Prosodically Guided Phonetic Engine for Punjabi Language

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology

in

Computer Science and Applications

Submitted By

Neeshu Agarwal

(Roll No. 601303021)

Under the supervision of

Dr. R.K. Sharma

Professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2015

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Prosodically Guided Phonetic Engine for Punjabi Language*", in partial fulfillment of the requirements for the award of degree of Master of Technology in *Computer Science and Applications* submitted in Computer Science and Engineering Department 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 the thesis has not been submitted for award of any other degree of this or any other University.

(Neeshu Agarwal)

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

(R.K. Sharma)
22/7/15
Professor, CSED

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENTS

The completion of thesis work mainly concerned with many people, who contributed directly or indirectly through their 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 owed 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. Deepak Garg**, Associate Professor and Head, Department of Computer Science and Engineering, for his support and motivation that encouraged me for the dissertation work.

I am also in debt to **Dr. Rajesh Kumar**, Associate Professor and former Head (SMCA), 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 Department of Computer Science and Engineering (CSED) 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.

(Neeshu Agarwal)

ABSTRACT

This thesis deals with prosodically guided phonetic engine for Automatic Speech Recognition (ASR) for Punjabi language using trainable systems. The goal of this work is the development of a phonetic Engine using Phonetic Transcription and so to build acoustic models for Punjabi language. This is done by employing Hidden Markov Models (HMMs) that provide statistical representation of each of the distinct sounds that make up a word and to train the parameters of the models developed. The speech recognition modelling can be done in two ways: Acoustic modelling and Language Modelling. Out of these two, Acoustic Modelling has been used. In this work, Acoustic modelling has been worked out at a phonetic level, allowing general speech recognition applications. For this purpose, the tool HTK is employed, stated as Hidden Markov Model toolkit. HTK uses different commands to produce HMM models for each phone and computing various parameters for mixture. Different phone models have been developed and tested. This thesis is divided into six chapters. A brief review of these chapters is given below.

Chapter 1 includes the definitions of the terms and the brief idea of the concepts, description of tools used in this work.

Chapter 2 includes the literature survey which depicts the ideology, and previously proposed models and methodologies to train a phonetic engine.

Chapter 3 depicts the problem statement and its cause that motivates to work in this domain.

Chapter 4 describes the process of data collection of read, lecture, and conversational modes of Punjabi Speech, experimental set up and methodology which depicts the process of development of prosodically guided phonetic engine. For this purpose, Ubuntu Linux (14.04 64 bit) has been considered as environment. Wave Surfer tool has been used for manual transcription and Hidden Markov Model toolkit has been used to train and test the phonetic engine for various modes of speech.

Chapter 5 presents the results of phonetic engine developed in this work for two categories, 30 phones and 34 phones. Gender-wise and transcription-wise performance of the developed engine have also been presented in this chapter.

Chapter 6 includes the conclusion and future scope of this work.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBRIVATIONS

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 Modelling
CD-HMM	Continuous-Density Hidden Markov Model

CMU	Carnegie Mellon University
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

CHAPTER 1

Introduction

All of us are living in a digital age in which our digital gadgets like desktops, laptops, tablets, phones (or we should say smart phones), music players, digital-watches, etc. are playing a vital role in our lives. In current scenario, our life sounds incomplete without these gadgets. All of these gadgets are the result of technological advancements and intelligence of human brain. In the early years of digital age, we used to interact with computers using hardware components along with their compatible software program. All these interactions made tasks like mathematical calculations and computations such as calculus computations, co-ordinate and geometry calculations, building design and beam bending calculations etc., very easy. Office work like preparations of office reports including graphs and charts, simulation of lab tests and their analysis work, data collection and its analysis in a mega retail store, medical report preparation etc. were also made easy to execute.

Nowadays, all of these tasks can't be possible without being comfortable and proficient in digital technology. We use the provided hardware component and compatible software component to deal with these gadgets. As our society is getting increasingly dependent on these digital devices, there is an imminent requirement of making these devices hand-free and voice-controlled. Making the digital devices speech-controlled will improve the output of our work a great deal. The necessity of such type of speech-controlled devices and software programs, which are able to direct the digital machines, has increased in recent years. It has happened due to the requirement to make the current tasks easier for every person of the society and due to the fact that research in this domain is at its peak and is touching new heights every day. A real life example that we can consider is of a disabled person who can't type on a manual keyboard. It would be very difficult for him to use computers. In order to empower such people to use technology independently and live a dignified and progressive life, it is necessary to have an application which will make controlling computers and digitally assisted devices, easy in their own native language. Here, language plays an important role as not every person is fluent or proficient enough in English, German, French or any other standard language to use the computers. Many studies have proved that India and other Asian countries contribute the highest number of digital technology users and most of them prefer to work in own their native language. Another reason of using their native language in work is to

save and propagate their heritage and language for their next generation. In this digital age, if regional languages and heritage is left behind because of technological trends, it will not take long for these languages to be classified as 'endangered'. This requirement enforces the researchers to explore the domain of speech recognition for Punjabi and other such languages.

Both Speech recognition and speech synthesis 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.*). Symbolic linguistic representation consists two elements, Phonetic transcription and prosody information. T hen, 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 a module that uses the acoustic phonetic information for converting the speech signal into symbolic form. This symbolic form is nothing but the basic sound units presenting 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 Terminologies

1.1.1 Phoneme

Every language has a smallest and most fundamental unit of sound, named as phoneme. When combined with other phonemes, they make meaningful units such as words. There is a distinction between phone and phoneme. A phone is only a sound and is infinite in number. It is not necessary that combining different phones would produce some meaningful unit. It can simply be a noise, word, animal cry, etc. However, phoneme always produces a meaningful unit. For example, words 'madder' and 'matter', both composed of different phonemes but in American English, both words sounds same when pronounced, *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 in different languages 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. After several experiments, it has been observed that the language model depends on the recognizer being used. For instance, the transcription of radiologically dictated reports requires different language model than the movie reviews. If text is to be produced, then the language model may reasonably be constructed by processing examples of corresponding written materials.

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 'ਗੁਰਮੁਖੀ', this can be transcribed as 'gʊrmukʰi'. Phonetic transcription and orthography provide different functionality. Orthography includes rule of spelling. In order to write a particular language in its own flavor, Orthography provides a standardized system. 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 another advantage of transcribing the words using IPA chart, which is the fact that computer can be made to understand utterances of any language as all languages can be transcribed into IPA Format.

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
ਗੁਰੂ ਰਾਮਦਾਸ ਜੀ ਕਿਰਪਾ ਕਰੋ	gʊru ramdas ʤi kɪrpa kəro
ਸ਼੍ਰੀ ਗੁਰੂ ਤੇਗ ਬਹਾਦੁਰ ਜੀ	ʃri gʊru teg bəhadʊr ʤi
ਸਾਰਾ ਮਨੁੱਖੀ ਪਰਿਵਾਰ ਆਪਣੀ ਮਹਿਮਾ	sara mənʊkʰi pərɪvar apɳi məhɪma
ਵਾਹਿਗੁਰੂ ਜੀ ਕਾ ਖਾਲਸਾ ਵਾਹਿਗੁਰੂ ਜੀ ਕੀ ਫ਼ਤੇਹ	vahɪgʊru ʤi ka kʰalsa vahɪgʊru ʤi ki pʰteh
ਸੱਜੇ ਹਥ ਮੁੜ ਜਾਣਾ	səʤʤe həʈʰ mʊɳ ʤaɳa

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

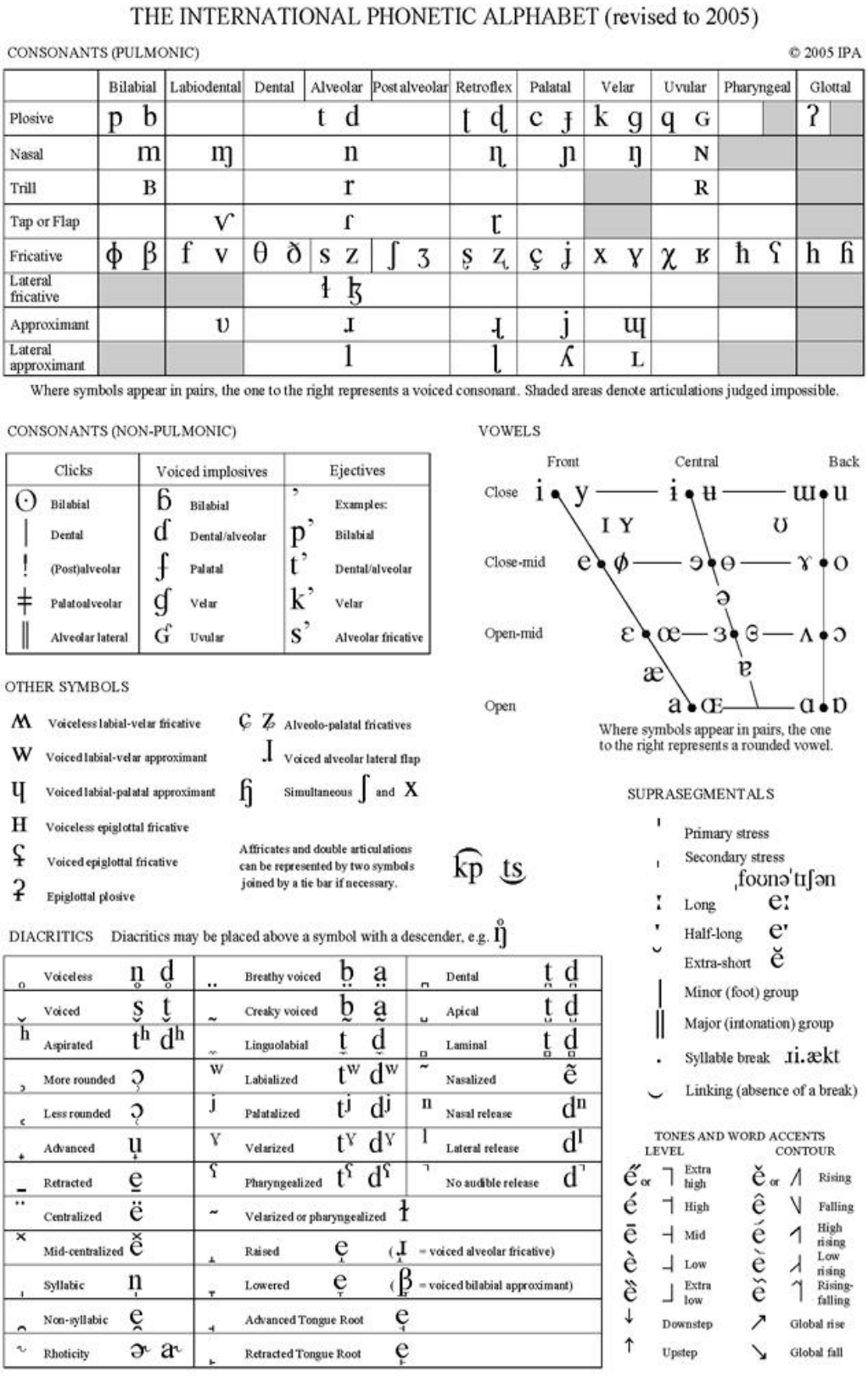


Figure 1.1: IPA Chart

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 is happening on the other side. Firstly, 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 is exactly going on the other side.

