical name. The physical name identifies the definition on disk and the logical name expresses the role of the model. Both physical and logical names are identical by default.

3.1.2 Pronunciation Dictionary

Pronunciation dictionary specifies the words pronunciations as the linear sequence of phonemes. Since, we are working at phoneme level, a file consisting of all phonemes with their corresponding pronunciation is prepared. If we work on word level then this file will consist of word and its corresponding pronunciation. It can be built easily from the sample sentences present in the training data. For example, if we talk about at word level, it will look like this:-

LAB l a b

BULB b u l b

and so on. The pronunciation is not case sensitive. If for a single word there are multiple pronunciations then there will be repeated entry for the word. For example:-

vakh v aa kh

vakh w aa kh

It should be noted that no blank line should be left after any line in the dictionary. At phoneme level, pronunciation dictionary looks like as shown in Figure 3.2. First column shows the phoneme and at word level it will be a word and second column shows the pronunciation of corresponding phoneme.

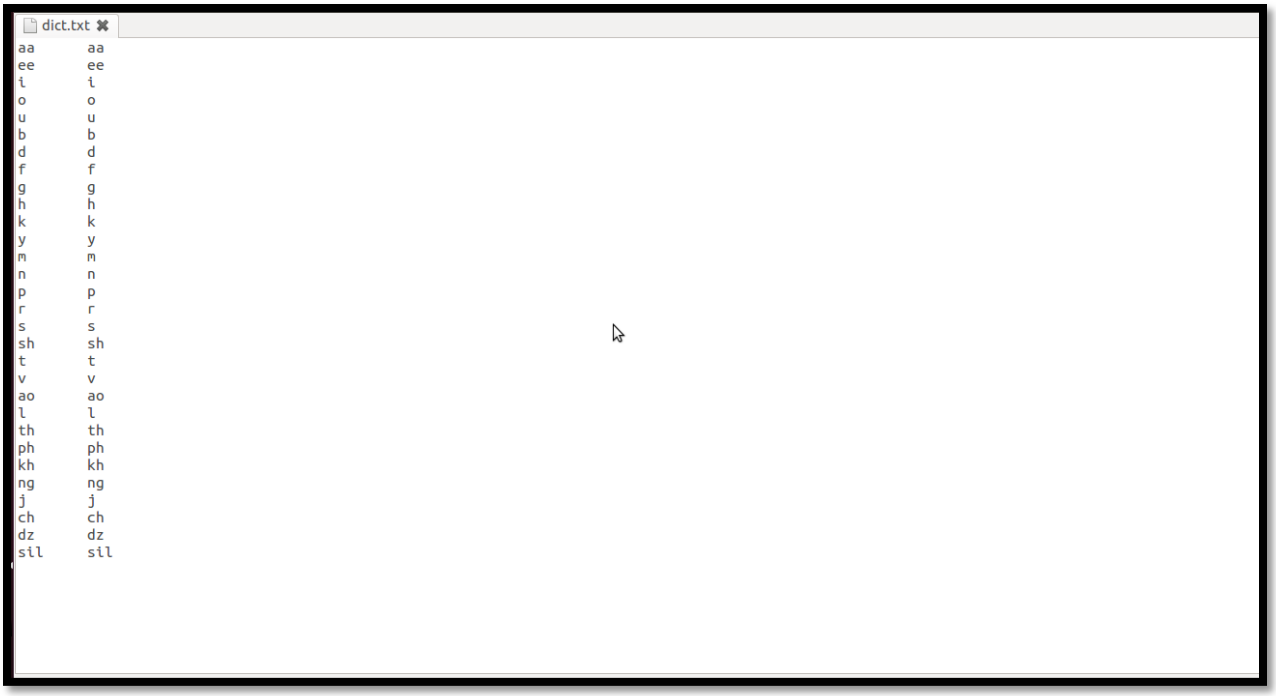


Figure 3.2: Pronunciation Dictionary

If we talk about word level, significance of using pronunciation dictionary is that when a speaker speaks out a word that need to be recognized, recognizer will listen the distinct sounds and looks for matching HMMs of each sound and then, determine the sequence of phones that make up a particular word based on training given to it and then these sequence of phonemes, *i.e.*, the pronunciation, are checked in dictionary, if entry exist then, the word mentioned against it, is picked up. The role of pronunciation dictionary is same at phoneme level.

3.1.3 Grammar

HTK basically requires a word network to get each word to word transition and each word instance. For this, a grammar definition language is provided by the HTK for specifying the simple task grammar. This grammar is processed by HTK and HTK creates a word network for itself. Grammar consists of some variable definitions followed by regular expressions. For processing this grammar, HTK uses its HParse tool. Figure 3.3 shows the working of HParse tool. This tool takes as input the 'grammar' and using this input it creates a word network.

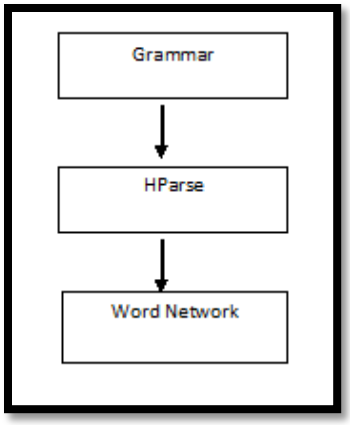


Figure 3.3: Working of HParse

The Grammar that has been used in this work is shown in Figure 3.4.

```
$WORD =  aa | ee | i | o | u | b | d | f | g | h | k | y | m | n | p | r | s | sh | t | v | ao | l | th |
ph | kh | ng | j | ch | dz ;
(<$WORD>)
```

Figure 3.4: Grammar File

where, vertical bars specify alternatives and angle braces specifies one or more repetitions. This complete grammar is converted by HTK into a network.

If we talk about word level, the significance of using task grammar is that when the word is picked up from the pronunciation dictionary, that word is checked against the grammar, if exists, then that word is shown to the user as a result. This same procedure applies at phoneme level.

3.1.4 Transcription File

Prepare a single transcription file at word level for each wav file such that it includes both wav file name and its corresponding transcription. This transcription is nothing but the spoken utterances in the speech audio file. IPA symbols defined in transcription file need to be converted such that all IPA symbols get replaced with their corresponding ASCII characters. This is done so because HTK understands only the ASCII characters. Transcription file that has been used in this work is given in Figure 3.5.

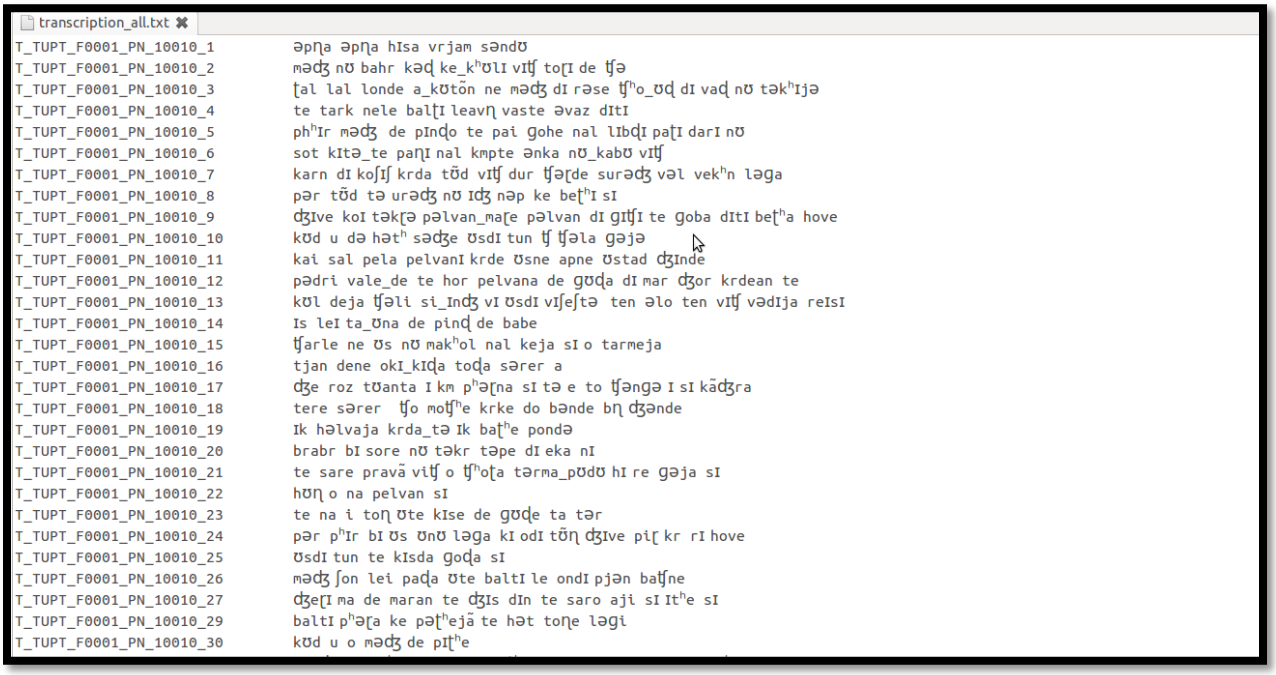


Figure 3.5: Transcription File

In Figure 3.5, first column shows speech audio file names and second column shows their corresponding transcribed spoken utterances.

Table 3.1 shows phonetic alphabets and their corresponding ASCII characters.

Table 3.1: Mapping between IPA and ASCII

Phonetic Alphabet	ASCII Character
a:	aa
a	aa
á	aa
à	aa

e:	ee
e	ee
è	ee
é	ee
i:	i
i	i
í	i
o:	o
o	o
ò	o
ó	o
u	u
ù	u
ú	u
b	b
d	d
d:	d
f	f
g	g
h	h
k	k
j	y
m	m
n	n
p	p
ɾ	r
r	r
s	s
ʃ	sh
t	t

v	v
ɖ	d
ə	ao
əː	ao
ə̀	ao
ə́	ao
ɭ	l
ʈ	t
t^h	th
$ʈ^h$	th
uː	u
ʊ	v
p^h	ph
k^h	kh
ŋ	ng
z	j

Some of the phonetic alphabets are substituted by the same ASCII character for getting the good accuracy as for a particular phone many examples needs to be provided to HMM at the time of training and examples related to phonetic alphabets with diacritic were short. So, phonetic alphabets with diacritics are substituted with phonetic alphabets without diacritics resembling almost the same sound.

3.1.5 Training and Testing Files

Prepare two master label files, one for training purpose and another for testing purpose at phoneme level such that a single line must consists of single phoneme. Format of both files should remain same. Format includes MLF header at the beginning of both files and the wav file names with the extension of .lab followed by the spoken utterances of each wav file with the 'sil' keyword at the beginning and end of each label file. Training file consists of only those label files which are to be used for training purpose and testing file consists of those label files which are to be recognized. Figure 3.6 shows the format of both files that have been used in this work.

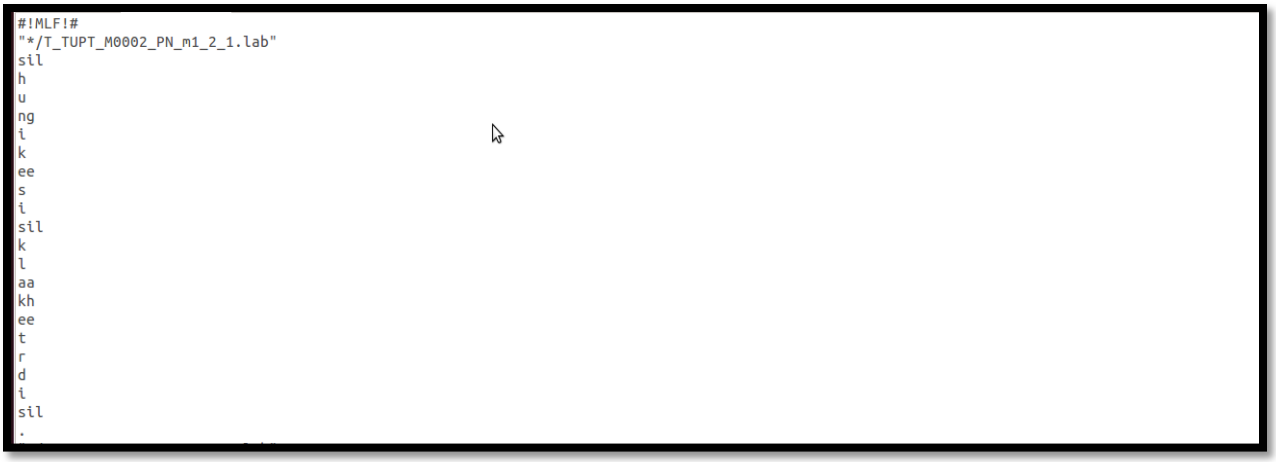


Figure 3.6: Format of Training and Testing Files

3.1.6 Feature Extraction

A sequence of feature vectors are extracted from each wav file. For doing this, a configuration file specifying all the parameters to be applied on each wav file and what features are to be extracted and a list of wav files of which features are to be extracted are provided as input to HTK and HTK automatically extracts the features using all the parameters specified in configuration file. HTK works only on the Mel-frequency Cepstral Coefficients (MFCCs). Feature vectors extracted in this work includes 13 dimensional MFCCs, 13 dimensional velocity and 13 dimensional acceleration parameters. HTK do this using HCopy tool.

3.1.7 Prototype Model

For providing the training to HMMs, it is essential to define a prototype model on the basis of which it will compute the parameters. For phoneme based system, a good topology to use is 3-state left-right topology. At the beginning of training, the prototype is used by HCOMPV tool of HTK toolkit to find the global mean and variance such that at the initial level all HMMs are initialized with these global parameters. The function of this prototype definition to define the form and topology of HMM. The possible transitions between states are indicated by putting non-zero values in the corresponding elements of the transition matrix and zeros elsewhere. In transition matrix, each row must sum to one except for the final row which must be all zero. Diagonal variance must always be positive but all mean values can be zero. Prototype model that has been used in this work is given in Figure 3.7.

Figure 3.7: Prototype Model

3.2 Training and Testing of the System

The general framework of phonetic engine is shown in Figure 3.8.

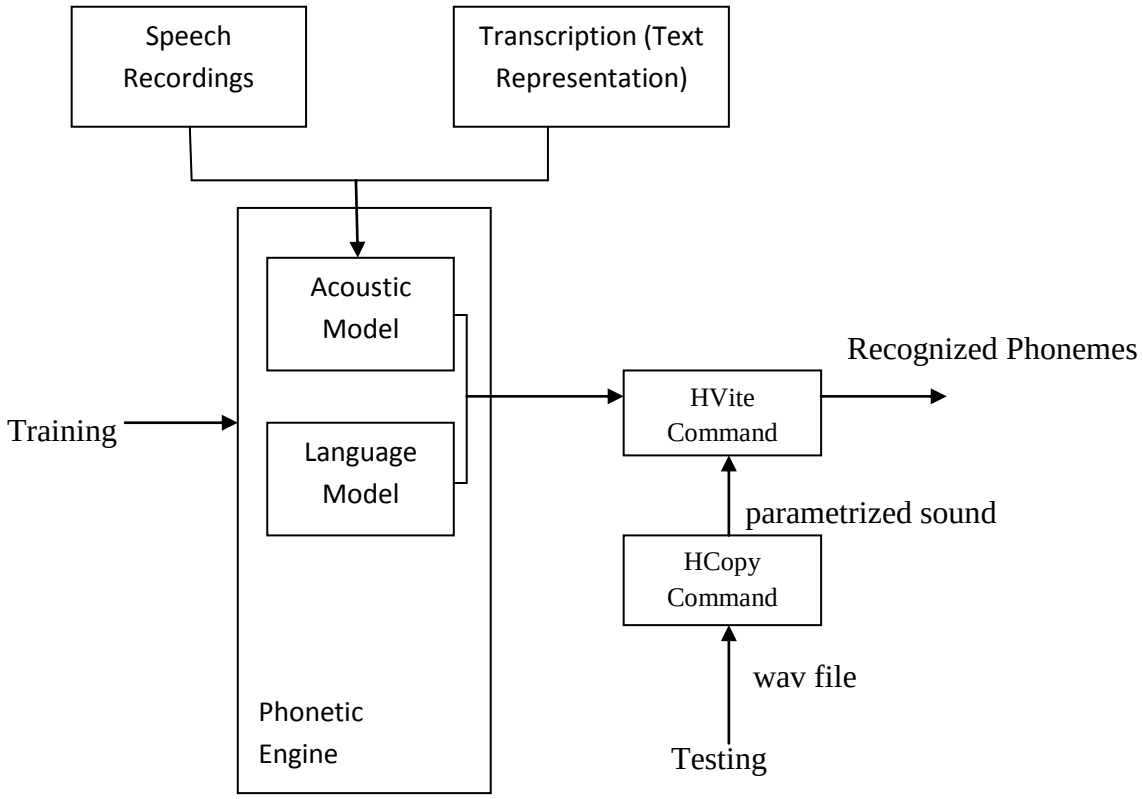


Figure 3.8: General Framework of Phonetic Engine

3.2.1 Training

The process of estimating the parameters of HMM from the examples of data sequences called training. For providing the training to HMMs for each phone, many examples of each phoneme at least 3 should be there in training data. More examples, more will be the accuracy of system at the time of recognition.

3.2.1.1 Components Required for Training

For training, following files are required.

i. Training file (say, train.mlf).
ii. A file consisting of path of MFCC features of all wav files used for training (say, train.list).
iii. List of Models (say, hmmlist.txt).
iv. configuration file (say, analysis_train.conf).
v. Prototype Model (say, proto.txt).

3.2.1.2 Training Algorithm

Step1- No. of mixtures (*MIXTURES*) = 32 and No. of states (*NUM_STATES*) = 5.

Step 2- Create Hmm folders from hmm0 to hmm32.

Repeat for *i* = 0 to *MIXTURES*

mkdir ./work_all_3_train_read_test_feb_2014$*NUM_STATES*/hmm$i

Step 3- Calculate the global mean and variance according to given prototype model.

HCompV -T 1 -D -A -C ./analysis_train.conf -I ./train.mlf -f 0.01 -m -S ./train.list -M

./work_all_3_train_read_test_feb_2014$*NUM_STATES*/hmm0 ./proto.txt

Step 4- Generates the hmmdefs (master macro file) and macros for hmm0 using generated proto and vfloors files where, hmmdefs consists of parameters computed of all the models listed in hmmlist.txt.

Step 5- Re-estimate the parameters of all the models defined in hmmdef of hmm0 using HERest re-estimation tool and store the output in a new directory hmm1.

Step 6- Increment mixture size followed by re-estimation of parameters using re-estimation tool.

REPEAT for *i* = 2 to *MIXTURES*

HHEd -T 1 -D -A -C ./analysis_train.conf -H ./work_all_3_train_read_test_feb_2014 $*NUM* _ *STATES*/hmm$((i-1))/macros -H ./work_all_3_train_read_test_feb_2014$*NUM_STATES* /hmm$((i-1))/hmmdefs -M ./work_all_3_train_read_test_feb_2014NUM_STATES/hmmi ./work_all_3_train_read_test_feb_2014$NUM_STATES/mix_$i.hed ./hmmlist.txt

REPEAT for *i* = 1 to 6

HERest -T 1 -D -A -C ./analysis_train.conf -I ./train.mlf -t 250.0 150.0 1000.0 -S ./train.list -H ./work_all_3_train_read_test_feb_2014$*NUM_STATES*/hmm$i/macros -H ./work_all_3_ train _read_test_feb_2014$*NUM_STATES*/hmm$i/hmmdefs -M ./work_all_3_train_read_test_feb_ 2014$*NUM_STATES*/hmm$i ./hmmlist.txt

Step 7- END.

3.2.2 Testing

After providing the training, the performance of trained system can be checked using the HTK analysis tool 'HResults'.

3.2.2.1 Components Required for Testing

Files required at the time of evaluation are:-

i. List of Models (say, hmmlist.txt).
ii. Pronunciation dictionary (say, dict.txt).
iii. Configuration file (analysis_train.conf).
iv. A file consisting of path of MFCC features of wav files that are to be recognized (say, test.list).
v. Grammar (say, gram.txt).
vi. Master Macro File in which 32 mixtures of each state of each HMM model are computed (say, hmmdef.txt).

3.2.2.2 Testing Algorithm

Step 1- Create word network (wdnet.txt) using HParse tool.

HParse gram.txt wdnet.txt

Step 2- No. of states (*NUM_STATES*) = 5.

Step 3- No. of Mixtures = 32.

Step 4- Recognizing the test data.

HVite -T 1 -D -A -H ./work_all_3_train_read_test_feb_2014NUM_STATES /hmmi /macros -H ./work_all_3_train_read_test_feb_2014$*NUM_STATES*/hmm$i/hmmdefs -S ./test.list -C ./analysis_train.conf -I ./test.mlf -i ./work_all_3_train_read_test_feb_2014$*NUM_STATES* /hmm$i /recout_ test.mlf -o SWT -w ./wdnet.txt -p -10.0 -s 0 ./dict.txt ./hmmlist.txt

Step 5- Determine the actual performance by comparing desired result and actual result generated by recognizer.

HResults -p -I ./test.mlf ./hmmlist.txt ./work_all_3_train_read_test_feb_2014$*NUM_STATES* /hmm$i/recout_test.mlf >> ./work_all_3_train_read_test_feb_2014$*NUM_STATES* /hmm$i/result_test

Step 6- END.

CHAPTER 4

Three Modes of Data Collection and Results

After applying the algorithm discussed in previous chapter the output is in the form of confusion matrix which is resultant of comparison of desired output and actual output.

4.1 Data Collection

Data has been collected in three different modes, namely, read speech mode, lecture mode and conversation mode.

i. **Read Speech Mode:**

In this mode, data has been collected from four native Punjabi speakers (2 males ad 2 females) for a duration of about 3 hours. Recording has been done in two types of environment: in studio environment using a microphone channel maintained at a sampling frequency of 48 KHZ and in normal room environment using a microphone channel maintained at a sampling frequency of 22050 HZ. For recording the data, Punjabi speakers read out the paragraphs from Punjabi books.

ii. **Lecture Mode:**

In this mode, data has been collected from a radio channel, Punjabi radio USA. The recording in this mode has been done with a sampling frequency of 48 KHZ and a bit rate of 16 bits per sample.

iii. **Conversation Mode:**

In this mode, the conversation and discussion of Punjabi speakers have been recorded. The recording has been done with the same sampling frequency and bits per sample as in lecture mode of speech. It has been done in two types of environment: open environment and closed environment.

The total duration of data collected for each mode is shown in Table 4.1.

Table 4.1: Total Duration of each Mode of Speech

Mode	Total Duration(in Minutes)
Read Speech Mode	185.93
Lecture Mode	20.05
Conversation Mode	20.22

4.2 Performance Evaluation

PE developed has been trained and tested for the following cases: read speech mode, lecture mode, conversation mode, for each gender in read speech mode and for each speaker in read speech mode. For each case, 75% of data has been used for training purpose and 25% of data has been used for testing purpose. The results have been obtained in the form of confusion matrix which is generated by comparing the desired result with the actual result. This matrix has been generated for each case separately which shows the overall accuracy of trained PE and other related information such as total number of phonemes in testing data, total number of phonemes substituted *etc*. Total number of rows in generated confusion matrix are 30 as the total number of unique phonemes including silence we have used are 30 and total number of columns are 29 excluding the silence. Each row represents the instances in actual class and each column represents the instances in predicted class. All correct guesses are shown diagonally in the matrix.

To compute the accuracy of trained system given formulae is used.

$$Accuracy = (N - S - D - I)/N \quad \text{... (4.1)}$$

where, *N* is the total number of phonemes, *S* is the number of phonemes substituted, *D* is the number of phonemes deleted and *I* is the number of phonemes inserted.

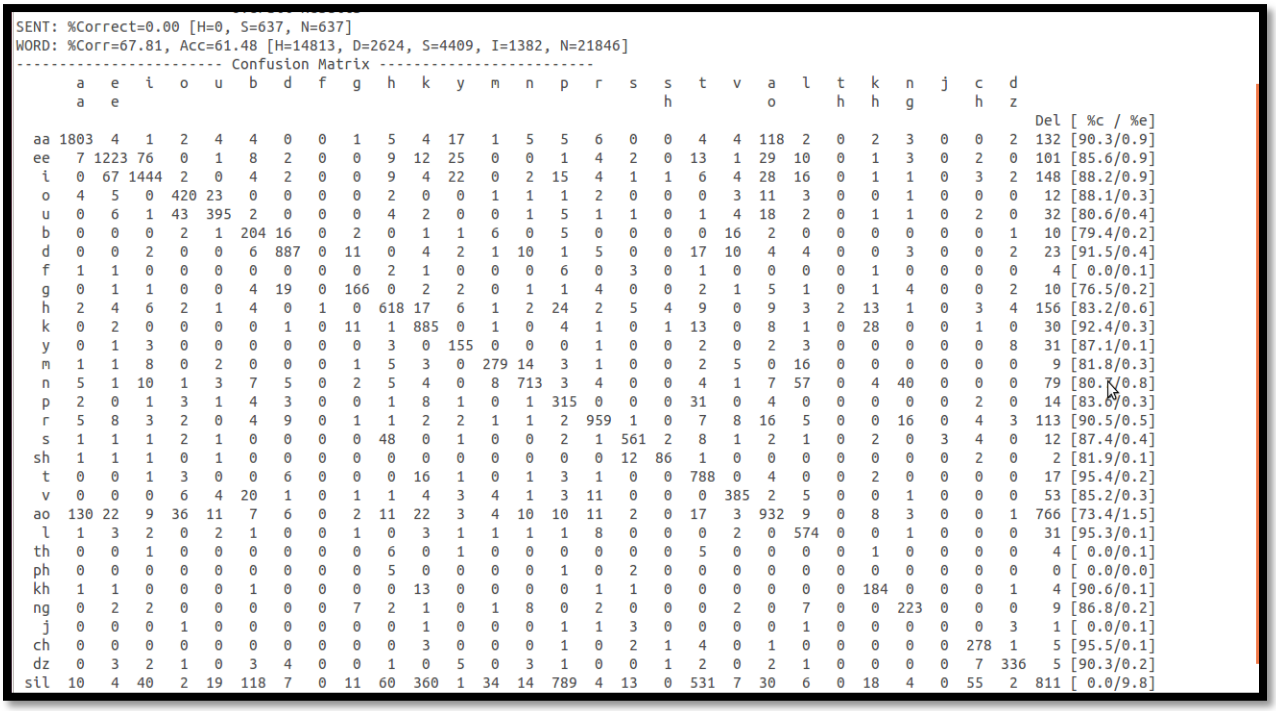
The percentage number of phonemes correctly recognized is computed using the given formula.

$$\%Correct = (H/N) * 100 \quad \text{... (4.2)}$$

where, '*H*' is the total number of phonemes correctly recognized.

Following are the results for the testing accuracy of the PE developed in this work.