An 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 in two categories, 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,'ਗੁਰਮੁਖੀ' **(gʊrmukʰi)** which 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 time, then it is 's' for longer 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 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 the HMMs for phoneme, a stationary part is sandwiched between two transition parts). After concatenation of smaller phonetic HMMs, larger HMMs are produced which represent the words.

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. These observations can either be scalars or vectors.

For discrete observation HMM, following notations are used.

M = number of observation symbols.

$Q = \{q_1, q_2 \ldots\ldots q_n\}$ are the states.

$O = \{o1, o2 \ldots\ldots op\}$ are the discrete set of possible symbols observations.

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

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

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

1.3.2 Continuous Observation HMM

In Continuous Observation HMM, all the notations which are used indiscrete observation HMM, denote the similar meaning here 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 only be log-concave or elliptically symmetric density. The most 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.

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

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

$b_{jm}(o_p)$ 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.

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

…….(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.

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

$$b_{jm}(o_p) = N\left(o_p, \mu_{jm}, \sum jm\right) = \left\{\frac{1}{(2\pi)^{d/2}} \left(\prod_{l=1}^{d} \sigma_{jml}\right)^{1/2}\right\} e^{-\sum_{l=1}^{d} \frac{o_{pl} - \mu_{jml}}{2\sigma_{jml}^2}}$$

……(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 on the basis of 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.

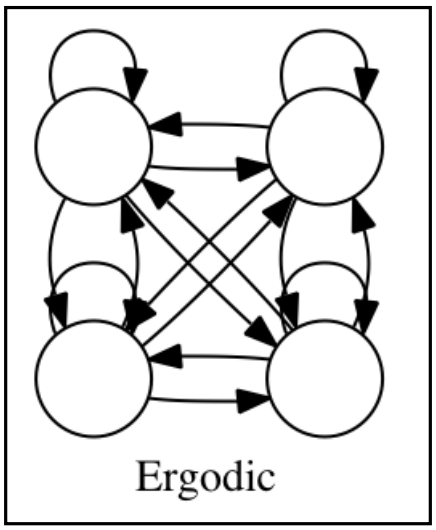


Figure 1.2: Ergodic Topology

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, i.e. from left to right. This topology is generally used for speech recognition.

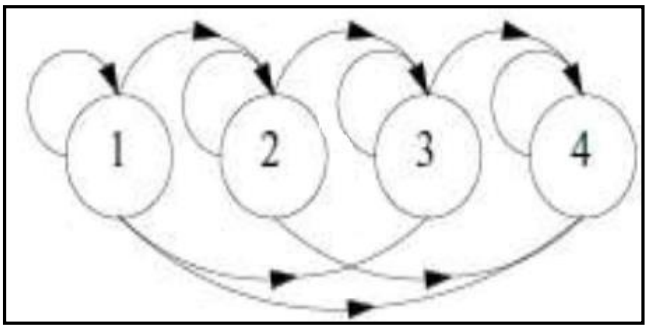


Figure 1.3: General Left-to-Right Topology

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

$$A = \begin{bmatrix} a11 & a12 & 0 & 0 \\ 0 & a22 & a23 & 0 \\ 0 & 0 & a34 & a34 \\ 0 & 0 & 0 & a44 \end{bmatrix}$$

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 $O = o_1, o_2 \ldots o_p$ 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

  $O = o_1, o_2 \ldots o_p$ 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 $O = o_1, o_2 \ldots o_p$ are given and the probability that model *M* has produced this sequence is to be computed. To resolve this issues, 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

$O = o_1, o_2 \ldots o_p$. 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

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 sequences 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 divided 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. The process of recognition is shown in Figure 1.4.

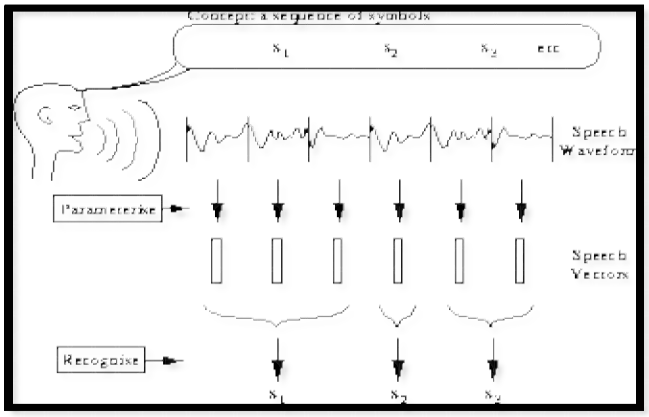


Figure 1.4: Speech recognizer

1.3.5.2 Elements of HMM

- These are finite number of states '*N*' such that signal possesses distinct properties within each state.
- 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 tool kit 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.

Using training utterances (recorded speech used for training purpose) and their corresponding transcriptions, the HTK training tools compute the parameters of set of HMMs. Unknown utterances (recorded speech whose transcription is to be done) are transcribed using the HTK recognition tools. Working of phonetic engine is shown in Figure 1.5.

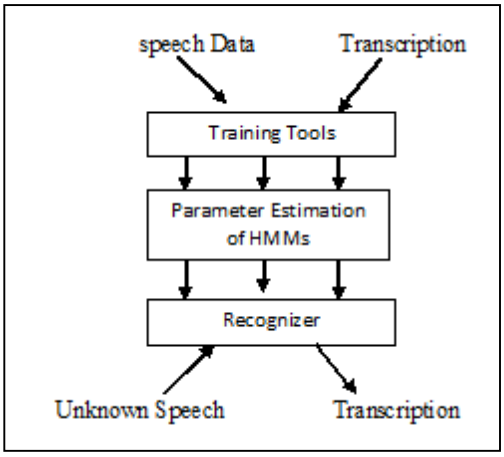


Figure 1.5: Working of phonetic engine

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. The process of training is shown in Figure 1.6.

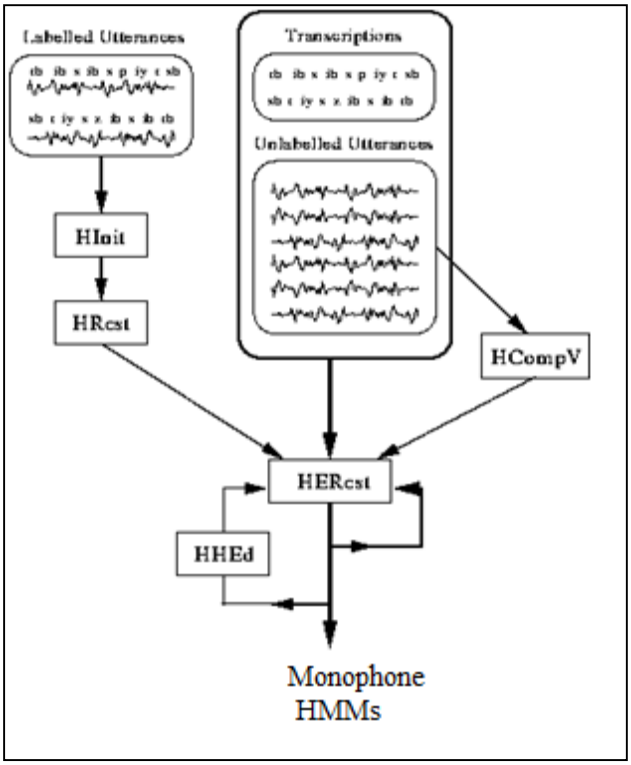


Figure 1.6: Training HMMs

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 has 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 and so on. In our case, it is a single file namely, hmmdef, which defines all the mono phone HMM models, then carries out the editing operations as mentioned in the cmds.hed and then writes 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, mixtures are needed to be computed twice, which is defined in a file having extension .hed say, mix_2.hed. Figure 1.7 shows the commands defined in mix_2.hed.

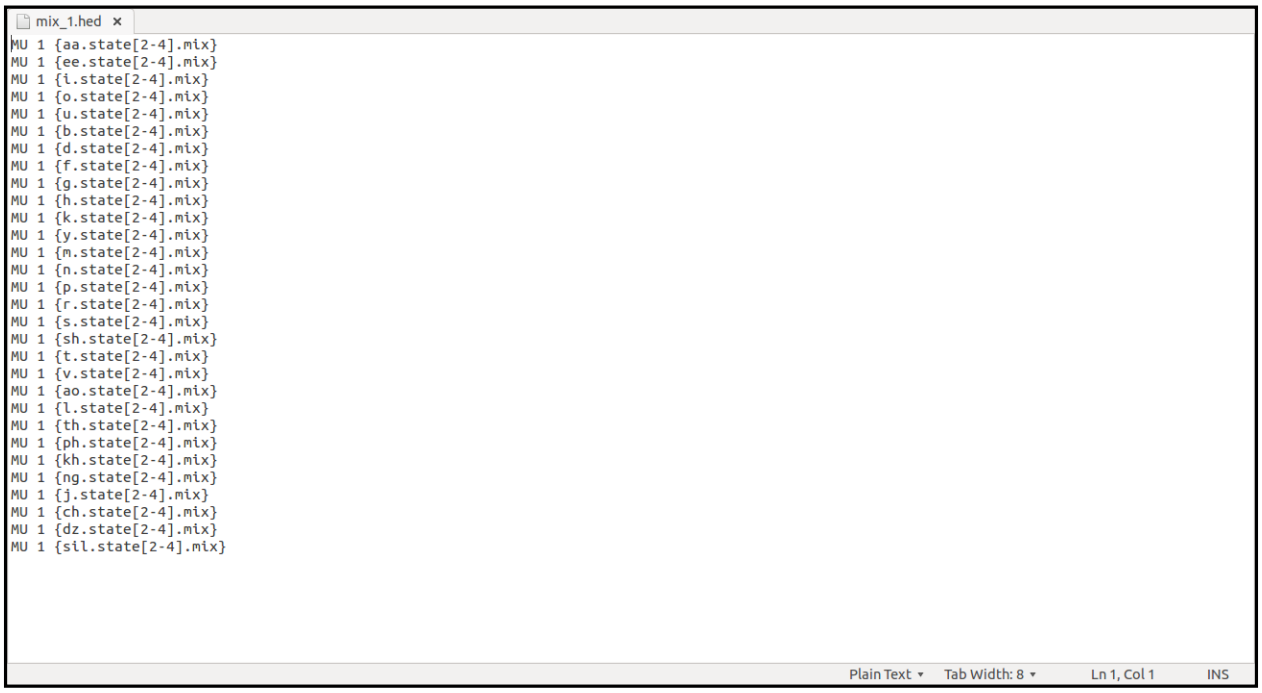


Figure 1.7: HMM Editor - HHED mix_1.hed

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 performs 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 performs isolated-unit training and HERest operates 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, Testing is done to check the accuracy of trained system. For Testing, HTK uses some of the tools which are given in Figure 1.8.

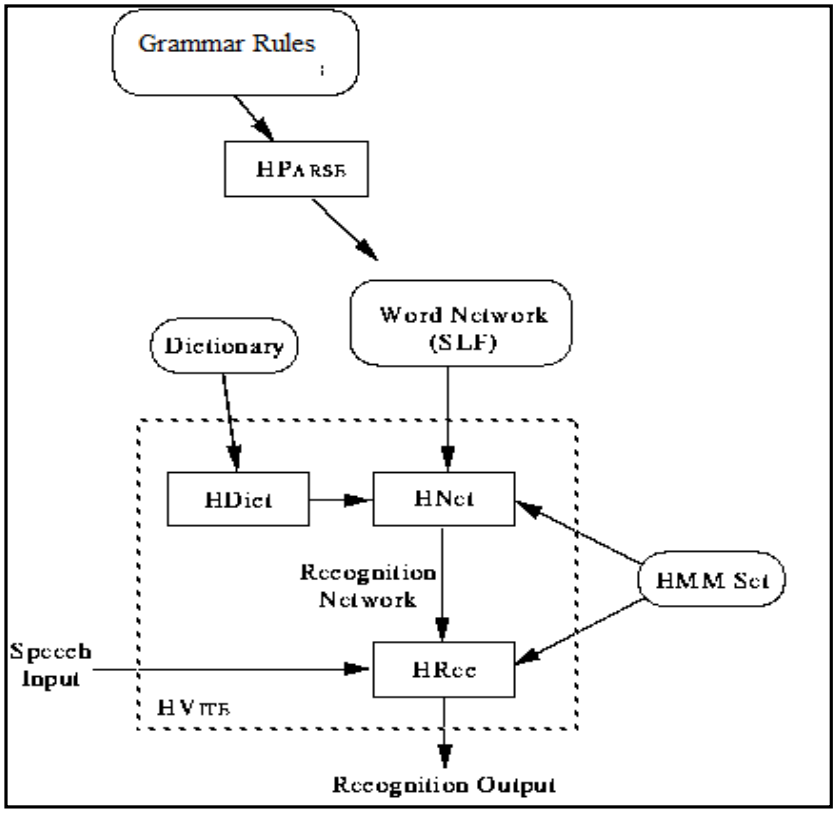


Figure 1.8: Recognition process

HVite

HTK provides a tool HVite which is responsible for recognition, *i.e.*, it provides the transcription of unlabeled 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).

The working of HVite and HResults is mentioned in Figure 1.9.

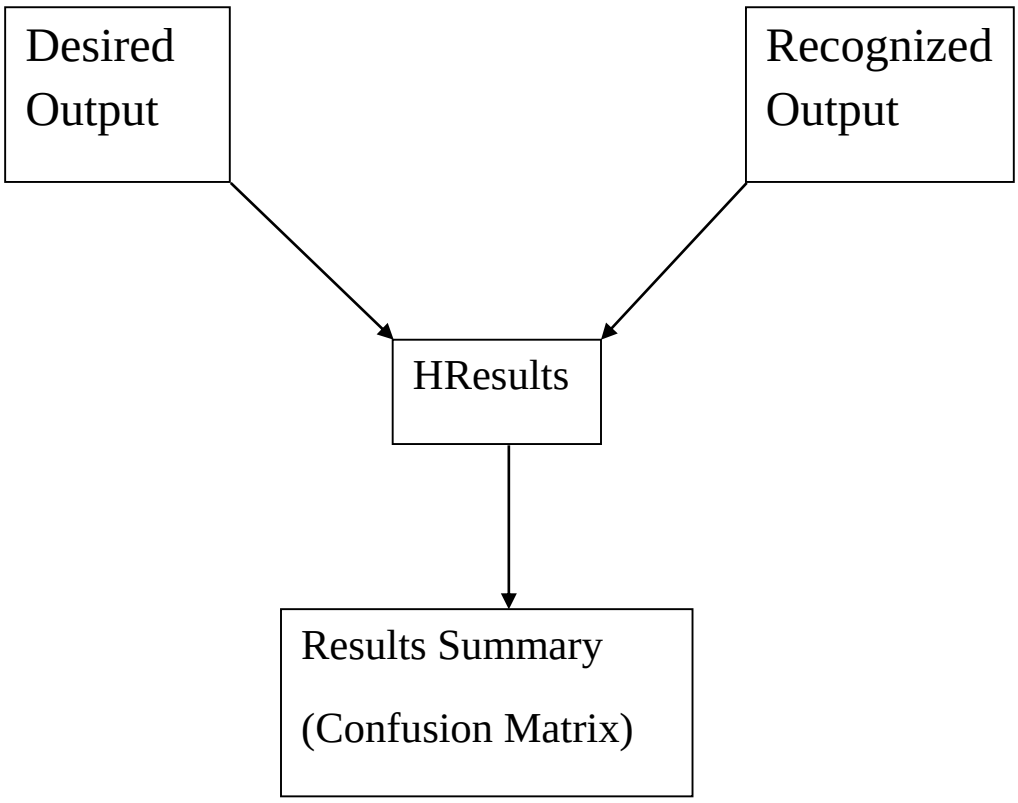


Figure 1.9: Working of HResults

1.5 Feature Extraction Technique

It is theoretically possible that speech can be recognized directly from digital waveform but practically it is an extremely 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 and then windowing is done on each frame. 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 : $w(n) = 0.5(1-\text{Cos}\ (2*pi*n / (N-1)))$
- Hamming Window : $w(n) = 0.54 – 0.46\ \text{Cos}\ (2*pi*n / (N-1)))$
- Cosine Window : $w(n) = \text{Cos}\ ((pi\ *n/n-1) – (pi/2))$

Most common window is hamming window. Figure 1.10 shows the sample graphs for each of window.

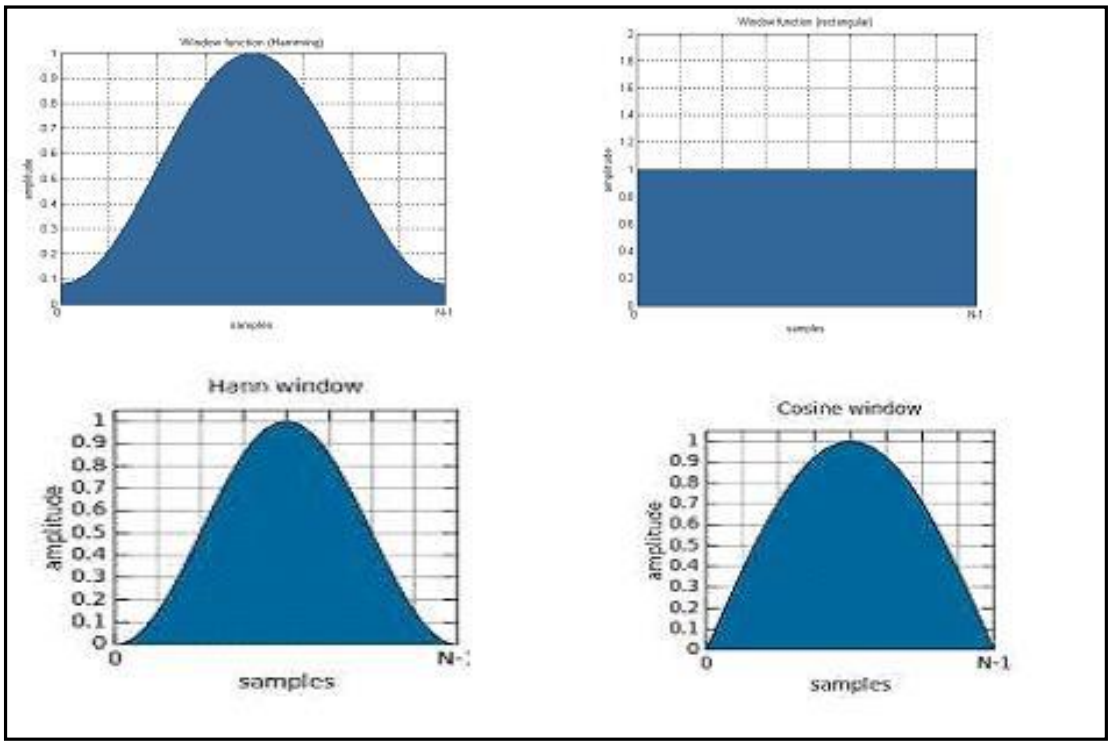


Figure 1.10: Window Functions

Human beings have the ability of differentiating among sounds of same words, even when spoken by different people of different geographical areas. Even when different people speak same word differently, we 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. The process of feature extraction is shown in Figure 1.11.

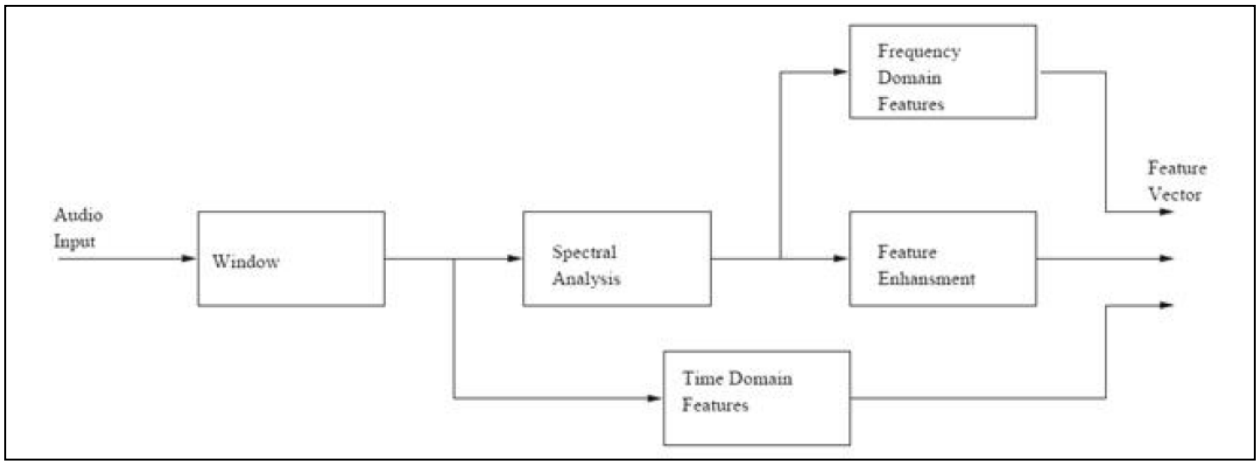


Figure 1.11: Feature extraction

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.

MFCC can be computed using the given equation.

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

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).

The complete pipeline of computing MFCCs as a feature extraction technique is shown in Figure 1.12.

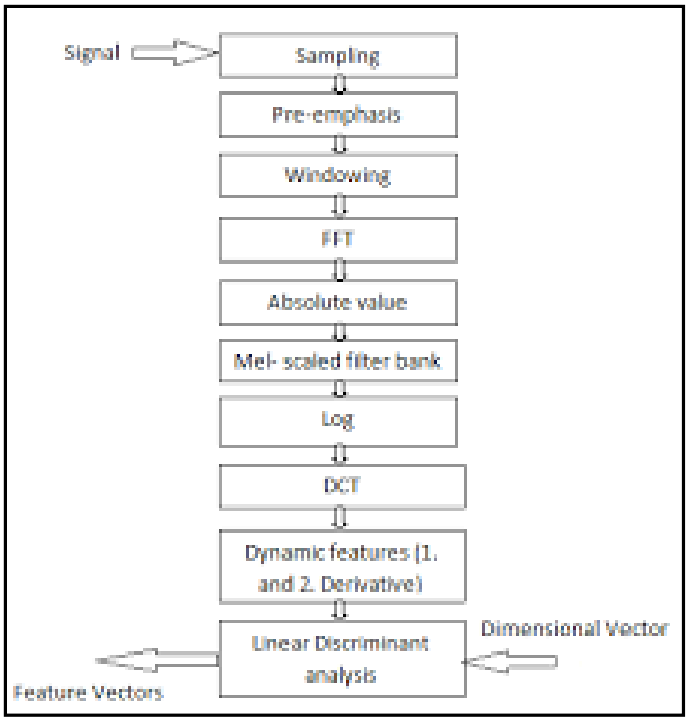


Figure 1.12: Complete Pipeline of MFCC

CHAPTER 2

Literature Survey

In this thesis, literature survey is divided into three parts, the first part deals with existing recognition systems that have been developed using HTK toolkit, second part deals with those existing recognition systems that have been developed using HMMs but use some other tools and feature extraction technique and the third part copes with the existing recognition systems that have been developed using MFCC technique as a feature extraction.