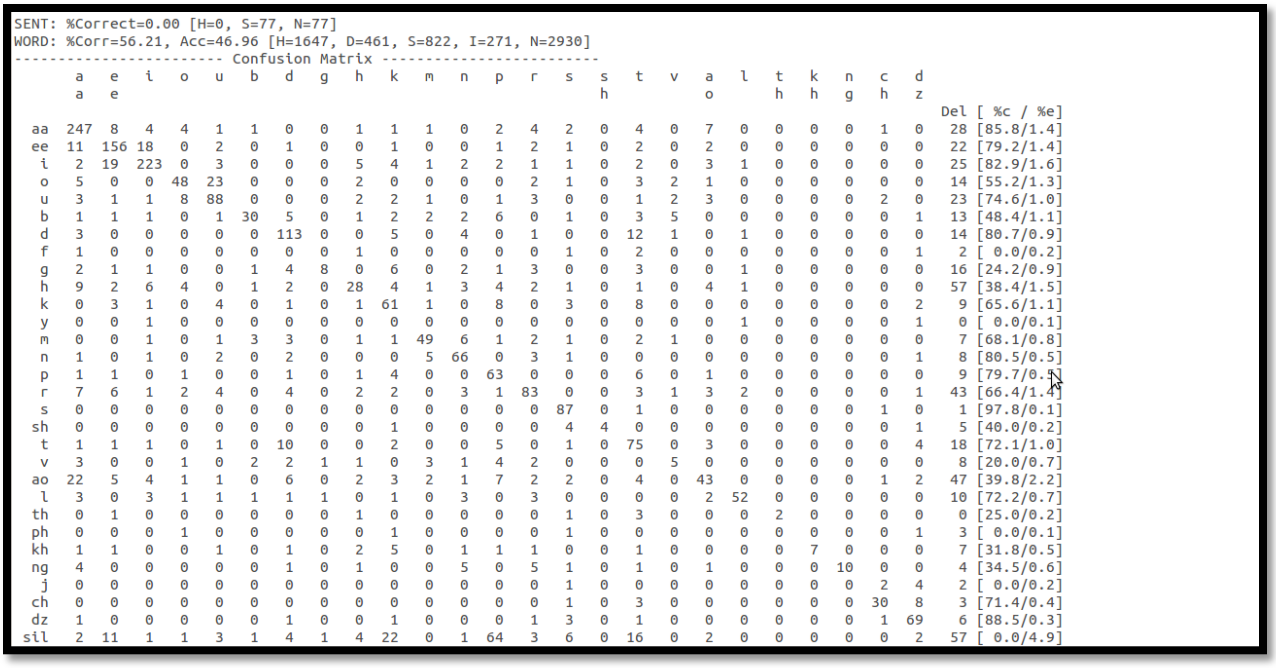
Figure 4.1, Figure 4.2 and Figure 4.3 show the confusion matrix generated after testing the PE in read speech mode, lecture mode and conversation mode.



```
SENT: %Correct=0.00 [H=0, S=637, N=637]
WORD: %Corr=67.81, Acc=61.48 [H=14813, D=2624, S=4409, I=1382, N=21846]
---------------------- Confusion Matrix -----------------------
       a   e   i   o   u   b   d   f   g   h   k   y   m   n   p   r   s   s   t   v   a   l   t   k   n   j   c   d
       a   e                                                   h           o       h   h   g       h   z
                                                                                                          Del [ %c / %e]
  aa 1803  4   1   2   4   4   0   0   1   5   4  17   1   5   5   6   0   0   4   4 118   2   0   2   3   0   0   2  132 [90.3/0.9]
  ee    7 1223 76   0   1   8   2   0   0   9  12  25   0   0   1   4   2   0  13   1  29  10   0   1   3   0   2   0  101 [85.6/0.9]
   i    0  67 1444  2   0   4   2   0   0   9   4  22   0   2  15   4   1   1   6   4  28  16   0   1   1   0   3   2  148 [88.2/0.9]
   o    4   5   0 420  23   0   0   0   0   2   0   0   1   1   1   2   0   0   0   3  11   3   0   0   1   0   0   0   12 [88.1/0.3]
   u    0   6   1  43 395   2   0   0   0   4   2   0   0   1   5   1   1   0   1   4  18   2   0   1   1   0   2   0   32 [80.6/0.4]
   b    0   0   0   2   1 204  16   0   2   0   1   1   6   0   5   0   0   0   0  16   2   0   0   0   0   0   0   1   10 [79.4/0.2]
   d    0   0   2   0   0   6 887   0  11   0   4   2   1  10   1   5   0   0  17  10   4   4   0   0   3   0   0   2   23 [91.5/0.4]
   f    1   1   0   0   0   0   0   0   0   2   1   0   0   0   6   0   3   0   1   0   0   0   0   1   0   0   0   0    4 [ 0.0/0.1]
   g    0   1   1   0   0   4  19   0 166   0   2   2   0   1   1   4   0   0   2   1   5   1   0   1   4   0   0   2   10 [76.5/0.2]
   h    2   4   6   2   1   4   0   1   0 618  17   6   1   2  24   2   5   4   9   0   9   3   2  13   1   0   3   4  156 [83.2/0.6]
   k    0   2   0   0   0   0   1   0  11   1 885   0   1   0   4   1   0   1  13   0   8   1   0  28   0   0   1   0   30 [92.4/0.3]
   y    0   1   3   0   0   0   0   0   0   3   0 155   0   0   0   1   0   0   2   0   2   3   0   0   0   0   0   8   31 [87.1/0.1]
   m    1   1   8   0   2   0   0   0   1   5   3   0 279  14   3   1   0   0   2   5   0  16   0   0   0   0   0   0    9 [81.8/0.3]
   n    5   1  10   1   3   7   5   0   2   5   4   0   8 713   3   4   0   0   4   1   7  57   0   4  40   0   0   0   79 [80.7/0.8]
   p    2   0   1   3   1   4   3   0   0   1   8   1   0   1 315   0   0   0  31   0   4   0   0   0   0   0   2   0   14 [83.6/0.3]
   r    5   8   3   2   0   4   9   0   1   1   2   2   1   1   2 959   1   0   7   8  16   5   0   0  16   0   4   3  113 [90.5/0.5]
   s    1   1   1   2   1   0   0   0   0  48   0   1   0   0   2   1 561   2   8   1   2   1   0   2   0   3   4   0   12 [87.4/0.4]
  sh    1   1   1   0   1   0   0   0   0   0   0   0   0   0   0   0  12  86   1   0   0   0   0   0   0   0   2   0    2 [81.9/0.1]
   t    0   0   1   3   0   0   6   0   0   0  16   1   0   1   3   1   0   0 788   0   4   0   0   2   0   0   0   0   17 [95.4/0.2]
   v    0   0   0   6   4  20   1   0   1   1   4   3   4   1   3  11   0   0   0 385   2   5   0   0   1   0   0   0   53 [85.2/0.3]
  ao  130  22   9  36  11   7   6   0   2  11  22   3   4  10  10  11   2   0  17   3 932   9   0   8   3   0   0   1  766 [73.4/1.5]
   l    1   3   2   0   2   1   0   0   1   0   3   1   1   1   1   8   0   0   0   2   0 574   0   0   1   0   0   0   31 [95.3/0.1]
  th    0   0   1   0   0   0   0   0   0   6   0   1   0   0   0   0   0   0   5   0   0   0   0   1   0   0   0   0    4 [ 0.0/0.1]
  ph    0   0   0   0   0   0   0   0   0   5   0   0   0   0   1   0   2   0   0   0   0   0   0   0   0   0   0   0    0 [ 0.0/0.0]
  kh    1   1   0   0   0   1   0   0   0   0  13   0   0   0   0   1   1   0   0   0   0   0   0 184   0   0   0   1    4 [90.6/0.1]
  ng    0   2   2   0   0   0   0   0   7   2   1   0   1   8   0   2   0   0   0   2   0   7   0   0 223   0   0   0    9 [86.8/0.2]
   j    0   0   0   1   0   0   0   0   0   0   1   0   0   0   1   1   3   0   0   0   0   1   0   0   0   0   0   3    1 [ 0.0/0.1]
  ch    0   0   0   0   0   0   0   0   0   0   3   0   0   0   1   0   2   1   4   0   1   0   0   0   0   0 278   1    5 [95.5/0.1]
  dz    0   3   2   1   0   3   4   0   0   1   0   5   0   3   1   0   0   1   2   0   2   1   0   0   0   0   7 336    5 [90.3/0.2]
 sil   10   4  40   2  19 118   7   0  11  60 360   1  34  14 789   4  13   0 531   7  30   6   0  18   4   0  55   2  811 [ 0.0/9.8]
```

Figure 4.1: Confusion Matrix- Read Speech Mode

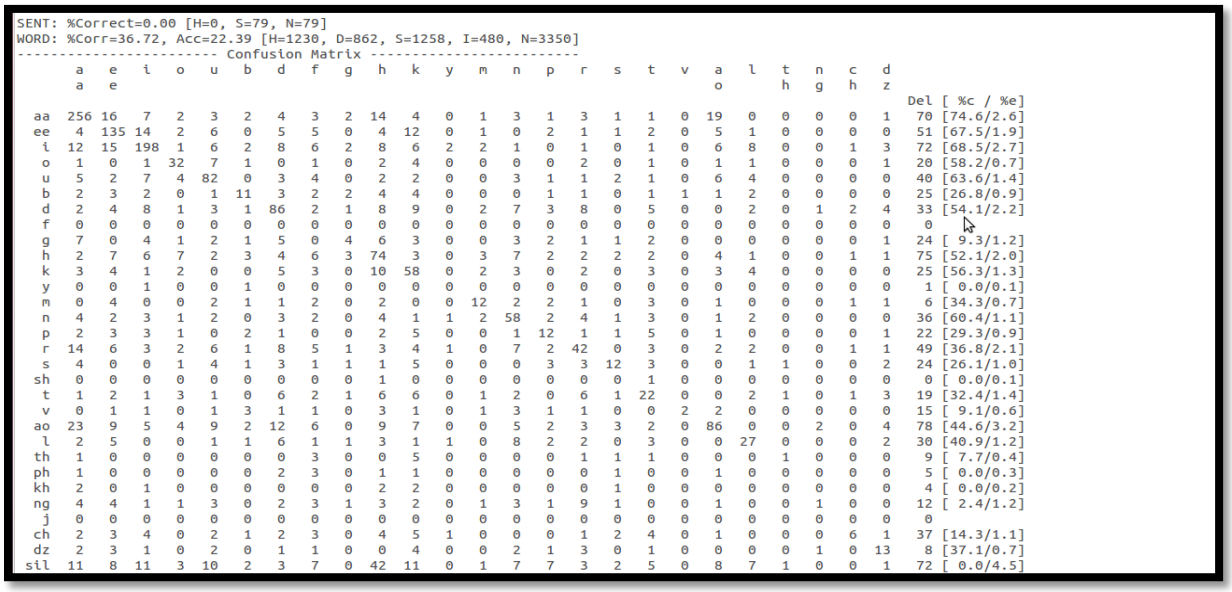
In Figure 4.1, each cell represents number of times an actual class is predicted as another class. For example, row 1 labeled as '*aa*' is predicted as '*ee*' four times, as '*i*' one time and so on. First line shows the sentence level accuracy based on total number of label files which are identical to transcription files, in this mode it is 0.00%. The second line shows the word accuracy based on matches between label files and transcriptions. In this mode, *i.e.*, read speech mode, correctness percentage is 67.81% and accuracy percentage is 61.48%. Total number of phonemes substituted (*S*) are 4409, total number of phonemes deleted (*D*) are 2624, number of phones inserted (*I*) are 1382, total number of phonemes correctly recognized (*H*) are 14813 and total number of phonemes (*N*) are 21846. '*%c*' shows how many times a phoneme instance has been correctly labeled, *i.e.*, percentage correct in the row (number of correct instances divided by the total number of instances in the row) and '*%e*' shows percentage of incorrectly labeled phonemes in the row as a percentage of total number of phonemes (*N*).



```
SENT: %Correct=0.00 [H=0, S=77, N=77]
WORD: %Corr=56.21, Acc=46.96 [H=1647, D=461, S=822, I=271, N=2930]
---------------------- Confusion Matrix -------------------------
      a   e   i   o   u   b   d   g   h   k   m   n   p   r   s   s   t   v   a   l   t   k   n   c   d
      a   e                                                   h           o       h   h   g   h   z
                                                                                                    Del [ %c / %e]
  aa  247  8   4   4   1   1   0   0   1   1   1   0   2   4   2   0   4   0   7   0   0   0   0   1   0   28 [85.8/1.4]
  ee   11 156 18   0   2   0   1   0   0   1   0   0   1   2   1   0   2   0   2   0   0   0   0   0   0   22 [79.2/1.4]
   i    2  19 223  0   3   0   0   0   5   4   1   2   2   1   1   0   2   0   3   1   0   0   0   0   0   25 [82.9/1.6]
   o    5   0   0 48  23   0   0   0   2   0   0   0   0   2   1   0   3   2   1   0   0   0   0   0   0   14 [55.2/1.3]
   u    3   1   1   8 88   0   0   0   2   2   1   0   1   3   0   0   1   2   3   0   0   0   0   2   0   23 [74.6/1.0]
   b    1   1   1   0   1 30   5   0   1   2   2   2   6   0   1   0   3   5   0   0   0   0   0   0   1   13 [48.4/1.1]
   d    3   0   0   0   0   0 113  0   0   5   0   4   0   1   0   0 12   1   0   1   0   0   0   0   0   14 [80.7/0.9]
   f    1   0   0   0   0   0   0   0   1   0   0   0   0   0   1   0   2   0   0   0   0   0   0   0   1    2 [ 0.0/0.2]
   g    2   1   1   0   0   1   4   8   0   6   0   2   1   3   0   0   3   0   0   1   0   0   0   0   0   16 [24.2/0.9]
   h    9   2   6   4   0   1   2   0 28   4   1   3   4   2   1   0   1   0   4   1   0   0   0   0   0   57 [38.4/1.5]
   k    0   3   1   0   4   0   1   0   1 61   1   0   8   0   3   0   8   0   0   0   0   0   0   0   2    9 [65.6/1.1]
   y    0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   1    0 [ 0.0/0.1]
   m    0   0   1   0   1   3   3   0   1   1 49   6   1   2   1   0   2   1   0   0   0   0   0   0   0    7 [68.1/0.8]
   n    1   0   1   0   2   0   2   0   0   0   5 66   0   3   1   0   0   0   0   0   0   0   0   0   1    8 [80.5/0.5]
   p    1   1   0   1   0   0   1   0   1   4   0   0 63   0   0   0   6   0   1   0   0   0   0   0   0    9 [79.7/0.5]
   r    7   6   1   2   4   0   4   0   2   2   0   3   1 83   0   0   3   1   3   2   0   0   0   0   1   43 [66.4/1.4]
   s    0   0   0   0   0   0   0   0   0   0   0   0   0   0 87   0   1   0   0   0   0   0   0   1   0    1 [97.8/0.1]
  sh    0   0   0   0   0   0   0   0   0   1   0   0   0   0   4   4   0   0   0   0   0   0   0   0   1    5 [40.0/0.2]
   t    1   1   1   0   1   0 10   0   0   2   0   0   5   0   1   0 75   0   3   0   0   0   0   0   4   18 [72.1/1.0]
   v    3   0   0   1   0   2   2   1   1   0   3   1   4   2   0   0   0   5   0   0   0   0   0   0   0    8 [20.0/0.7]
  ao   22   5   4   1   1   0   6   0   2   3   2   1   7   2   2   0   4   0 43   0   0   0   0   1   2   47 [39.8/2.2]
   l    3   0   3   1   1   1   1   1   0   1   0   3   0   3   0   0   0   0   2 52   0   0   0   0   0   10 [72.2/0.7]
  th    0   1   0   0   0   0   0   0   1   0   0   0   0   0   1   0   3   0   0   0   2   0   0   0   0    0 [25.0/0.2]
  ph    0   0   0   1   0   0   0   0   0   1   0   0   0   0   0   1   0   0   0   0   0   0   0   0   1    3 [ 0.0/0.1]
  kh    1   1   0   0   1   0   1   0   2   5   0   1   1   1   0   0   1   0   0   0   0   7   0   0   0    7 [31.8/0.5]
  ng    4   0   0   0   0   0   0   1   0   1   0   0   5   0   5   1   0   1   0   1   0   0   0 10   0    4 [34.5/0.6]
   j    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   2    4 [ 0.0/0.2]
  ch    0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   3   0   0   0   0   0   0   0 30    8 [71.4/0.4]
  dz    1   0   0   0   0   0   0   1   0   0   1   0   0   0   1   3   0   1   0   0   0   0   0   0   1   69   6 [88.5/0.3]
 sil    2  11   1   1   3   1   4   1   4  22   0   1  64   3   6   0  16   0   2   0   0   0   0   0   2   57 [ 0.0/4.9]
```