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 twelve cepstrum's, normalized log energy, delta and acceleration coefficients, *i.e.,* first and second order derivatives of parameters. In this work, vocabulary consisting of 65464 words, four-gram language model was used.

Azmi *et al.* (2008) proposed automatic speech recognition for Arabic speech using syllables and did a comparison of mono-phone, tri-phone 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 forty four speakers. MFCCs were used as feature extraction technique. It was observed that syllable based recognition overcome the performance of other recognizers as recognition rate of mono-phone based recognizer, tri-phone based recognizer, word based recognizer was 90.7%, 92.2%, 91.6% and of syllable based-recognizer, it was 93.4%.

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 were based on tri-phones. Data was collected from thirteen Arabian native speakers which further divided into ten speaker dependent and three speaker independent

data for evaluating the performance of the system. For both Continuous and isolated kinds of speech, it was observed that the system was able to recognize the speech for speaker dependent and speaker independent categories. The overall performance of system was 90.6% for sentence correction, 98.0%for word correction and 97.9% 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 thirty 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 94.6% and word error rate was 5.4%.

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 one hundred fifteen 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 94.0% to 96.0%.

Kumar *et al.* (2012) proposed connected-words speech recognition system for Hindi language based HMMs using HTK toolkit. Training data was collected from twelve speakers including both males and females and five speakers data was collected for testing purpose. The system was trained for recognizing any sequence of words from a vocabulary of size one hundred two 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, and 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 87.0%.

Choudhary *et al.* (2013) proposed automatic speech recognition for Hindi language using HMM. Proposed system was developed for recognizing the isolated and connected words of Hindi speech and implemented using HTK toolkit. System was trained for hundred different isolated words and each word was uttered ten times. After evaluation the performance of system recognition rate was 95.0% and word error rate was 5.0%.

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, and 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 93.9%, with eight states in HMM topology accuracy was 91.7%, with 10 states in HMM topology accuracy was 96.6%. In second case, with 6 states in HMM topology accuracy was92.7%, with 8 states in HMM topology accuracy was 91.2%, with 10 states in HMM topology accuracy was 95.5%.

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 47.3% in read speech mode, 45.3% in lecture speech mode and 36.1% 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 dependent 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.

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 96.0% was obtained and with null grammar word recognition accuracy was 82.0%.

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, 64.0% was the recognition rate with context-independent phone models and with right context dependent phone models recognition rate was 73.8%.

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 was 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, 60.0% 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 mono-phone 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 tri-phones. It was observed that with supervised adaptation 37.0% error reduction was achieved and with unsupervised adaptation 32.0% 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 get the trade-off required between low computations 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 69.0% of accuracy.

Ming *et al.* (1998) proposed phone recognition system for continuous speech signal based on Bayesian trip hone models. In this work, a new statistical framework was introduced for building tri-phone 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 98.0% accuracy for in-vocabulary commands and 95.0% 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 an N-gram model. MFCCs were 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 tri-phones 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 92.1% 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 84.5% was the accuracy of system for speaker 1 and for speaker 2 it was 84.3%.

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 quin-phone 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 75.0% 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 Matlab 6.1 in windows XP 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 100.0% 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 100.0% 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 Holy Qur'an and standard Arabic language. In this work, three corpus were developed completely: Holy Qur'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) proposed a novel approach for building an automated Speech recognition System for Arabic language. 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 86.7%, for speaker 2 it was 86.7% and for speaker 3 it was 83.3%.

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 4.4% 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 24.6% for standard model, 27.1% for missing feature training (MFT) model and 21.3% for multi-conditional training model. MFCC was used as a feature extraction technique.

Satori *et al.* (2009) proposed a novel approach for building an automated Speech recognition System for Arabic language. 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. The 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 were 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 96.7%, for male speaker 2 it was 93.3%, for male speaker 3 it was 93.3%, for female speaker 1 it was 86.7%, for female speaker 2 it was 83.3% and for female speaker 3 it was 90.0%.

Kumar *et al.* (2010) proposed comparison between HMM and Dynamic Time warp (DTW) technique for speaker dependent 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 Windows 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 88.0% and 67.0% accuracy. In continuous speech recognition tested in clean environment, multi-speaker mode and speaker independent mode achieved 72.5% and 56.3% accuracy and in noisy environment, multi-speaker mode and speaker independent mode achieved 82.5% and 76.7%.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 speech recognition system for Tamil language. It was speaker independent in nature. 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. Word error rate (WER) was the parameter that was considered for measuring the performance. Data was collected from ten speakers out of which four speaker's data were used for performance evaluation. For these experiments, two thousand and five hundred was the vocabulary size and 88.0% of accuracy was observed with minimum word error rate of 0.88.

2.3 MFCC as a Feature Extraction Technique

Tiwari (2010) exploited the characteristic of speech signal that speech signal and all of its corresponding spectral properties are a function of time, therefore the time varying Fourier representation has been used to analyze the spectral properties. Energy, correlation and other temporal properties considered as a constant for short duration of time. It was done because, during a short time segment of continuous speech, the values of temporal properties represent it as a stationary signal. That's why; Speech or wave signal has been sliced into a number of segments of short duration of time, using hamming window technique in order to apply normal Fourier transform. The Mel frequency Cestrum Coefficient (MFCC) feature has been used for designing a text dependent speaker identification system. First of all MFCCs of each speaker has been computed in both training and testing phase and then Euclidean distance between each speaker has been measured and the speaker with minimum Euclidean distance assumed as the correct speaker.

Tiwari (2010) experimented the process with different no. of filters as 12, 22, 32, 42 and efficiency occurred as 65.0%, 75.0%, 85.0% and 80.0% respectively. The experiment with 32 filters has been revised with 2 different windows named as Hanning window and rectangular windows and efficiency occurred as 75.0% and 55.0% respectively.

Chavan *et al.* (2013) implemented and measured the performance of Text Dependent Speaker Independent Isolated Word Speech Recognition System which was developed using HMM and MFCC as a parameterization or feature extraction technique. These MFCC features / parameters employed in the training of the system. Forward backward algorithm with EM principle has been used for parameter estimation and re-estimation in HMM modeling of the system. For each and every state of HMM, GMM has been used to model the distribution of speech parameters/ features. In the final, the calculated HMM parameter values for vocabulary words were stored in corresponding HMM models as a reference database. The probability of generation of speech

observations and the most probable path sequence using each stored HMM model has been calculated to recognize the spoken utterance. At the end, the one with maximum likelihood path i.e. path with maximum probability has been selected as recognized word. The system has been trained with their own built database, which consists of 60 speech samples of selected words and tested. In Noisy environment, for particularly selected three words, the recognition accuracy was 92.0%, 92.0% & 88.0% respectively.

Dhingra *et al.* (2013) discussed an avenue of isolated speech recognition through the MFCC and DTW has been described. Myriad feature parameters has been extracted from a wave signal of spoken utterance. A speech database of total five speakers has been prepared for the experiment. Each of these speakers spoke 10 digits under acoustically controlled room. Then as a process of feature extraction, MFCC computed from wave signal of spoken utterance. To cope with variation of speaking speeds, Dynamic Time Warping (DTW) has been used. DTW has been used for measuring the match between two speech segments, which may fluctuate in time or speed. The FFT/DCT and the MFCC differ in the way of their operation. On the mel-scale, the frequency bands are aligned logarithmically in the case of MFCC, whereas in the case of FET or DCT, these frequency bands are linearly spaced. Logarithmically aligned frequency bands i.e. MFCC fits the response of human auditory system much better than the linearly aligned frequency bands of FFT or DCT. Implementation of extraction of MFCC parameters in feature extraction algorithm is less complex than FFT or DCT. Some coefficients of these logarithmically aligned frequency bands corresponding to the frequencies of Mel scale of speech Cepstrum are computed with the help of spoken word samples that were stored in database. The project was aimed to recognize isolated speech utterances using MFCC and DTW approaches. MFCC technique was applied for the purpose of feature extraction and DTW approach was applied to resolve the issue of feature matching. For mapping the unknown speech utterance with the speech database, a variation measure based on minimizing the Euclidean distance has been applied. Using MATLAB, the experimental results has been analyzed and it has been proved that the results were efficient for the experiments. This process is also applicable for any number of speakers. The project has been demonstrated that DTW is the best nonlinear feature matching technique in speech recognition that results minimal error rates as well as fast computing speed.

Umarani (2014) suggested that speech recognition system based on Digital Signal Processor with enhanced performance score in terms of accuracies and cost of computation. The extensive survey comprises various techniques of feature parameters extraction like Mel filter banks with Mel Frequency Cepstrum Coefficients (MFCC). The work demonstrated a technique of isolated speech recognition by Digital Signal Processor TMS320C6713 using Mel scale Frequency Cepstral Coefficients and Euclidean distance. Various speech features a vector has been computed from speech signal of spoken utterance. An experimental speech database, inclusive of total five speakers has been built, in which each speaker spoke 5-10 words under acoustically controlled room. MFCC were extracted from speech signal of spoken utterance. Concept of minimal Euclidean distance has been applied to compare inter speaking differences. For a real time database of 15 words, the accuracy of system was 76.7% with 30.0% of sensitivity.

Gin *et al.* (2015) implemented an Automatic Speech Recognition System. The system was based on the mentioned steps: Preprocessing of speech signal, Feature vector Extraction (MFCC- Mel Frequency Cepstrum Coefficients) and Hidden Markov Model (used in training and training as well as recognition phase). The purpose of the work was to convert the human voice into a digital machine readable input, for ex. binary-coded, or sequence of characters. The three models such as Acoustic phonetic model, Knowledge based approach and Pattern recognition model that has been used in speech recognition. Pattern recognition technique has been used in this paper. This technique didn't require any kind of explicit knowledge of speech. This approach has two fundamental steps: the first one is training of speech utterances (this pattern of speech utterances was dependent on some generic feature spectral parameter set and recognition of utterances through pattern matching).This study of recognition and Hidden Markov Model was further applied to design a speech based GUI system. In myriad applications, the designed GUI system is applicable. These applications can be associated with persons with disability. With the help of ASR (Automatic Speech Recognition) system, such kind of persons will be able to use computer with their voice commands. Another application can be for those users who are not comfortable with English language and love to work with their native language like Hindi, Bangla, and Tamil etc.