Figure 4.2: Confusion Matrix- Lecture Mode

In Figure 4.2, sentence level accuracy is 0.00% and percentage of correctly recognized phonemes is 56.21% and accuracy of PE is 46.96%. Total number of phonemes substituted (*S*) are 822, total number of phonemes deleted (*D*) are 461, number of phones inserted (*I*) are 271, total number of phonemes correctly recognized (*H*) are 1647 and total number of phonemes (*N*) are 2930.



```
SENT: %Correct=0.00 [H=0, S=79, N=79]
WORD: %Corr=36.72, Acc=22.39 [H=1230, D=862, S=1258, I=480, N=3350]
---------------------- Confusion Matrix -------------------------
      a   e   i   o   u   b   d   f   g   h   k   y   m   n   p   r   s   t   v   a   l   t   n   c   d
      a   e                                                                   o       h   g   h   z
                                                                                                    Del [ %c / %e]
  aa  256 16   7   2   3   2   4   3   2  14   4   0   1   3   1   3   1   1   0  19   0   0   0   0   1   70 [74.6/2.6]
  ee   4  135 14   2   6   0   5   5   0   4  12   0   1   0   2   1   1   2   0   5   1   0   0   0   0   51 [67.5/1.9]
   i  12  15 198  1   6   2   8   6   2   8   6   2   2   1   0   1   0   1   0   6   8   0   0   1   3   72 [68.5/2.7]
   o   1   0   1 32   7   1   0   1   0   2   4   0   0   0   0   2   0   1   0   1   1   0   0   0   1   20 [58.2/0.7]
   u   5   2   7   4 82   0   3   4   0   2   2   0   0   3   1   1   2   1   0   6   4   0   0   0   0   40 [63.6/1.4]
   b   2   3   2   0   1 11   3   2   2   4   4   0   0   0   1   1   0   1   1   1   2   0   0   0   0   25 [26.8/0.9]
   d   2   4   8   1   3   1 86   2   1   8   9   0   2   7   3   8   0   5   0   0   2   0   1   2   4   33 [54.1/2.2]
   f   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0
   g   7   0   4   1   2   1   5   0   4   6   3   0   0   3   2   1   1   2   0   0   0   0   0   0   1   24 [ 9.3/1.2]
   h   2   7   6   7   2   3   4   6   3 74   3   0   3   7   2   2   2   2   0   4   1   0   0   1   1   75 [52.1/2.0]
   k   3   4   1   2   0   0   5   3   0 10  58   0   2   3   0   2   0   3   0   3   4   0   0   0   0   25 [56.3/1.3]
   y   0   0   1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    1 [ 0.0/0.1]
   m   0   4   0   0   2   1   1   2   0   2   0   0 12   2   2   1   0   3   0   1   0   0   0   1   1    6 [34.3/0.7]
   n   4   2   3   1   2   0   3   2   0   4   1   1   2 58   2   4   1   3   0   1   2   0   0   0   0   36 [60.4/1.1]
   p   2   3   3   1   0   2   1   0   0   2   5   0   0   1 12   1   1   5   0   1   0   0   0   0   1   22 [29.3/0.9]
   r  14   6   3   2   6   1   8   5   1   3   4   1   0   7   2 42   0   3   0   2   2   0   0   1   1   49 [36.8/2.1]
   s   4   0   0   1   4   1   3   1   1   1   5   0   0   0   3   3 12   3   0   0   1   1   0   0   2   24 [26.1/1.0]
  sh   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   1   0   0   0   0   0   0   0    0 [ 0.0/0.1]
   t   1   2   1   3   1   0   6   2   1   6   6   0   1   2   0   6   1 22   0   0   2   1   0   1   3   19 [32.4/1.4]
   v   0   1   1   0   1   3   1   1   0   3   1   0   1   3   1   1   0   0   2   2   0   0   0   0   0   15 [ 9.1/0.6]
  ao  23   9   5   4   9   2 12   6   0   9   7   0   0   5   2   3   3   2   0 86   0   0   2   0   4   78 [44.6/3.2]
   l   2   5   0   0   1   1   6   1   1   3   1   1   0   8   2   2   0   3   0   0 27   0   0   0   2   30 [40.9/1.2]
  th   1   0   0   0   0   0   0   3   0   0   5   0   0   0   0   1   1   1   0   0   0   1   0   0   0    9 [ 7.7/0.4]
  ph   1   0   0   0   0   0   2   3   0   1   1   0   0   0   0   0   1   0   0   1   0   0   0   0   0    5 [ 0.0/0.3]
  kh   2   0   1   0   0   0   0   0   0   2   2   0   0   0   0   0   1   0   0   0   0   0   0   0   0    4 [ 0.0/0.2]
  ng   4   4   1   1   3   0   2   3   1   3   2   0   1   3   1   9   1   0   0   1   0   0   1   0   0   12 [ 2.4/1.2]
   j   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0
  ch   2   3   4   0   2   1   2   3   0   4   5   1   0   0   0   1   2   4   0   1   0   0   0   6   1   37 [14.3/1.1]
  dz   2   3   1   0   2   0   1   1   0   0   4   0   0   2   1   3   0   1   0   0   0   0   1   0  13    8 [37.1/0.7]
 sil  11   8  11   3  10   2   3   7   0  42  11   0   1   7   7   3   2   5   0   8   7   1   0   0   1   72 [ 0.0/4.5]
```

Figure 4.3: Confusion Matrix- Conversation Mode

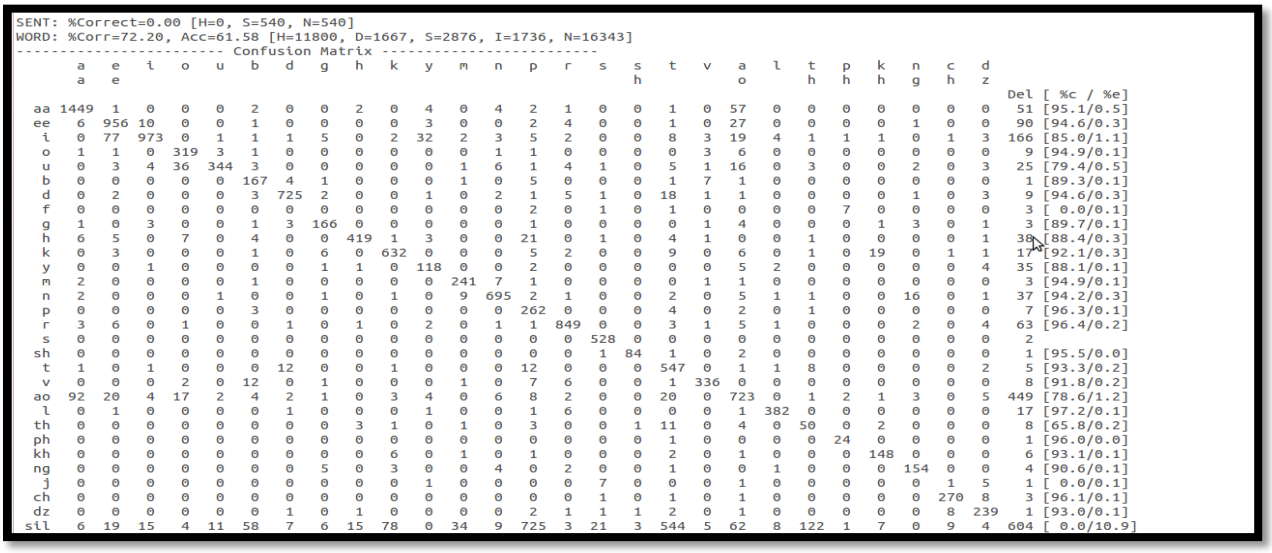
In Figure 4.3, sentence level accuracy is 0.00% and percentage of correctly recognized phonemes is 36.72% and accuracy of PE is 22.39%. Total number of phonemes substituted (*S*) are 1258, total number of phonemes deleted (*D*) are 862, number of phones inserted (*I*) are 480, total number of phonemes correctly recognized (*H*) are 1230 and total number of phonemes (*N*) are 3350.

Table 4.2 consists of the testing accuracy for the three modes of speech (read speech mode, lecture mode and conversation mode). This also contains the metadata on training and testing patterns.

Table 4.2: Testing Accuracy of Each Mode

Mode of Speech	No. of Speakers	Training Data (in Minutes)	Testing Data (in Minutes)	Testing Accuracy
Read Speech Mode	4	138.2	47.73	61.48%
Lecture Mode	1	15.3	4.75	46.96%
Conversation Mode	10	15.1	5.12	22.39%

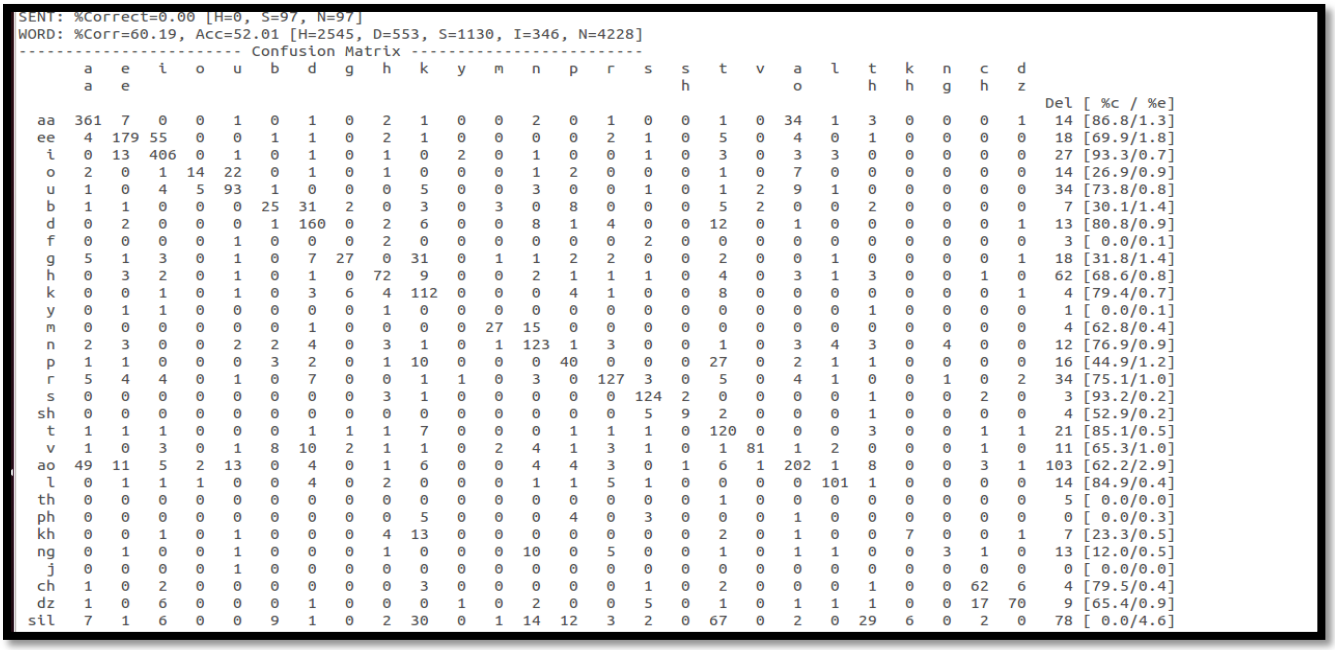
Figure 4.4 and Figure 4.5 show the confusion matrix generated after testing the PE for each gender in Read Speech Mode.



```
SENT: %Correct=0.00 [H=0, S=540, N=540]
WORD: %Corr=72.20, Acc=61.58 [H=11800, D=1667, S=2876, I=1736, N=16343]
---------------------- Confusion Matrix ------------------------
       a   e   i   o   u   b   d   g   h   k   y   m   n   p   r   s   s   t   v   a   l   t   p   k   n   c   d
       a   e                                                   h           o       h   h   h   g   h   z
                                                                                                          Del [ %c / %e]
  aa 1449   1   0   0   0   2   0   0   2   0   4   0   4   2   1   0   0   1   0  57   0   0   0   0   0   0   0   51 [95.1/0.5]
  ee    6 956  10   0   0   1   0   0   0   0   3   0   0   2   4   0   0   1   0  27   0   0   0   0   1   0   0   90 [94.6/0.3]
   i    0  77 973   0   1   1   1   5   0   2  32   2   3   5   2   0   0   8   3  19   4   1   1   1   0   1   3  166 [85.0/1.1]
   o    1   1   0 319   3   1   0   0   0   0   0   0   1   1   0   0   0   0   3   6   0   0   0   0   0   0   0   9 [94.9/0.1]
   u    0   3   4  36 344   3   0   0   0   0   0   1   6   1   4   1   0   5   1  16   0   3   0   0   2   0   3   25 [79.4/0.5]
   b    0   0   0   0   0 167   4   1   0   0   0   0   1   0   5   0   0   0   1   7   1   0   0   0   0   0   0   1 [89.3/0.1]
   d    0   2   0   0   0   3 725   2   0   0   1   0   2   1   5   1   0  18   1   1   0   0   0   0   1   0   3   9 [94.6/0.3]
   f    0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   1   0   1   0   0   0   0   7   0   0   0   0   3 [ 0.0/0.1]
   g    1   0   3   0   0   1   3 166   0   0   0   0   0   1   0   0   0   0   1   4   0   0   0   0   1   3   0   1   3 [89.7/0.1]
   h    6   5   0   7   0   4   0   0 419   1   3   0   0  21   0   1   0   4   1   0   0   1   0   0   0   0   1   38 [88.4/0.3]
   k    0   3   0   0   0   1   0   6   0 632   0   0   0   5   2   0   0   9   0   6   0   1   0  19   0   1   1   17 [92.1/0.3]
   y    0   0   1   0   0   0   0   1   1   0 118   0   0   2   0   0   0   0   0   5   2   0   0   0   0   0   4   35 [88.1/0.1]
   m    2   0   0   0   0   1   0   0   0   0   0 241   7   1   0   0   0   0   1   1   0   0   0   0   0   0   0   3 [94.9/0.1]
   n    2   0   0   0   1   0   0   1   0   1   0   9 695   2   1   0   0   2   0   5   1   1   0   0  16   0   1   37 [94.2/0.3]
   p    0   0   0   0   0   3   0   0   0   0   0   0   0 262   0   0   0   4   0   2   0   1   0   0   0   0   0   7 [96.3/0.1]
   r    3   6   0   1   0   0   1   0   1   0   2   0   1   1 849   0   0   3   1   5   1   0   0   0   2   0   4   63 [96.4/0.2]
   s    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 528   0   0   0   0   0   0   0   0   0   0   0   2
  sh    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1  84   1   0   2   0   0   0   0   0   0   0   1 [95.5/0.0]
   t    1   0   1   0   0   0  12   0   0   1   0   0   0  12   0   0   0 547   0   1   1   8   0   0   0   0   2   5 [93.3/0.2]
   v    0   0   0   2   0  12   0   1   0   0   0   1   0   7   6   0   0   1 336   0   0   0   0   0   0   0   0   8 [91.8/0.2]
  ao   92  20   4  17   2   4   2   1   0   3   4   0   6   8   2   0   0  20   0 723   0   1   2   1   3   0   5 449 [78.6/1.2]
   l    0   1   0   0   0   0   1   0   0   0   1   0   0   1   6   0   0   0   0   1 382   0   0   0   0   0   0   17 [97.2/0.1]
  th    0   0   0   0   0   0   0   0   3   1   0   1   0   3   0   0   1  11   0   4   0  50   0   2   0   0   0   8 [65.8/0.2]
  ph    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0  24   0   0   0   0   1 [96.0/0.0]
  kh    0   0   0   0   0   0   0   0   0   6   0   1   0   1   0   0   0   2   0   1   0   0   0 148   0   0   0   6 [93.1/0.1]
  ng    0   0   0   0   0   0   0   0   5   0   3   0   0   4   0   2   0   0   1   0   0   1   0   0   0 154   0   0   4 [90.6/0.1]
   j    0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   7   0   0   0   1   0   0   0   0   0   1   5   1 [ 0.0/0.1]
  ch    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   1   0   1   0   0   0   0   0 270   8   3 [96.1/0.1]
  dz    0   0   0   0   0   0   0   1   0   0   0   0   0   2   1   1   1   2   0   1   0   0   0   0   0   8 239   1 [93.0/0.1]
 sil    6  19  15   4  11  58   7   6  15  78   0  34   9 725   3  21   3 544   5  62   8 122   1   7   0   9   4 604 [ 0.0/10.9]
```

Figure 4.4: Confusion Matrix- Read Speech Mode_Males

In Figure 4.4, sentence level accuracy is 0.00% and percentage of correctly recognized phonemes is 72.20% and accuracy of PE is 61.58%. Total number of phonemes substituted (*S*) are 2876, total number of phonemes deleted (*D*) are 1667, number of phones inserted (*I*) are 1736, total number of phonemes correctly recognized (*H*) are 11800 and total number of phonemes (*N*) are 16343.



```
SENT: %Correct=0.00 [H=0, S=97, N=97]
WORD: %Corr=60.19, Acc=52.01 [H=2545, D=553, S=1130, I=346, N=4228]
------------------------ Confusion Matrix -------------------------
       a   e   i   o   u   b   d   g   h   k   y   m   n   p   r   s   s   t   v   a   l   t   k   n   c   d
       a   e                                                   h           o       h   h   g   h   z
                                                                                                      Del [ %c / %e]
  aa  361   7   0   0   1   0   1   0   2   1   0   0   2   0   1   0   0   1   0  34   1   3   0   0   0   1   14 [86.8/1.3]
  ee    4 179  55   0   0   1   1   0   2   1   0   0   0   0   2   1   0   5   0   4   0   1   0   0   0   0   18 [69.9/1.8]
   i    0  13 406   0   1   0   1   1   0   2   0   0   1   0   0   1   0   3   0   3   3   0   0   0   0   0   27 [93.3/0.7]
   o    2   0   1  14  22   0   1   0   1   0   0   0   1   2   0   0   0   1   0   7   0   0   0   0   0   0   14 [26.9/0.9]
   u    1   0   4   5  93   1   0   0   0   5   0   0   3   0   0   1   0   1   2   9   1   0   0   0   0   0   34 [73.8/0.8]
   b    1   1   0   0   0  25  31   2   0   3   0   3   0   8   0   0   0   5   2   0   0   2   0   0   0   0    7 [30.1/1.4]
   d    0   2   0   0   0   1 160   0   2   6   0   0   8   1   4   0   0  12   0   1   0   0   0   0   0   1   13 [80.8/0.9]
   f    0   0   0   0   1   0   0   0   2   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0    3 [ 0.0/0.1]
   g    5   1   3   0   1   0   7  27   0  31   0   1   1   2   2   0   0   2   0   0   1   0   0   0   0   1   18 [31.8/1.4]
   h    0   3   2   0   1   0   1   0  72   9   0   0   2   1   1   1   0   4   0   3   1   3   0   0   1   0   62 [68.6/0.8]
   k    0   0   1   0   1   0   3   6   4 112   0   0   0   4   1   0   0   8   0   0   0   0   0   0   0   1    4 [79.4/0.7]
   y    0   1   1   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0    1 [ 0.0/0.1]
   m    0   0   0   0   0   0   1   0   0   0   0  27  15   0   0   0   0   0   0   0   0   0   0   0   0   0    4 [62.8/0.4]
   n    2   3   0   0   2   2   4   0   3   1   0   1 123   1   3   0   0   1   0   3   4   3   0   4   0   0   12 [76.9/0.9]
   p    1   1   0   0   0   3   2   0   1  10   0   0   0  40   0   0   0  27   0   2   1   1   0   0   0   0   16 [44.9/1.2]
   r    5   4   4   0   1   0   7   0   0   1   1   0   3   0 127   3   0   5   0   4   1   0   0   1   0   2   34 [75.1/1.0]
   s    0   0   0   0   0   0   0   0   3   1   0   0   0   0   0 124   2   0   0   0   0   1   0   0   2   0    3 [93.2/0.2]
  sh    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   5   9   2   0   0   0   1   0   0   0   0    4 [52.9/0.2]
   t    1   1   1   0   0   0   1   1   1   7   0   0   0   1   1   1   0 120   0   0   0   3   0   0   1   1   21 [85.1/0.5]
   v    1   0   3   0   1   8  10   2   1   1   0   2   4   1   3   1   0   1  81   1   2   0   0   0   1   0   11 [65.3/1.0]
  ao   49  11   5   2  13   0   4   0   1   6   0   0   4   4   3   0   1   6   1 202   1   8   0   0   3   1  103 [62.2/2.9]
   l    0   1   1   1   0   0   4   0   2   0   0   0   1   1   5   1   0   0   0   0 101   1   0   0   0   0   14 [84.9/0.4]
  th    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0    5 [ 0.0/0.0]
  ph    0   0   0   0   0   0   0   0   0   5   0   0   0   0   4   0   3   0   0   0   1   0   0   0   0   0    0 [ 0.0/0.3]
  kh    0   0   1   0   1   0   0   0   4  13   0   0   0   0   0   0   0   2   0   1   0   0   7   0   0   1    7 [23.3/0.5]
  ng    0   1   0   0   1   0   0   0   1   0   0   0  10   0   5   0   0   1   0   1   1   0   0   3   1   0   13 [12.0/0.5]
   j    0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0 [ 0.0/0.0]
  ch    1   0   2   0   0   0   0   0   0   3   0   0   0   0   0   1   0   2   0   0   0   1   0   0  62   6    4 [79.5/0.4]
  dz    1   0   6   0   0   0   1   0   0   0   1   0   2   0   0   5   0   1   0   1   1   1   0   0  17  70    9 [65.4/0.9]
 sil    7   1   6   0   0   9   1   0   2  30   0   1  14  12   3   2   0  67   0   2   0  29   6   0   2   0   78 [ 0.0/4.6]
```

Figure 4.5: Confusion Matrix- Read Speech Mode_Females

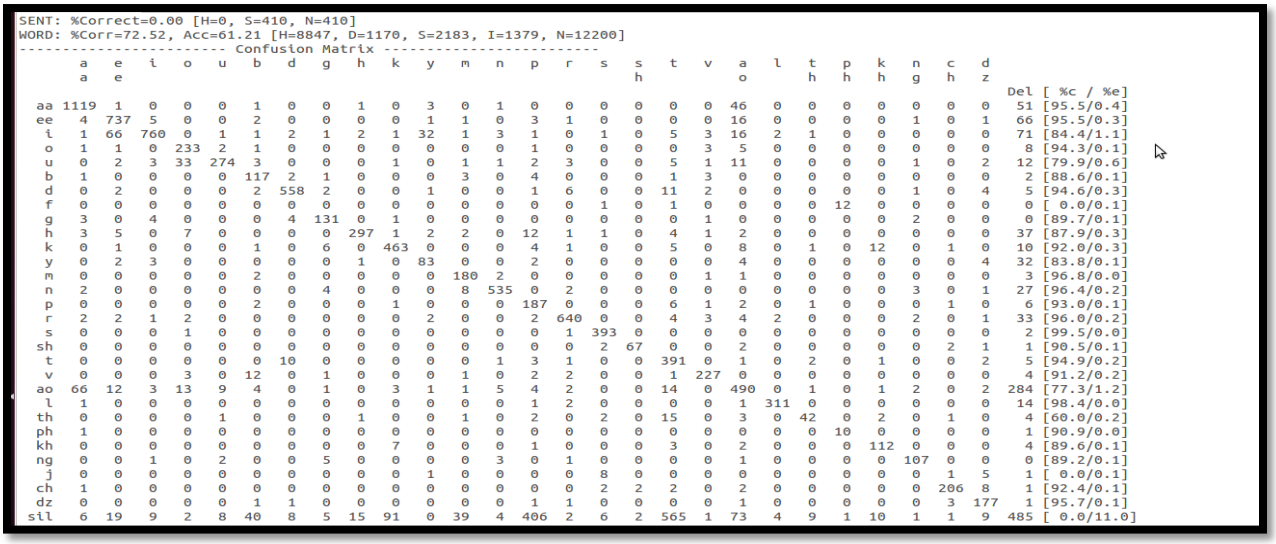
In Figure 4.5, sentence level accuracy is 0.00% and percentage of correctly recognized phonemes is 60.19% and accuracy of PE is 52.01%. Total number of phonemes substituted (*S*) are 1130, total number of phonemes deleted (*D*) are 553, number of phones inserted (*I*) are 346, total number of phonemes correctly recognized (*H*) are 2545 and total number of phonemes (*N*) are 4228.

Table 4.3 consists of testing accuracy of each gender separately in read speech mode. This also contains the metadata on training and testing patterns.