Muda *et al.* (2010) Feature Extraction and Feature Matching, digital signal processes that were applied to represent the speech signal. Myriad techniques like LPC, HMM, ANN and others were exercised with an

intention to an efficient and effective method for speech utterances. As a first step pre-processing and signal filtering has been done then matching process was implemented. To model the human auditory perception system, the non-parametric, logarithmically aligned frequency bands, Mel Frequency Cepstral Coefficients (MFCCs) has been extracted. The coin-flipper of Dynamic Time Warping (DTW), Sakoe Chiba, has been applied for features matching process. The work presented the feasibility of MFCC to compute speech feature parameters and DTW to match the test patterns.

Glass *et al.* (1996) demonstrated the speech recognizer that generate the segment-based network based on frames, in which each segment is represented by fixed-dimensional features. In this approach, a temporal network of features vectors has been formed that was used by these feature based recognizers. A slice of single speech utterance used the subset of speech vectors. The work is based on the maximum ***a posteriori*** decoding strategy and by using segment based Viterbi (A* search), speech recognizer has been developed. The phonetic experiment has been exercised on TIMIT speech corpus which resulted the 64.1% and 69.5% of diaphonic accuracy for context-independent and context-dependent testing data set.

Han *et al.* (2006) proposed a new algorithm for computing MFCC for speech recognition. The proposed algorithm has been reduced the involvement of computation complexity by 53.0% in comparison to the conventional MFCC extraction algorithm. The simulation of proposed algorithm resulted accuracy of 92.9% in recognition of speech utterances. Reduction of 1.5% has been noted in recognition accuracy as compared to the conventional MFCC algorithm. In the conventional MFCC extraction algorithm, the accuracy was of 94.4%. It is also important to note that to implement the proposed algorithm, the required number of logic gates were approximately half of the conventional MFCC algorithm. And this proposed technique of employment of logic gates in the new algorithm made the hardware implementation very efficient.

Yao *et al.* (2006) said a speech recognition approach with mixed parameters has been demonstrated. In this approach, as a feature parameter, fractal feature and traditional MFCC features has been combined. MFCC is the feature extraction technique can't represent nonlinearity in speech but that possess higher spectrum resolution at low frequency segment. It also provide resemblance with the human's auditory system. That's why, it was the most logical approach to apply. As a quantitative measure, to represent the nonlinearity in speech utterance airflow, Fractal dimension has been applied. Experimental results demonstrated that the applied method was

promising in enhancing speech recognition performance. Most popular and applicable feature vector extraction techniques, like linear predictive Cepstral coefficients (LPCC) and Mel-Frequency Cepstral coefficients (MFCC) are based on the sound channel model and auditory mechanics. The nonlinearity of the ear's perception on frequency can be described by the Cepstral coefficients that can be computed from the Mel-scale frequency domain. As MFCC possess higher spectrum resolution at low frequency domain and works better in noise than LPCC, it has been applied to various experiments and applications. The correctness of recognition obtained for 85.0% for single MFCC feature with sample size of 250 whereas in the case of mixed parameter with MFCC and Fractal dimension, the correctness of speech recognition for same sample size and same speech data were 87.6%.

Rahman *et al.* (2010) said speech recognition has been done in the Aurora-2 speech database for model building by using the HMM and MLPC and MFCC has been applied to extract the auditory features and on the basis of these experiments, a front-end has been developed. The experiments has been exercised in the noisy environment. The clean data set has been applied for training purpose and testing data set has been issued to measure the performance. It was found in the experiment that approximately the same recognition results occurred for both the MLPC and MFCC feature vectors. The average correctness for MLPC was 59.1% and for MFCC was 59.2%. During the experiments, it has also been experienced for Noisy exhibition that MLPC feature vector was more effective and efficient than the MFCC feature vector. It was also experienced that for car noises and babble, MFCC is much better to use. For babble and car noises, the average word accuracy was 51.9% and 56.6% respectively in the case of MFCC. On the other hand, speech recognition using MLPC, for babble noise 48.1% and for car noises 53.8% average word accuracy has been resulted.

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 dependent 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 dependent environment and MFCC as feature extraction technique, speaker independent environment and MFCC as feature extraction technique, speaker dependent 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.

CHAPTER 3

Problem Statement

Background

Punjabi is a tonal language that belongs to Indo-Aryan family of languages. Punjabi is the 10th most widely spoken language around the world. It has more than one hundred and seven million native speakers in India and Pakistan. Apart from India and Pakistan, this language is also spoken in Shri Lanka, Maldives Island, Canada, New Zealand, USA, Australia, Fiji, and Mauritius. In short, it is a widely spoken language. Most of the speakers of Punjabi are native residents of Punjab. Punjabi with its Gurmukhi Script is listed among 22 language with official status in Indian. Spoken Punjabi relies heavily on Sanskrit vocabulary and possesses many dialects such as Majhi, Potwari, Dhani, Hindko, Malwi etc. Majhi is the most popular dialect of Punjabi. This language possess 32 different dialects. In spite of having tonal nature, variation in emotional stress in pronunciation deviates the sense of speech. For extended and long-lasting effects, Sprouting of consonants takes place. Gurmukhi script is written left-to-right and spelled phonetically. There are 25 consonants, 10 vowels, 3 auxiliary signs, and 7 diphthongs in this script. Tonal features are segmental and phonetic in nature.

Punjabi language has gained wide acceptance in media and communication and hence it deserves to get a pace in the growing field of ASR which has been already explored for number of other Indian and foreign languages successfully. Some work has been done in the field of isolated word and connected word speech recognition for Punjabi language.

Statement

Continuous speech recognition is one area where very less work has been done so far. There is a need to work on different modes of Punjabi speech along with their complete set of tonal sounds in terms of ASCII format which is the standard format of digital machines. In this work, Punjabi Speech has been categorized into read mode, lecture mode and conversational mode.

For recognition process, there is a need to perform IPA transcription of recorded speech which then converted to ASCII form as a pre-processing stage.

In general, there is a requirement to work on speech recognition for all three modes of Punjabi speech along with their enhanced set of tonal sound unit.

CHAPTER 4

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 30 and 34 unique phonemes** including silence.

4.1 Requirements for System Implementation

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

4.1.1 List of Models (HMM List)

As a first step, we prepare a list of all those phonemes whose HMM models are to be built, both for 30 and 34 phones (excluding Silence) category. Both the lists has been prepared separately to develop the engine. Figure 4.1 shows the list of 30 different phonemes and Figure 4.2 shows the list of 34 different phones that have been used to build HMM models.

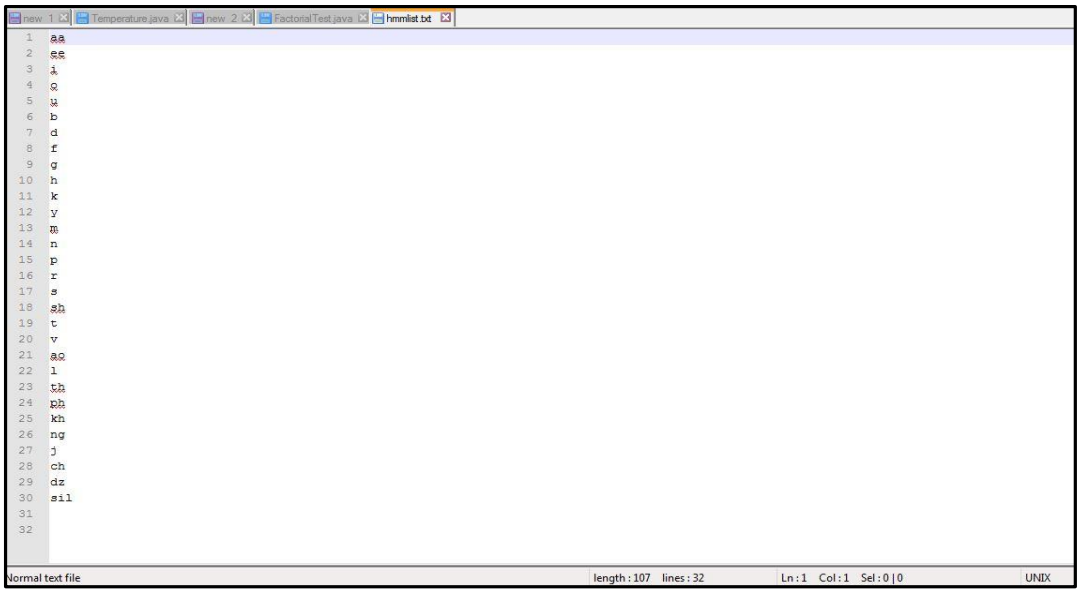


Figure 4.1: HMM list for 30 phones

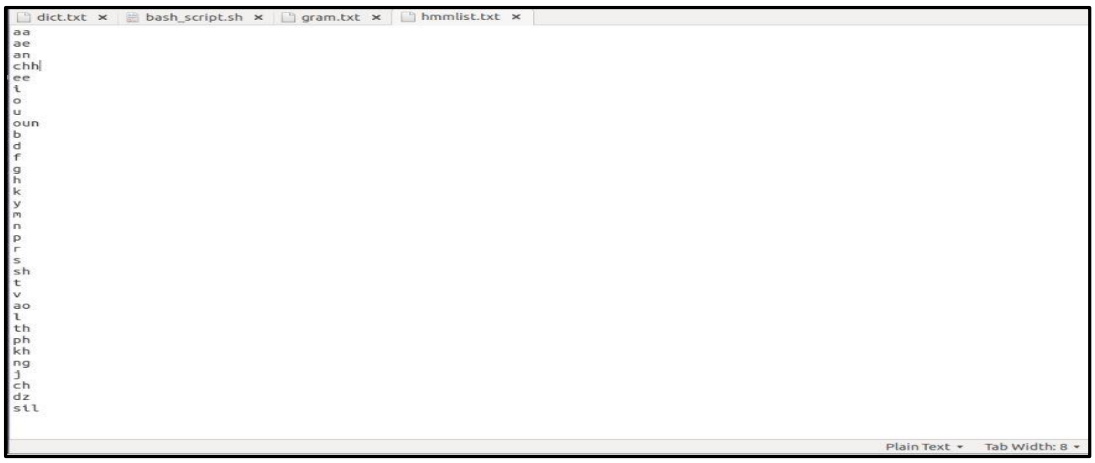


Figure 4.2: HMM list for 34 phones

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.

4.1.2 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:-

CAB	c a b
CALM	c a m

and so on. The pronunciation is not case sensitive. If multiple pronunciations exist for a single word 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 4.3 and Figure 4.4. First column shows the phoneme (and at word level it will be a word) and second column shows the pronunciation of corresponding phoneme. This file will be different for 30 and 34 phone category. Figure 4.3 represents the Pronunciation Dictionary for 30 phones whereas Figure 4.4 represents the Pronunciation Dictionary for 34 phones.

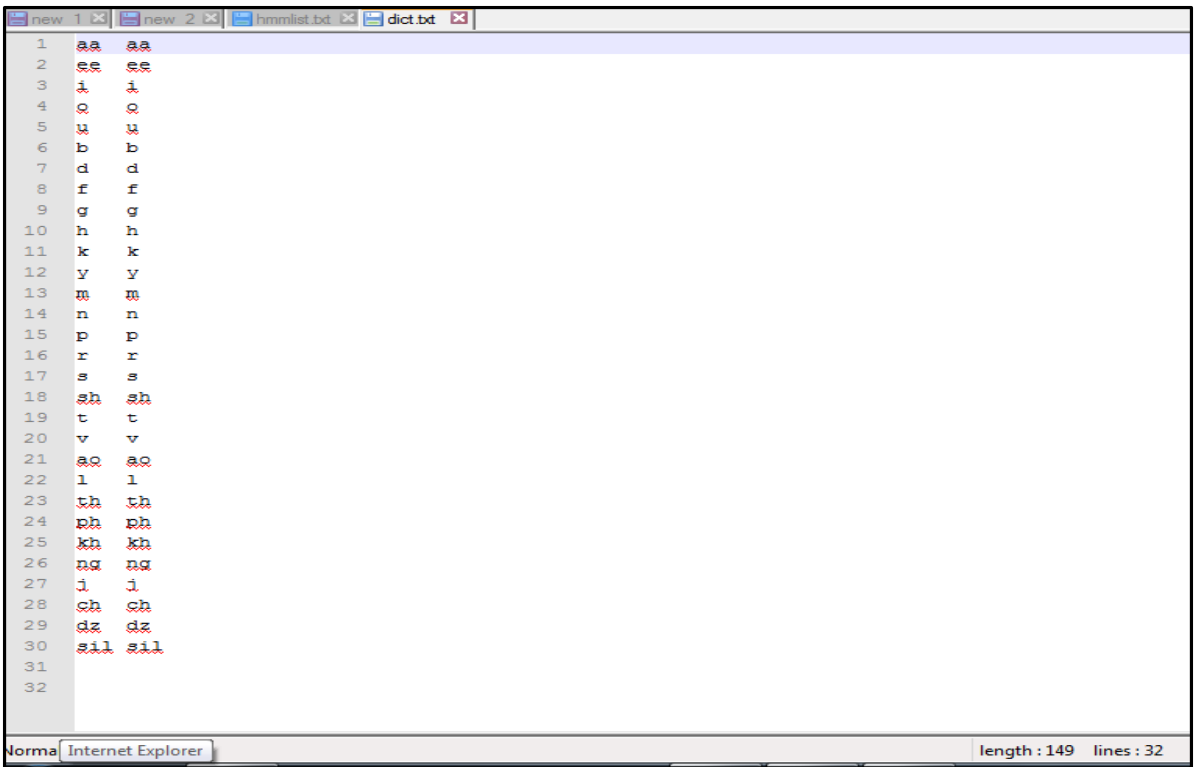


Figure 4.3: Pronunciation dictionary for 30 phones

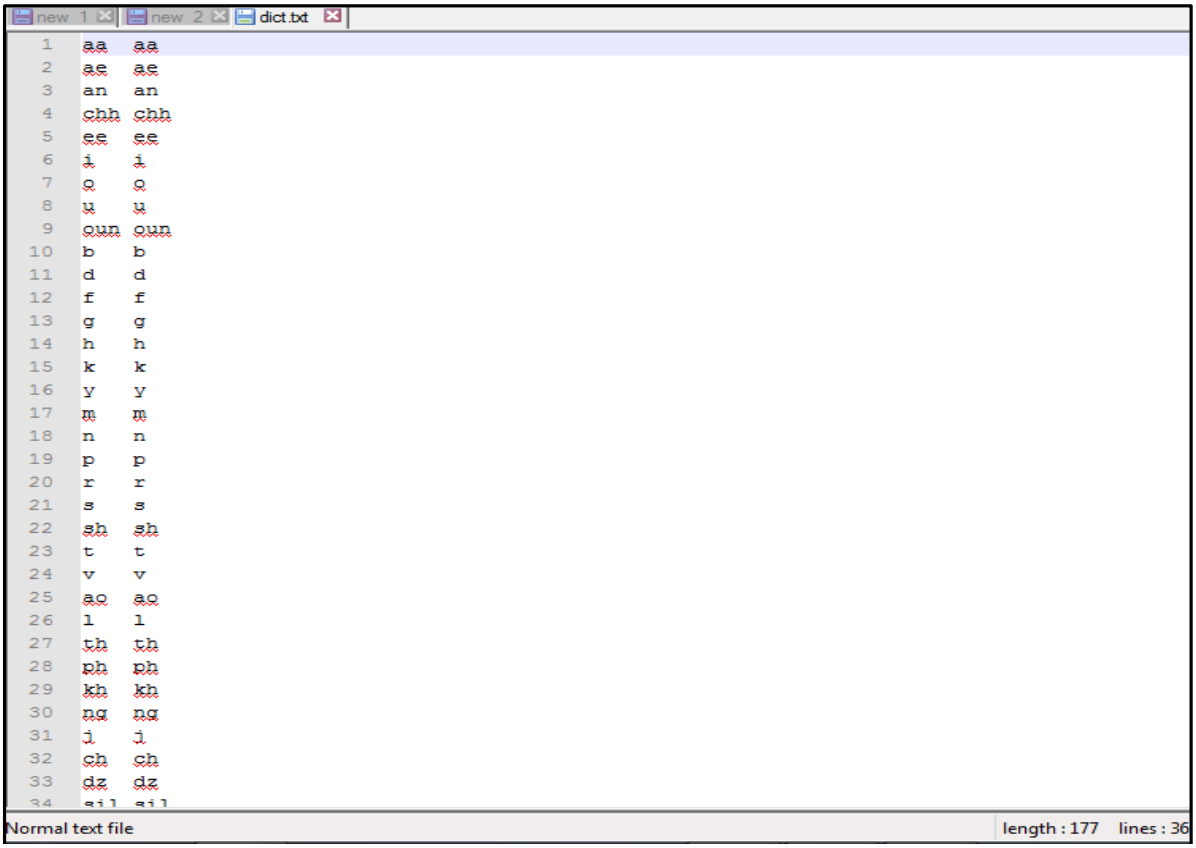


Figure 4.4: Pronunciation dictionary for 34 phones

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.

4.1.3 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 4.5 shows the working of HParse tool. This tool takes as input the 'grammar' and using this input it creates a word network.

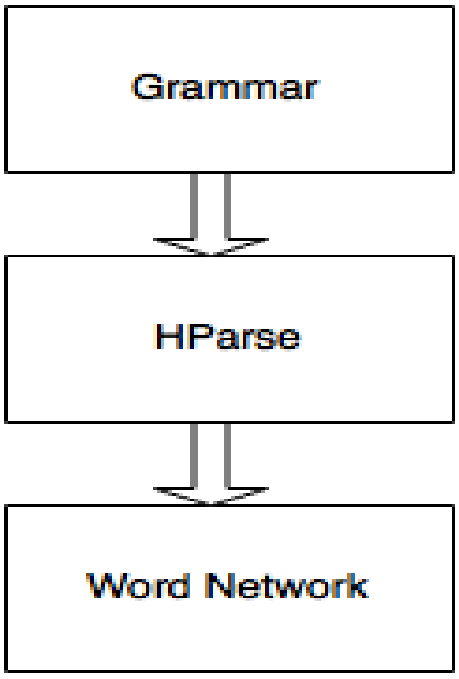


Figure 4.5: Working of HParse

The Grammar that has been used in this work is shown in Figure 4.6 is a form of regular expression as it is regular grammar according to Chomsky Hierarchy.

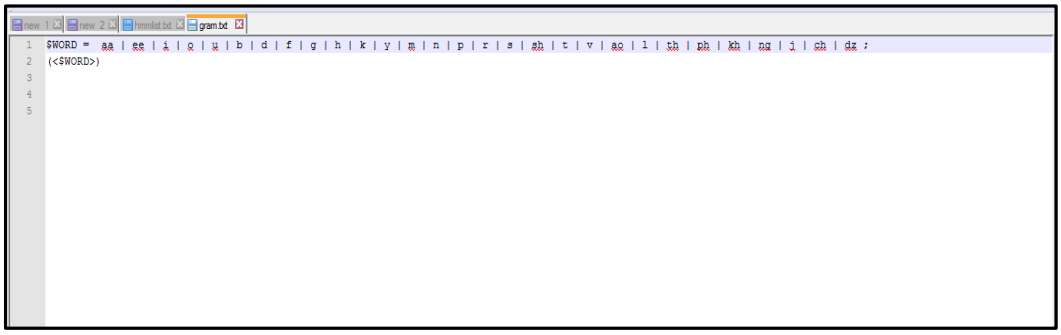


Figure 4.6: Grammar for 30 phone category engine

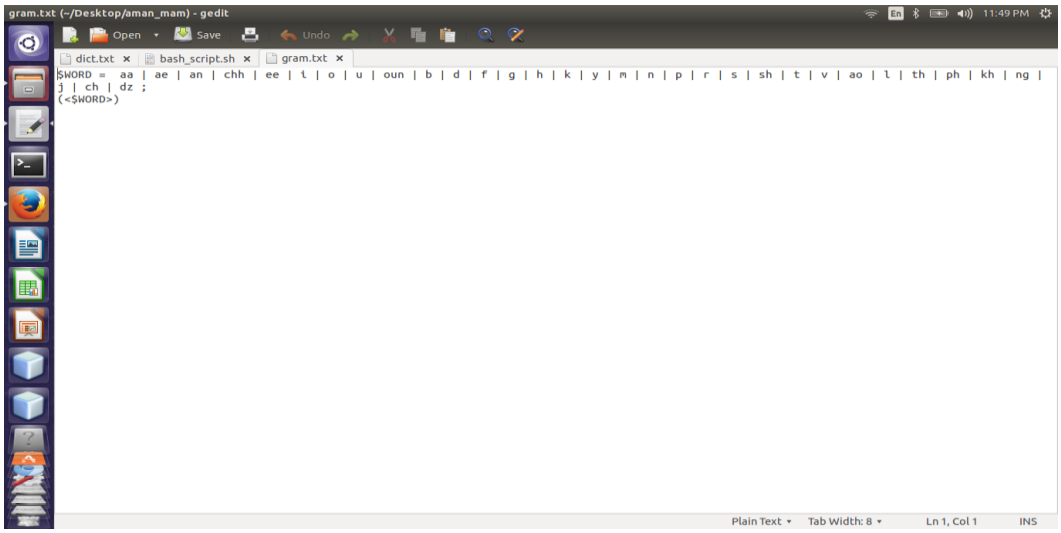


Figure 4.7: Grammar for 34 phone category engine

Where, vertical bars specify alternatives and angle braces specifies one or more repetitions. This complete grammar is converted by HTK into a network. Figure 4.6 displays the grammar for engine to be built with 30 phones (including silence) and Figure 4.7 displays the grammar for engine with 34 phones (including silence).

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.

4.1.4 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. These transcriptions demonstrates 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. This file does not differ for developing engine with 30 and 34 phone category. Transcription file that has been used in this work is given in Figure 4.8.