Table 4.3: Testing Accuracy of Read Speech Mode

Gender	No. of Speakers	Training Data (in Minutes)	Testing Data (in Minutes)	Testing Accuracy
Male	2	122.41	38.65	61.58%
Female	2	18.48	6.35	52.01%

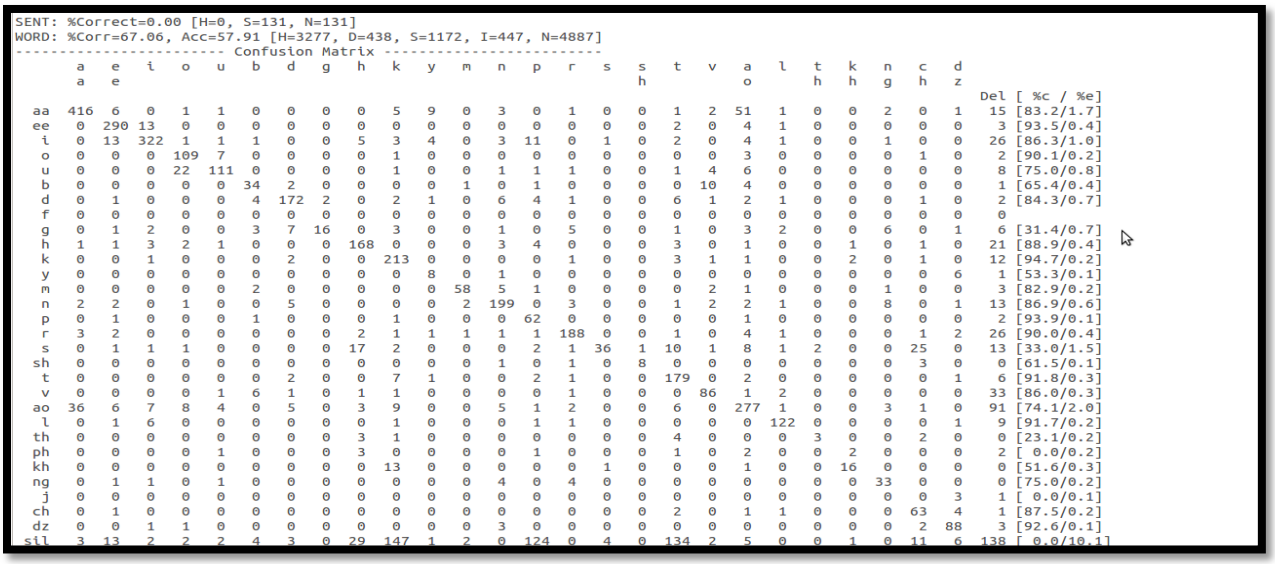
Figure 4.6 and Figure 4.7 show the confusion matrix generated after testing the PE for each male speaker in Read Speech Mode.



```
SENT: %Correct=0.00 [H=0, S=410, N=410]
WORD: %Corr=72.52, Acc=61.21 [H=8847, D=1170, S=2183, I=1379, N=12200]
------------------------ Confusion Matrix -------------------------
       a   e   i   o   u   b   d   g   h   k   y   m   n   p   r   s   s   t   v   a   l   t   p   k   n   c   d
       a   e                                                   h               o       h   h   h   g   h   z
                                                                                                          Del [ %c / %e]
  aa 1119   1   0   0   0   1   0   0   1   0   3   0   1   0   0   0   0   0   0  46   0   0   0   0   0   0   0   51 [95.5/0.4]
  ee    4 737   5   0   0   2   0   0   0   0   1   1   0   3   1   0   0   0   0  16   0   0   0   0   1   0   1   66 [95.5/0.3]
   i    1  66 760   0   1   1   2   1   2   1  32   1   3   1   0   1   0   5   3  16   2   1   0   0   0   0   0   71 [84.4/1.1]
   o    1   1   0 233   2   1   0   0   0   0   0   0   0   1   0   0   0   0   3   5   0   0   0   0   0   0   0    8 [94.3/0.1]
   u    0   2   3  33 274   3   0   0   0   1   0   1   1   2   3   0   0   5   1  11   0   0   0   0   1   0   2   12 [79.9/0.6]
   b    1   0   0   0   0 117   2   1   0   0   0   3   0   4   0   0   0   1   3   0   0   0   0   0   0   0   0    2 [88.6/0.1]
   d    0   2   0   0   0   2 558   2   0   0   1   0   0   1   6   0   0  11   2   0   0   0   0   0   1   0   4    5 [94.6/0.3]
   f    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   1   0   0   0   0  12   0   0   0   0    0 [ 0.0/0.1]
   g    3   0   4   0   0   0   4 131   0   1   0   0   0   0   0   0   0   0   1   0   0   0   0   0   2   0   0    0 [89.7/0.1]
   h    3   5   0   7   0   0   0   0 297   1   2   2   0  12   1   1   0   4   1   2   0   0   0   0   0   0   0   37 [87.9/0.3]
   k    0   1   0   0   0   1   0   6   0 463   0   0   0   4   1   0   0   5   0   8   0   1   0  12   0   1   0   10 [92.0/0.3]
   y    0   2   3   0   0   0   0   0   1   0  83   0   0   2   0   0   0   0   0   4   0   0   0   0   0   0   4   32 [83.8/0.1]
   m    0   0   0   0   0   2   0   0   0   0   0 180   2   0   0   0   0   0   1   1   0   0   0   0   0   0   0    3 [96.8/0.0]
   n    2   0   0   0   0   0   0   4   0   0   0   8 535   0   2   0   0   0   0   0   0   0   0   0   3   0   1   27 [96.4/0.2]
   p    0   0   0   0   0   2   0   0   0   1   0   0   0 187   0   0   0   6   1   2   0   1   0   0   0   1   0    6 [93.0/0.1]
   r    2   2   1   2   0   0   0   0   0   0   2   0   0   2 640   0   0   4   3   4   2   0   0   0   0   2   0   33 [96.0/0.2]
   s    0   0   0   1   0   0   0   0   0   0   0   0   0   0   1 393   0   0   0   0   0   0   0   0   0   0   0    2 [99.5/0.0]
  sh    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2  67   0   0   2   0   0   0   0   0   2   1    1 [90.5/0.1]
   t    0   0   0   0   0   0  10   0   0   0   0   0   1   3   1   0   0 391   0   1   0   2   0   1   0   0   2    5 [94.9/0.2]
   v    0   0   0   3   0  12   0   1   0   0   0   1   0   2   2   0   0   1 227   0   0   0   0   0   0   0   0    4 [91.2/0.2]
  ao   66  12   3  13   9   4   0   1   0   3   1   1   5   4   2   0   0  14   0 490   0   1   0   1   2   0   2  284 [77.3/1.2]
   l    1   0   0   0   0   0   0   0   0   0   0   0   0   1   2   0   0   0   0   1 311   0   0   0   0   0   0   14 [98.4/0.0]
  th    0   0   0   0   1   0   0   0   1   0   0   1   0   2   0   2   0  15   0   3   0  42   0   2   0   1   0    4 [60.0/0.2]
  ph    1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0  10   0   0   0   0    1 [90.9/0.0]
  kh    0   0   0   0   0   0   0   0   0   7   0   0   0   1   0   0   0   3   0   2   0   0   0 112   0   0   0    4 [89.6/0.1]
  ng    0   0   1   0   2   0   0   5   0   0   0   0   3   0   1   0   0   0   0   1   0   0   0   0 107   0   0    0 [89.2/0.1]
   j    0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   8   0   0   0   0   0   0   0   0   1   5    1 [ 0.0/0.1]
  ch    1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   2   2   0   2   0   0   0   0   0 206    8 [92.4/0.1]
  dz    0   0   0   0   0   1   1   0   0   0   0   0   0   1   1   0   0   0   0   1   0   0   0   0   0   3 177    1 [95.7/0.1]
 sil    6  19   9   2   8  40   8   5  15  91   0  39   4 406   2   6   2 565   1  73   4   9   1  10   1   1   9  485 [ 0.0/11.0]
```

Figure 4.6: Confusion Matrix- Read Speech Mode_Male1

In Figure 4.6, sentence level accuracy is 0.00% and percentage of correctly recognized phonemes is 72.52% and accuracy of PE is 61.21%. Total number of phonemes substituted (*S*) are 2183, total number of phonemes deleted (*D*) are 1170, number of phones inserted (*I*) are 1379, total number of phonemes correctly recognized (*H*) are 8847 and total number of phonemes (*N*) are 12200.



```
SENT: %Correct=0.00 [H=0, S=131, N=131]
WORD: %Corr=67.06, Acc=57.91 [H=3277, D=438, S=1172, I=447, N=4887]
------------------------ Confusion Matrix -------------------------
       a   e   i   o   u   b   d   g   h   k   y   m   n   p   r   s   s   t   v   a   l   t   k   n   c   d
       a   e                                                   h               o       h   h   g   h   z
                                                                                                      Del [ %c / %e]
  aa  416   6   0   1   1   0   0   0   0   5   9   0   3   0   1   0   0   1   2  51   1   0   0   2   0   1   15 [83.2/1.7]
  ee    0 290  13   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   4   1   0   0   0   0   0    3 [93.5/0.4]
   i    0  13 322   1   1   1   0   0   5   3   4   0   3  11   0   1   0   2   0   4   1   0   0   1   0   0   26 [86.3/1.0]
   o    0   0   0 109   7   0   0   0   0   1   0   0   0   0   0   0   0   0   0   3   0   0   0   0   1   0    2 [90.1/0.2]
   u    0   0   0  22 111   0   0   0   0   1   0   0   1   1   1   0   0   1   4   6   0   0   0   0   0   0    8 [75.0/0.8]
   b    0   0   0   0   0  34   2   0   0   0   0   1   0   1   0   0   0   0  10   4   0   0   0   0   0   0    1 [65.4/0.4]
   d    0   1   0   0   0   4 172   2   0   2   1   0   6   4   1   0   0   6   1   2   1   0   0   0   1   0    2 [84.3/0.7]
   f    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0
   g    0   1   2   0   0   3   7  16   0   3   0   0   1   0   5   0   0   1   0   3   2   0   0   6   0   1    6 [31.4/0.7]
   h    1   1   3   2   1   0   0   0 168   0   0   0   3   4   0   0   0   3   0   1   0   0   1   0   1   0   21 [88.9/0.4]
   k    0   0   1   0   0   0   2   0   0 213   0   0   0   0   1   0   0   3   1   1   0   0   2   0   1   0   12 [94.7/0.2]
   y    0   0   0   0   0   0   0   0   0   0   8   0   1   0   0   0   0   0   0   0   0   0   0   0   0   6    1 [53.3/0.1]
   m    0   0   0   0   0   2   0   0   0   0   0  58   5   1   0   0   0   0   2   1   0   0   0   1   0   0    3 [82.9/0.2]
   n    2   2   0   1   0   0   5   0   0   0   0   2 199   0   3   0   0   1   2   2   1   0   0   8   0   1   13 [86.9/0.6]
   p    0   1   0   0   0   1   0   0   0   1   0   0   0  62   0   0   0   0   0   1   0   0   0   0   0   0    2 [93.9/0.1]
   r    3   2   0   0   0   0   0   0   2   1   1   1   1   1 188   0   0   1   0   4   1   0   0   0   1   2   26 [90.0/0.4]
   s    0   1   1   1   0   0   0   0  17   2   0   0   0   2   1  36   1  10   1   8   1   2   0   0  25   0   13 [33.0/1.5]
  sh    0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   1   0   8   0   0   0   0   0   0   3   0    0 [61.5/0.1]
   t    0   0   0   0   0   0   2   0   0   7   1   0   0   2   1   0   0 179   0   2   0   0   0   0   0   1    6 [91.8/0.3]
   v    0   0   0   0   1   6   1   0   1   1   0   0   0   0   1   0   0   0  86   1   2   0   0   0   0   0   33 [86.0/0.3]
  ao   36   6   7   8   4   0   5   0   3   9   0   0   5   1   2   0   0   6   0 277   1   0   0   3   1   0   91 [74.1/2.0]
   l    0   1   6   0   0   0   0   0   0   1   0   0   0   1   1   0   0   0   0   0 122   0   0   0   0   1    9 [91.7/0.2]
  th    0   0   0   0   0   0   0   0   3   1   0   0   0   0   0   0   0   4   0   0   0   3   0   0   2   0    0 [23.1/0.2]
  ph    0   0   0   0   1   0   0   0   3   0   0   0   0   1   0   0   0   1   0   2   0   0   2   0   0   0    2 [ 0.0/0.2]
  kh    0   0   0   0   0   0   0   0   0  13   0   0   0   0   0   1   0   0   0   1   0   0  16   0   0   0    0 [51.6/0.3]
  ng    0   1   1   0   1   0   0   0   0   0   0   0   4   0   4   0   0   0   0   0   0   0   0  33   0   0    0 [75.0/0.2]
   j    0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3   0    1 [ 0.0/0.1]
  ch    0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   1   1   0   0   0  63   4    1 [87.5/0.2]
  dz    0   0   1   1   0   0   0   0   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   2  88   3    3 [92.6/0.1]
 sil    3  13   2   2   2   4   3   0  29 147   1   2   0 124   0   4   0 134   2   5   0   0   1   0  11   6  138 [ 0.0/10.1]
```