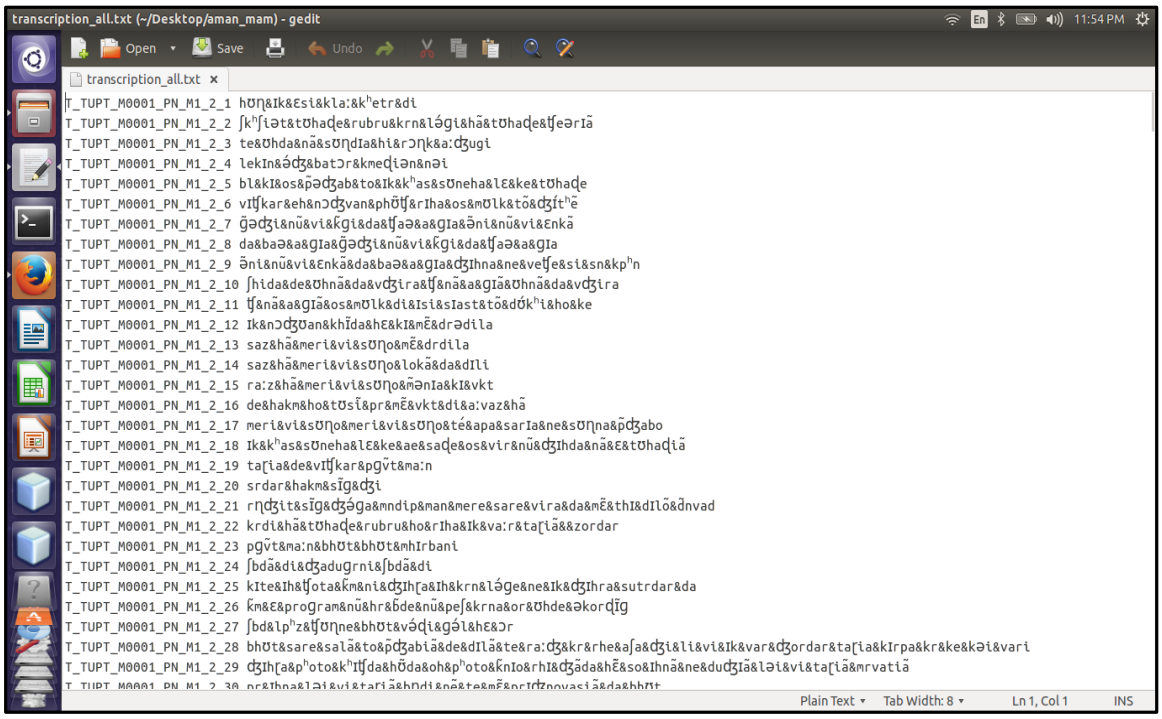


Figure 4.8: Transcription_all file

In Figure 4.8, first column shows speech audio file names and second column shows their corresponding transcribed spoken utterances.

Figure 4.9 shows a spreadsheet which consists the list of phonetic alphabets and their corresponding ASCII characters.

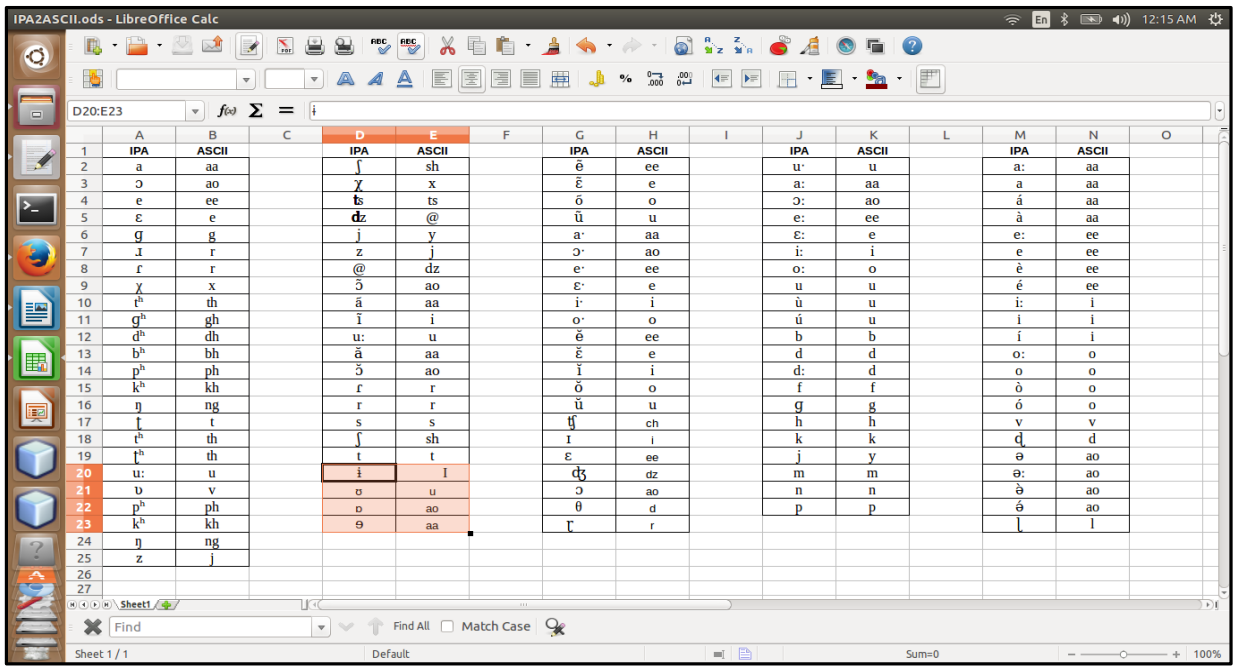


Figure 4.9: IPA to ASCII

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.

To develop the phonetic engine with 34 phones, 4 new phones has been introduced in comparison to phonetic engine with 30 phones. These phones are ‘ae’, ‘an’, ‘chh’, ‘oun’.

To develop a phonetic engine for Punjabi language, there is a need to identify all the phones which is present in the language. But due to its variations and less work done in this sub-domain, there is no standard list of phones is available. This work is a set of experiments as well as contribution towards the development of speech recognition engine for Punjabi language.

4.1.5 Training and Testing Files

In this step, there is a need to prepare two master label files (MLFs), one for training purpose and another for testing purpose at phoneme level such that a single line must consists of single phoneme. For a same transcription file, MLF file will be different for phonetic engine with 30 category and 34 category, respectively. Format of both files should remain same for both the categories. 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 4.10 shows the format of both files that have been used in this work.

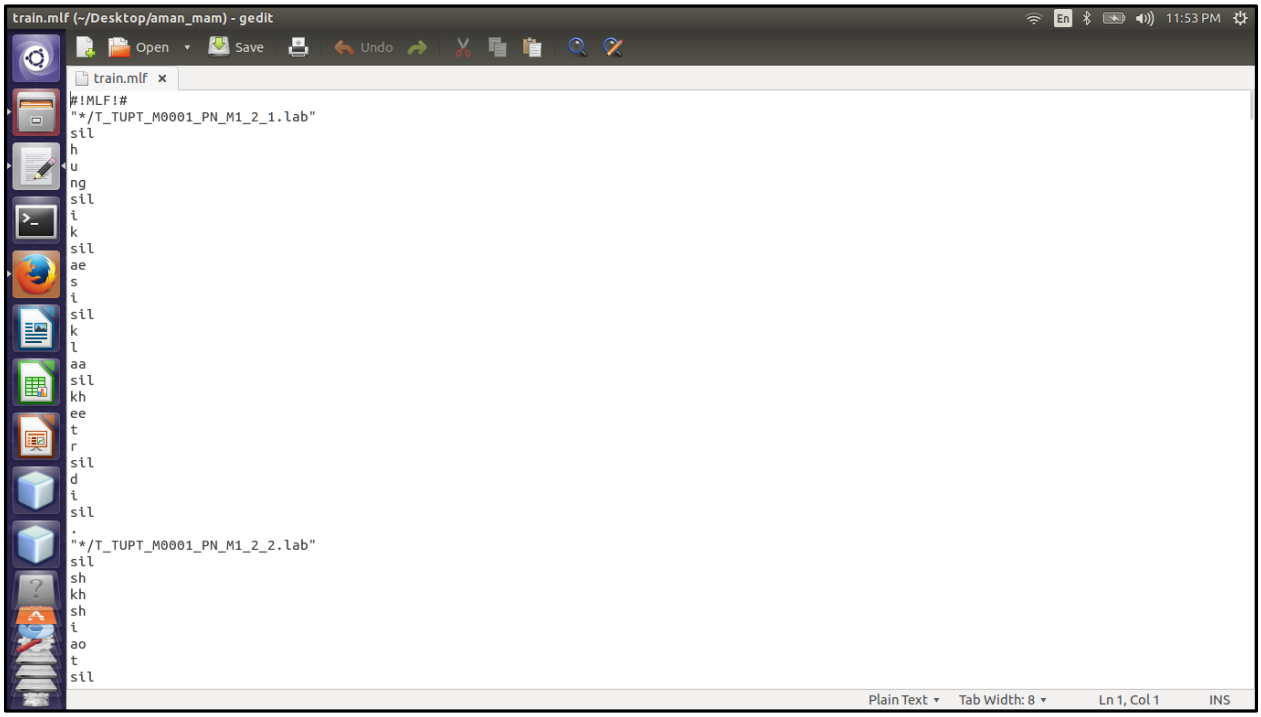


Figure 4.10: Training and testing MLF file

4.1.6 Feature Extraction

A sequence of feature vectors has been 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.

4.1.7 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 on-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 4.11.

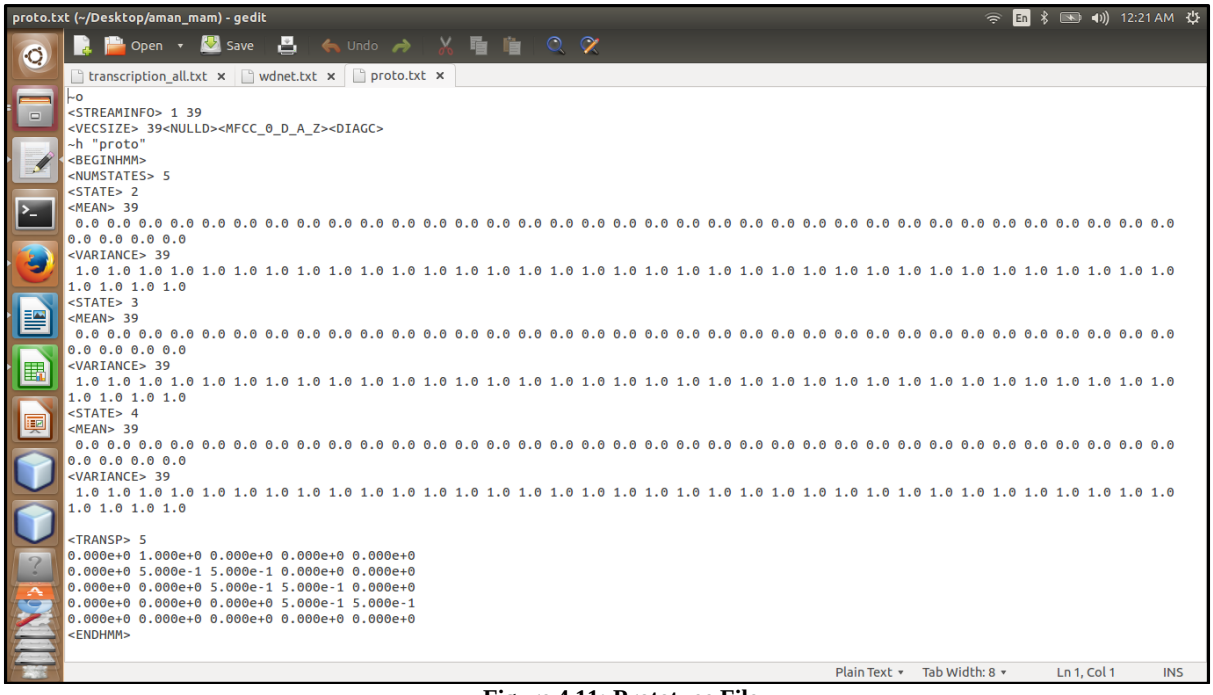


Figure 4.11: Prototype File

This file is same for the 30 phone based engine as well as 34 phone based engine.