Figure 4.7: Confusion Matrix- Read Speech Mode_Male2

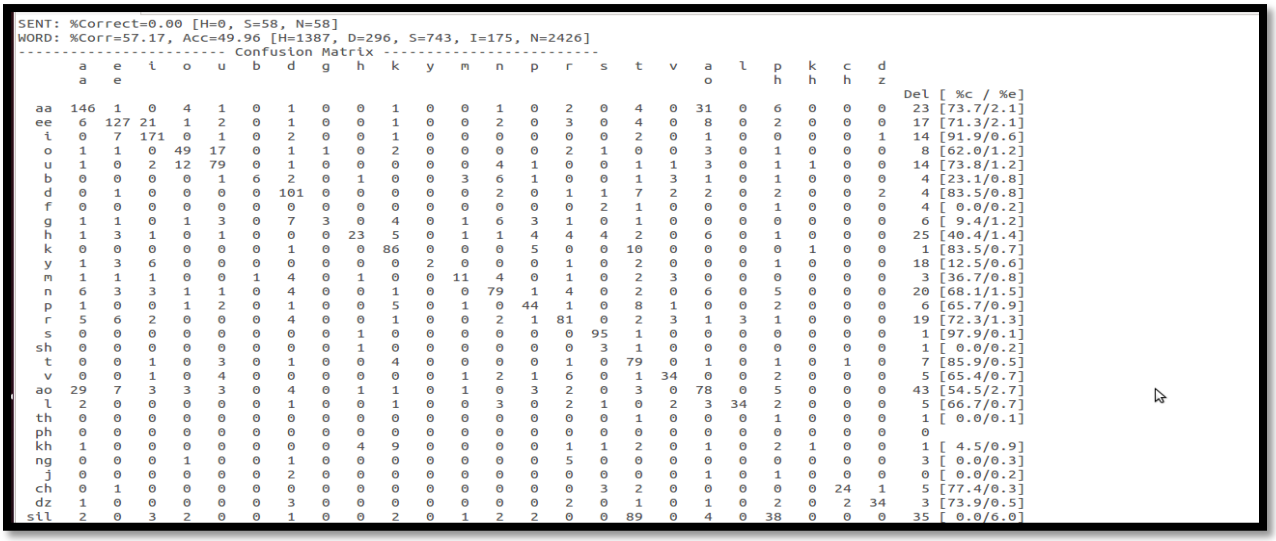
In Figure 4.7, sentence level accuracy is 0.00% and percentage of correctly recognized phonemes is 67.06% and accuracy of PE is 57.91%. Total number of phonemes substituted (*S*) are 1172, total number of phonemes deleted (*D*) are 438, number of phones inserted (*I*) are 447, total number of phonemes correctly recognized (*H*) are 3277 and total number of phonemes (*N*) are 4887.

Table 4.4 consists of testing accuracy of each male speaker in read speech mode. This also contains the metadata on training and testing patterns.

Table 4.4: Testing Accuracy of Read Speech Mode: Males

Male Id	Training Data (in Minutes)	Testing Data (in minutes)	Testing Accuracy
M0001	92.33	29.25	61.21%
M0002	29.5	9.98	57.91%

Figure 4.8 and Figure 4.9 shows the confusion matrix generated after testing the PE for each male speaker in Read Speech Mode.

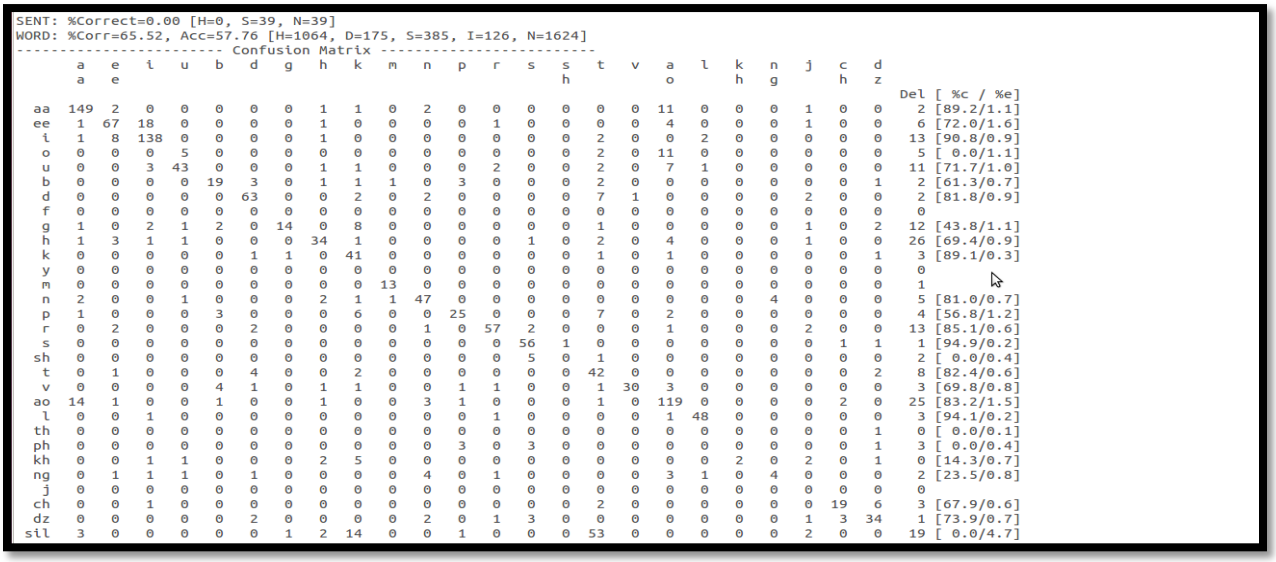


```
SENT: %Correct=0.00 [H=0, S=58, N=58]
WORD: %Corr=57.17, Acc=49.96 [H=1387, D=296, S=743, I=175, N=2426]
------------------------ Confusion Matrix -------------------------
       a   e   i   o   u   b   d   g   h   k   y   m   n   p   r   s   t   v   a   l   p   k   c   d
       a   e                                                       o       h   h   h   z
                                                                                           Del [ %c / %e]
  aa 146   1   0   4   1   0   1   0   0   1   0   0   1   0   2   0   4   0  31   0   6   0   0   0   23 [73.7/2.1]
  ee   6 127  21   1   2   0   1   0   0   1   0   0   2   0   3   0   4   0   8   0   2   0   0   0   17 [71.3/2.1]
   i   0   7 171   0   1   0   2   0   0   1   0   0   0   0   0   0   2   0   1   0   0   0   0   1   14 [91.9/0.6]
   o   1   1   0  49  17   0   1   1   0   2   0   0   0   0   2   1   0   0   3   0   1   0   0   0    8 [62.0/1.2]
   u   1   0   2  12  79   0   1   0   0   0   0   0   4   1   0   0   1   1   3   0   1   1   0   0   14 [73.8/1.2]
   b   0   0   0   0   1   6   2   0   1   0   0   3   6   1   0   0   1   3   1   0   1   0   0   0    4 [23.1/0.8]
   d   0   1   0   0   0   0 101   0   0   0   0   0   2   0   1   1   7   2   2   0   2   0   0   2    4 [83.5/0.8]
   f   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2   1   0   0   0   1   0   0   0    4 [ 0.0/0.2]
   g   1   1   0   1   3   0   7   3   0   4   0   1   6   3   1   0   1   0   0   0   0   0   0   0    6 [ 9.4/1.2]
   h   1   3   1   0   1   0   0   0  23   5   0   1   1   4   4   4   2   0   6   0   1   0   0   0   25 [40.4/1.4]
   k   0   0   0   0   0   0   1   0   0  86   0   0   0   5   0   0  10   0   0   0   0   1   0   0    1 [83.5/0.7]
   y   1   3   6   0   0   0   0   0   0   0   2   0   0   0   1   0   2   0   0   0   1   0   0   0   18 [12.5/0.6]
   m   1   1   1   0   0   1   4   0   1   0   0  11   4   0   1   0   2   3   0   0   0   0   0   0    3 [36.7/0.8]
   n   6   3   3   1   1   0   4   0   0   1   0   0  79   1   4   0   2   0   6   0   5   0   0   0   20 [68.1/1.5]
   p   1   0   0   1   2   0   1   0   0   5   0   1   0  44   1   0   8   1   0   0   2   0   0   0    6 [65.7/0.9]
   r   5   6   2   0   0   0   4   0   0   1   0   0   2   1  81   0   2   3   1   3   1   0   0   0   19 [72.3/1.3]
   s   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0  95   1   0   0   0   0   0   0   0    1 [97.9/0.1]
  sh   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   3   1   0   0   0   0   0   0   0    1 [ 0.0/0.2]
   t   0   0   1   0   3   0   1   0   0   4   0   0   0   0   1   0  79   0   1   0   1   0   1   0    7 [85.9/0.5]
   v   0   0   1   0   4   0   0   0   0   0   0   1   2   1   6   0   1  34   0   0   2   0   0   0    5 [65.4/0.7]
  ao  29   7   3   3   3   0   4   0   1   1   0   1   0   3   2   0   3   0  78   0   5   0   0   0   43 [54.5/2.7]
   l   2   0   0   0   0   0   1   0   0   1   0   0   3   0   2   1   0   2   3  34   2   0   0   0    5 [66.7/0.7]
  th   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0   0   0   1   0   0   0    1 [ 0.0/0.1]
  ph   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    0
  kh   1   0   0   0   0   0   0   0   0   4   9   0   0   0   0   0   1   1   2   0   1   0   2   1   0   0    1 [ 4.5/0.9]
  ng   0   0   0   1   0   0   1   0   0   0   0   0   0   0   0   5   0   0   0   0   0   0   0   0    3 [ 0.0/0.3]
   j   0   0   0   0   0   0   0   2   0   0   0   0   0   0   0   0   0   0   0   0   1   0   1   0   0   0    0 [ 0.0/0.2]
  ch   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   3   2   0   0   0   0   0  24   1    5 [77.4/0.3]
  dz   1   0   0   0   0   0   3   0   0   0   0   0   0   0   2   0   1   0   1   0   2   0   2  34    3 [73.9/0.5]
 sil   2   0   3   2   0   0   1   0   0   2   0   1   2   2   0   0  89   0   4   0  38   0   0   0   35 [ 0.0/6.0]
```

Figure 4.8: Confusion Matrix- Read Speech Mode_Female1

In Figure 4.8, sentence level accuracy is 0.00% and percentage of correctly recognized phonemes is 57.17% and accuracy of PE is 49.96%. Total number of phonemes substituted (*S*) are 743, total number of phonemes deleted (*D*) are 296, number of phones inserted (*I*) are 175, total

number of phonemes correctly recognized (*H*) are 1387 and total number of phonemes (*N*) are 2426.



```
SENT: %Correct=0.00 [H=0, S=39, N=39]
WORD: %Corr=65.52, Acc=57.76 [H=1064, D=175, S=385, I=126, N=1624]
------------------------ Confusion Matrix -------------------------
       a   e   i   u   b   d   g   h   k   m   n   p   r   s   s   t   v   a   l   k   n   j   c   d
       a   e                                                   h             o       h   g       h   z
                                                                                                     Del [ %c / %e]
  aa 149   2   0   0   0   0   0   1   1   0   2   0   0   0   0   0   0  11   0   0   0   1   0   0   2 [89.2/1.1]
  ee   1  67  18   0   0   0   0   1   0   0   0   0   1   0   0   0   0   4   0   0   0   1   0   0   6 [72.0/1.6]
   i   1   8 138   0   0   0   0   1   0   0   0   0   0   0   0   0   2   0   0   2   0   0   0   0   0  13 [90.8/0.9]
   o   0   0   0   5   0   0   0   0   0   0   0   0   0   0   0   0   2   0  11   0   0   0   0   0   0   5 [ 0.0/1.1]
   u   0   0   3  43   0   0   0   1   1   0   0   0   2   0   0   2   0   0   7   1   0   0   0   0   0  11 [71.7/1.0]
   b   0   0   0   0  19   3   0   1   1   1   0   3   0   0   0   2   0   0   0   0   0   0   0   0   1   2 [61.3/0.7]
   d   0   0   0   0   0  63   0   0   2   0   2   0   0   0   0   7   1   0   0   0   0   2   0   0   2 [81.8/0.9]
   f   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   g   1   0   2   1   2   0  14   0   8   0   0   0   0   0   0   1   0   0   0   0   0   1   0   2  12 [43.8/1.1]
   h   1   3   1   1   0   0   0  34   1   0   0   0   0   1   0   2   0   4   0   0   0   1   0   0  26 [69.4/0.9]
   k   0   0   0   0   0   1   1   0  41   0   0   0   0   0   0   1   0   1   0   0   0   0   0   1   3 [89.1/0.3]
   y   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
   m   0   0   0   0   0   0   0   0   0  13   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
   n   2   0   0   1   0   0   0   2   1   1  47   0   0   0   0   0   0   0   0   0   0   4   0   0   0   5 [81.0/0.7]
   p   1   0   0   0   3   0   0   0   6   0   0  25   0   0   0   7   0   2   0   0   0   0   0   0   0   4 [56.8/1.2]
   r   0   2   0   0   0   2   0   0   0   0   1   0  57   2   0   0   0   1   0   0   0   2   0   0   0  13 [85.1/0.6]
   s   0   0   0   0   0   0   0   0   0   0   0   0   0  56   1   0   0   0   0   0   0   0   1   1   1 [94.9/0.2]
  sh   0   0   0   0   0   0   0   0   0   0   0   0   0   5   0   1   0   0   0   0   0   0   0   0   0   2 [ 0.0/0.4]
   t   0   1   0   0   0   4   0   0   2   0   0   0   0   0   0  42   0   0   0   0   0   0   0   0   2   8 [82.4/0.6]
   v   0   0   0   0   4   1   0   1   1   0   0   1   1   0   0   1  30   3   0   0   0   0   0   0   0   3 [69.8/0.8]
  ao  14   1   0   0   1   0   0   1   0   0   3   1   0   0   0   1   0 119   0   0   0   0   2   0  25 [83.2/1.5]
   l   0   0   1   0   0   0   0   0   0   0   0   0   1   0   0   0   0   1  48   0   0   0   0   0   0   3 [94.1/0.2]
  th   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   0 [ 0.0/0.1]
  ph   0   0   0   0   0   0   0   0   0   0   0   3   0   3   0   0   0   0   0   0   0   0   0   1   3 [ 0.0/0.4]
  kh   0   0   1   1   0   0   0   2   5   0   0   0   0   0   0   0   0   0   0   2   0   2   0   1   0 [14.3/0.7]
  ng   0   1   1   1   0   1   0   0   0   0   4   0   1   0   0   0   0   3   1   0   4   0   0   0   2 [23.5/0.8]
   j   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  ch   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0   0   0   0   0  19   6   3 [67.9/0.6]
  dz   0   0   0   0   0   2   0   0   0   0   2   0   1   3   0   0   0   0   0   0   0   1   3  34   1 [73.9/0.7]
 sil   3   0   0   0   0   0   1   2  14   0   0   1   0   0   0  53   0   0   0   0   0   2   0   0  19 [ 0.0/4.7]
```

Figure 4.9: Confusion Matrix- Read Speech Mode_Female2

In Figure 4.9, sentence level accuracy is 0.00% and percentage of correctly recognized phonemes is 65.52% and accuracy of PE is 57.76%. Total number of phonemes substituted (*S*) are 385, total number of phonemes deleted (*D*) are 175, number of phones inserted (*I*) are 126, total number of phonemes correctly recognized (*H*) are 1064 and total number of phonemes (*N*) are 1624.

Table 4.5 consists of testing accuracy of each female speaker in read speech mode. This also contains the metadata on training and testing patterns.

Table 4.5: Testing Accuracy of Read Speech Mode: Females

Female Id	Training Data (in Minutes)	Testing Data (in Minutes)	Testing Accuracy
F0001	10.35	3.45	49.96%
F0002	7.47	2.4	57.76%

CHAPTER 5

Conclusion and Future Scope

5.1 Conclusion

In this work, data has been collected in three modes: read speech mode, lecture mode and conversation mode. In each mode, recording is done using a sampling frequency of 48 KHZ and a bit rate of 16 bits per second. In read speech mode, data has been collected from four native Punjabi speakers for a duration of about 185.93 minutes, in lecture mode, collected from a radio channel, Punjabi radio USA of about 20.05 minutes and in conversation mode, the conversation and discussion of Punjabi speakers are recorded of about 20.22 minutes.

Collected data is transcribed using IPA chart such that all the basic sound units present in the spoken utterances are represented in the symbolic form.

After preparing all the data, speaker independent phonetic engine has been developed which provides the phoneme level recognition for continuous speech signal of Punjabi language. For this, HTK toolkit is used and the platform is Ubuntu-12.04 32-bit system. HTK toolkit is a statistical tool for building HMMs. To provide training to PE, a set of 30 unique phonemes including silence and continuous density HMMs have been used. PE is trained separately for the following cases: read speech mode, lecture mode, conversation mode, for each gender in read speech mode and for each Punjabi speaker in read speech mode. In each case, PE is trained with 75% of data and its performance was evaluated with 25% remaining data.

PE got an accuracy of 61.48% in read speech mode, 46.96% accuracy in lecture mode and 22.39% accuracy in conversation mode. In read speech mode, the overall accuracy of PE for male speakers is 61.58% and for female speakers is 52.01%.

5.2 Future Scope

The accuracy of PE can increased by collecting more data such that PE can be trained with a large amount of data and a good accuracy of PE can be obtained and also by increasing the number of phonemes in phoneme list better accuracy of PE can be obtained.

This system can also be developed for other Indian languages using the same procedure as used in developing this system.

This work can be extended to provide word level recognition of continuous speech signal.

REFERENCES

[1] Lee, K. F., Hon, H. W., Hwang, M. Y., and Mahajan, S. (1989), "The SPHINX speech recognition system", *Proceedings of the IEEE International Conference in Acoustics, Speech and Signal Processing*.

[2] Lee, K. F., and Hon, H. W. (1989). "Speaker-independent phone recognition using hidden Markov models", *IEEE Transactions on Acoustics, Speech and Signal Processing, 37*(11), 1641-1648.

[3] Rabiner, L., Wilpon, J. G., and Soong, F. K. (1989). "High performance connected digit recognition using hidden Markov models", *IEEE Transactions on Acoustics, Speech and Signal Processing, 37*(8), 1214-1225.

[4] Lamel, L. F., and Gauvain, J. L. (1992). "Experiments on speaker-independent phone recognition using BREF", *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP,* 1, 557-560.

[5] Ratnayake, N., Savic, M., and Sorensen, J. (1992). "Use of semi-Markov models for speaker-independent phoneme recognition", *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP,* 1, 565-568.

[6] Brugnara, F., Falavigna, D., and Omologo, M. (1993). "Automatic segmentation and labeling of speech based on Hidden Markov Models", *Speech Communication, 12*(4), 357-370.

[7] Kapadia, S., Valtchev, V., and Young, S. J. (1993). "MMI training for continuous phoneme recognition on the TIMIT database", *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2, 491-494.

[8] Angelini, B., Brugnara, F., Falavigna, D., Giuliani, D., Gretter, R., and Omologo, M. (1994). "Speaker independent continuous speech recognition using an acoustic-phonetic Italian corpus", *Proceedings of the International Conference on Spoken Language Processing, ICSLP*, 1391-1394.

[9] Leggetter, C. J., and Woodland, P. C. (1995). "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models", *Computer Speech and Language, 9*(2), 171-185.

[10] Woodland, P. C., Leggetter, C. J., Odell, J. J., Valtchev, V., and Young, S. J. (1995). "The

1994 HTK large vocabulary speech recognition system", IEEE *International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 1, 73-76.

[11] Knill, K. M., Gales, M. J., and Young, S. J. (1996). "Use of Gaussian selection in large vocabulary continuous speech recognition using HMMs", *IEEE Fourth International Conference on Spoken Language, ICSLP*, 1, 470-473.

[12] Mari, J. F., Fohr, D., and Junqua, J. C. (1996). "A second-order HMM for high performance word and phoneme-based continuous speech recognition", *Proceedings of* the *IEEE International Conference on Acoustics, Speech, and Signal Processing,. ICASSP*, 1, 435-438.

[13] Ming, J., and Smith, F. J. (1998). "Improved phone recognition using Bayesian triphone models", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1, 409-412.

[14] Sun, F., and Hu, G. (1998). "Speech recognition based on genetic algorithm for training HMM", *Electronics Letters*, *34*(16), 1563-1564.

[15] Young, S. (1999). "Acoustic modeling for large vocabulary continuous speech recognition", *Springer Berlin Heidelberg*, *Computational Models of Speech Pattern Processing*, 18-39.

[16] Zheng, F., Song, Z., Xu, M., Wu, J., Huang, Y., Wu, W., and Bi, C. (1999). "Easytalk: a large-vocabulary speaker-independent Chinese dictation machine", *EuroSpeech*.

[17] Pruthi, T., Saksena, S., and Das, P. K. (2000). "Swaranjali: Isolated word recognition for Hindi language using VQ and HMM", *International Conference on Multimedia Processing and Systems, ICMPS, IIT Madras*.

[18] Rao K. (2000). "Speaker Independent Isolated Digit Voice Recognition Using Discrete Hidden Markov Model", *Master's thesis, IIT Kanpur.*

[19] Nilsson, M., and Ejnarsson, M. (2002). "Speech Recognition System using Hidden Markov Model", *Master's thesis, Blekinge Institute of Technology, Sweden.*

[20] Woodland, P. C., and Povey, D. (2002). "Large scale discriminative training of hidden Markov models for speech recognition", *Computer Speech and Language*, *16*(1), 25-47.

[21] Nwe, T. L., Foo, S. W., and De Silva, L. C. (2003). "Speech emotion recognition using hidden Markov models", *Speech communication*, *41*(4), 603-623.

[22] Sheh, A., and Ellis, D. P. (2003). "Chord segmentation and recognition using EM-trained

hidden Markov models", *International Society for Music Information Retrieval, ISMIR*, 185-191.

[23] Hasan, M. R., Jamil, M., Rabbani, M. G., and Rahman, M. S. (2004). "Speaker Identification Using Mel Frequency Cepstral Coefficients", *International Conference on Electrical and Computer Engineering, ICECE*,565-568.

[24] Hyassat, H., and Zitar, R. A. (2006). "Arabic speech recognition using SPHINX engine", *International Journal of Speech Technology*, *9*(3-4), 133-150.

[25] Sha, F., and Saul, L. K. (2006). "Large margin Gaussian mixture modeling for phonetic classification and recognition", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ,ICASSP*, 1, 265-268.

[26] Sha, F., and Saul, L. K. (2007). "Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models", *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 4, 313-316.

[27] Satori, H., Harti, M., and Chenfour, N. (2007). "Introduction to Arabic speech recognition using CMUSphinx system", *Proceeding of the Information and Communication Technologies International Symposium, ICTIS*.

[28] Bhuriyakorn, P., Punyabukkana, P., and Suchato, A. (2008). "A genetic algorithm-aided Hidden Markov Model topology estimation for phoneme recognition of thai continuous speech", *IEEE Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD,* 475-480.

[29] Alotaibi, Y. A. (2008). "Comparative study of ANN and HMM to Arabic digits recognition systems", *Engineering Sciences, 19*(1), 43-60.

[30] Azmi, M., Tolba, H., Mahdy, S., and Fashal, M. (2008). "Syllable-based automatic Arabic speech recognition", *Proceedings of the 7th WSEAS International Conference on Signal Processing, Robotics and Automation*, World Scientific and Engineering Academy and Society, WSEAS, 246-250.

[31] Elshafei, M., Al-Muhtaseb, H., and Al-Ghamdi, M. (2008). "Speaker-independent natural Arabic speech recognition system", *International Conference on Intelligent Systems*.

[32] Jančovič, P., and Köküer, M. (2009). "Incorporating the voicing information into HMM-based automatic speech recognition in noisy environments", *Speech*

Communication, *51*(5), 438-451.

[33] Satori, H., Hiyassat, H., Harti, M., and Chenfour, N. (2009). "Investigation arabic speech recognition using CMU sphinx system", *International Arab Journal of Information Technology*, *6*(2), 186-190.

[34] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., and Woodland, P. (2009). *The HTK Book.* Cambridge University.

[35] Al-Qatab, B. A., and Ainon, R. N. (2010). "Arabic Speech Recognition Using Hidden Markov Model Toolkit (HTK)", *IEEE International Symposium in Information Technology, ITSim*, 2, 557-562.

[36] Abushariah, A. A. M., Gunawan, T. S., Abushariah, M. A. M, and Khalifa, O. O. (2010). "English Digits Speech Recognition System Based on Hidden Markov Models.", *International Conference on Computer and Communication Engineering, ICCCE.*

[37] Kumar, R. (2010). "Comparison of hmm and dtw for isolated word recognition system of punjabi language", *Springer Berlin Heidelberg, Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 244-252.

[38] Gupta, R. (2011). "Speech Recognition for Hindi", *Master's thesis, IIT, Bombay.*

[39] Kumar, K., and Aggarwal, R. K. (2011). "Hindi speech recognition system using HTK", *International Journal of Computing and Business Research*, *2*(2), 2229-6166.

[40] Paul, A., and Chayani, S. (2011). "Speech Recognition in Hindi", *Master's thesis, National Institute of Technology, Rourkela.*

[41] Dua, M., Aggarwal, R. K., Kadyan, V., and Dua, S. (2012). "Punjabi Automatic Speech Recognition using HTK", *International Journal of Computer Science Issues*, 9(4), 359-364.

[42] Ghai, W., and Singh, N. (2012). "Analysis of automatic speech recognition systems for indo-aryan languages: punjabi a case study", *International Journal of Soft Computing and Engineering, IJSCE*, 2231-2307.

[43] Kumar, K., Aggarwal, R. K., and Jain, A. (2012). "A Hindi speech recognition system for connected words using HTK", *International Journal of Computational Systems Engineering*, *1*(1), 25-32.

[44] Vimala, C. M., and Radha, V. (2012). "Speaker Independent Isolated Speech Recognition

System for Tamil Language using HMM", *Procedia Engineering*, *30*, 1097-1102.

[45] Choudhary, A., Chauhan, M. R., and Gupta, M. G. (2013). "Automatic Speech Recognition System For Isolated and Connected Words Of Hindi Language By Using Hidden Markov Model Toolkit (HTK)".

[46] Manjunath, K. E., Rao, K. S., and Pati, D. (2013). "Development of phonetic engine for Indian languages: Bengali and Oriya", *IEEE International Conference* on *Oriental COCOSDA Conference on Asian Spoken Language Research and Evaluation, O-COCOSDA/CASLRE*, 1-6.

[47] Saini, P., Kaur, P., and Dua, M. (2013). "Hindi Automatic Speech Recognition Using HTK", *International Journal Of Engineering Trends And Technology*, *4*.

[48] Sarma, B. D., Sarma, M., Sarma, M., and Prasanna, S. R. M. (2013). "Development of Assamese Phonetic Engine: Some Issues", *Annual IEEE India Conference, INDICON*.

[49] Thakuria, L. K., Acharjee, P., Das, A., and Talukdar, P. H. (2013). "BODO Speech Recognition based on Hidden Markov Model Toolkit (HTK)", *International Journal of Scientific and Engineering Research*, 4(12), 2309-2313.

[50] Tripathy, S., Baranwal, N., and Nandi, G. C. (2013). "A MFCC based Hindi speech recognition technique using HTK Toolkit", *IEEE Second International Conference on Image Information Processing, ICIIP*, 539-544.

[51] Mankala, S. R., Bojja, S. R., Ramaiah, V. S., and Rao, R. R. (2014). "Automatic Speech Processing Using HTK for Telugu Language", *International Journal of Advances in Engineering and Technology*, *6*(6), 2572-2578.