4.2 Training and Testing of the System

The general framework of phonetic engine is shown in Figure 4.12.

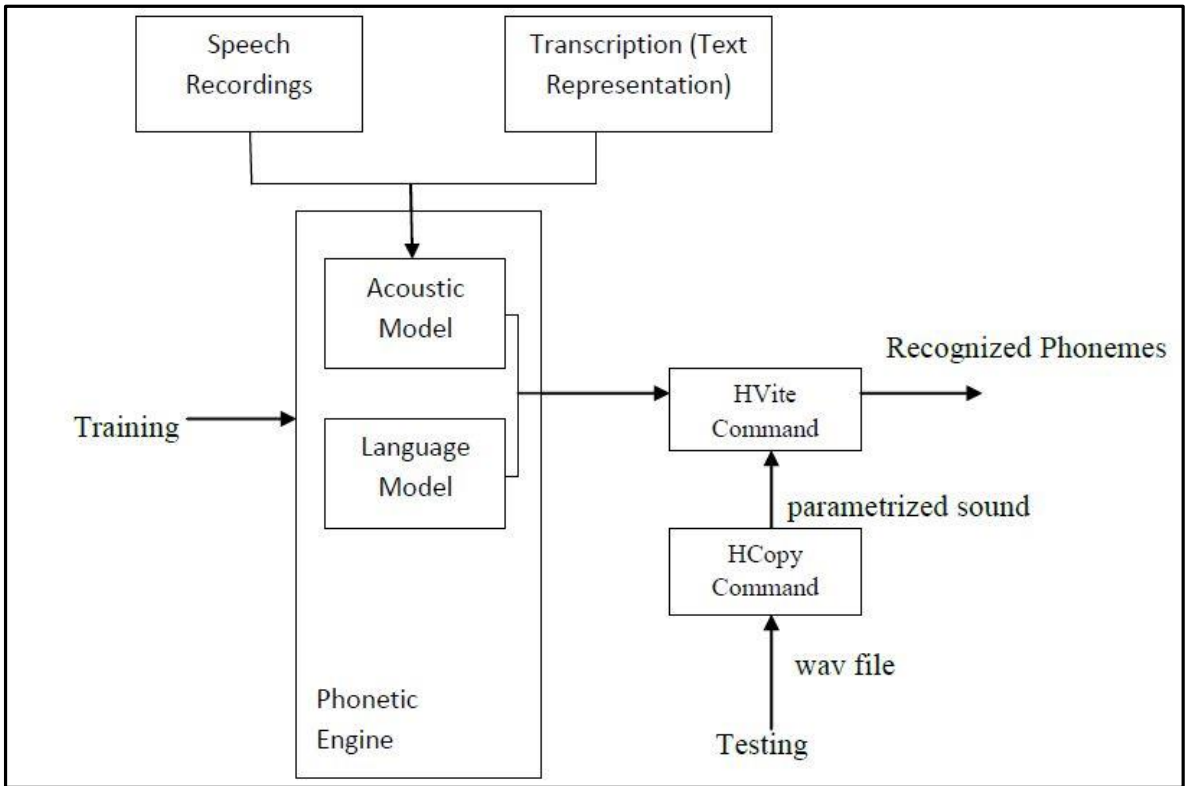


Figure 4.12: Skeleton of Phonetic Engine

4.2.1 Training

The process of estimating the parameters of HMM from the examples of data sequences called training. This process is the soul of developing phonetic engine. For providing the training to HMMs for each phone, many examples of each phoneme has been considered as a training data. It was taken care to have at least 3 example of each phoneme should be there in training data. More examples, more will be the accuracy of system at the time of recognition.

Training process has been done separately for 30 phones based Phonetic Engine (PE) and 34 phones based PE.

4.2.1.1 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).

4.2.1.2 Training Algorithm

Training algorithm is different in terms of mixtures that has been used for training. To develop the PE with 30 phones (excluding 'sil'), 32 mixtures have been used whereas to develop the PE with 34 phones (excluding 'sil'), 36 mixtures have been used.

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.

4.2.2 Testing

After providing the training, the performance of trained system can be checked using the HTK analysis tool 'HResults'. This process is also different for 30 phones and 34 phones based PE, respectively.

4.2.2.1 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 list file consist path of MFCC features of wav files (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).

4.2.2.2 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 5

Three Modes of Data Collection and Results

Speech is a natural and very easy way of exchanging information. If used as a medium to interact with the computer, it can solve various problems. For this, some speech interfaces such as speech synthesizer and speech recognizer are required. Speech recognition and speech synthesis both require phonetic transcription. 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.

5.1 Database collection and transcription

Punjabi speech data has been collected in three different modes of speech, namely, read mode, lecture mode and conversational mode speech.

i) Read Mode Speech:

For this mode of speech, data has been collected from 4 native Punjabi speakers for a duration of 3 hours approximately. Recording has been done in two different kind of environments. The first kind of recording has been done in Studio environment where microphone channel has been used as an apparatus to record by maintaining the sampling frequency 48 KHz and 16 bits per sample. The other kind of recording was done in a natural room environment by using the same apparatus, a microphone channel by maintaining the Sampling frequency at 22050 Hz. Punjabi speakers recited the passages and sections from Punjabi books.

ii) Lecture Mode Speech:

The data for the lecture mode has been taken from the radio channel, *Punjabi Radio USA*. This data is available in public domain. The recording of this mode of data has also been done with a sampling frequency of 48 KHz and a bit rate of 16 bits per sample. Approximately, 2 hours of data has been collected for training and testing.

The difference between read speech mode and lecture mode speech is that, in read mode, speech has recited by the speakers from books whereas in case of lecture mode, speech was collected in the natural form of speech, just like a person deliver the lecture.

iii) Conversational Mode Speech:

Speech data for this mode of speech has been recorded by native Punjabi speakers. This mode contains the conversation and dialogues of the speakers in their natural way of discussion. The data has been recorded into normal room environment as well as open room environment by using a microphone channel that was calibrated at a frequency of 48 KHz. Approximately, 20 minutes duration of data has been considered for training and testing purpose.

5.2 Development of phonetic engine

As the development of Phonetic Engine is prosodically guided. So the work includes various parts of prosody like Annotation or Transcription, Syllabification, Pitch Accent Marking, Break Index Marking etc. The work has been focused on annotation/transcription. Below mentioned sub-sections represents the details of prosody labeling.

5.2.1 Transcription/Annotation

Transcription of read speech, lecture speech, and conversational speech has been done using International Phonetic Alphabet (IPA) chart, which is available at *http://westonruter.github.io/ipa-chart/keyboard/*. There are 36 IPA symbols including Vowels, Semi vowels, and Consonants. Apart from this, diacritics, tone and word accents, and suprasegmentals were also used in transcription. Consonants include Steps, Velar, Affricates, Nasals, Laterals, and Fricatives.

Figure 5.1 contains the transcription of a lecture mode speech. After selecting a segment, its transcription has been noted down in the transcription pane using IPA chart and WaveSurfer, a standalone tool.

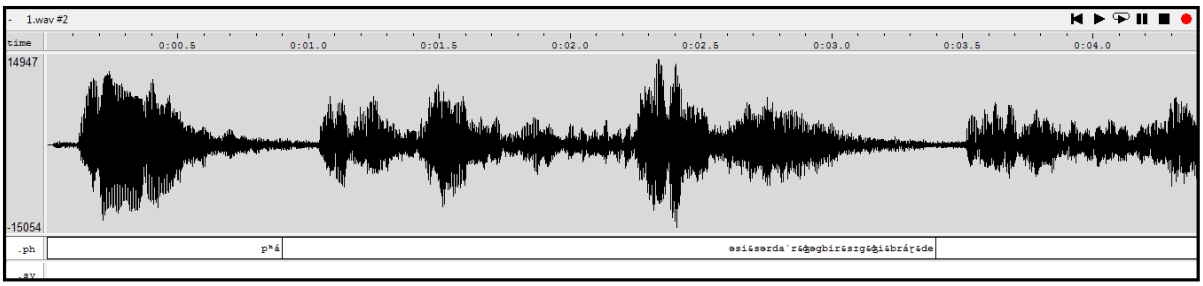


Figure 5.1: Transcription of a lecture speech file

5.2.2 Break Index Marking

Break index marking has been done to mark the break indexes and silence removal. For doing this, each wave file has been divided into overlapping frames, then energy level of each frame is computed, if it is less than 1.2 dB then it is detected as silence. Detected silence is then removed from the wave file. Figure 5.2 contains the snapshot of a file containing time stamping of the break indices and Figure 5.3 contains these markings on a wave file.

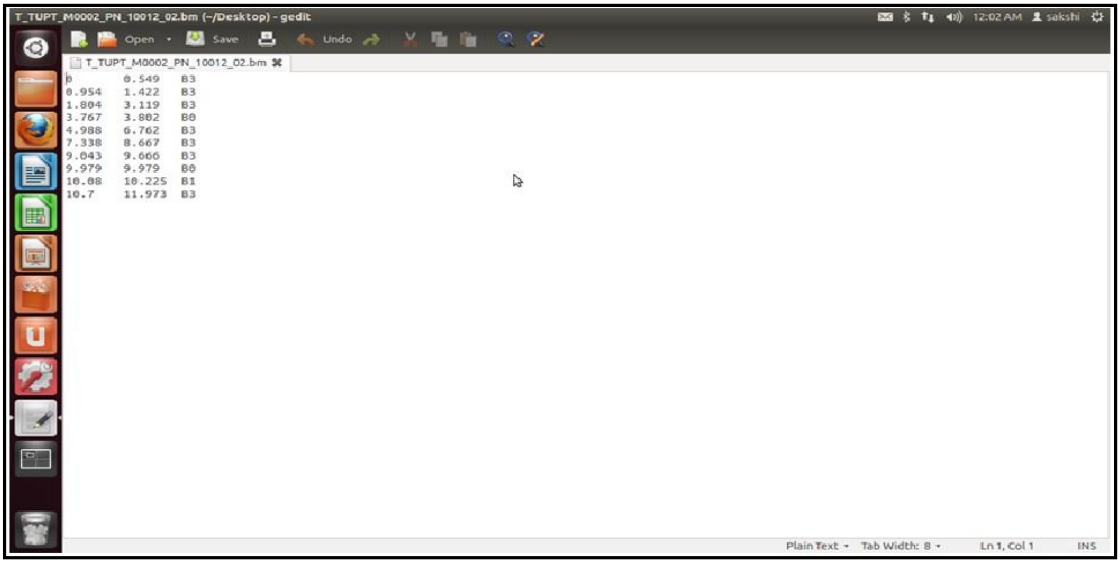


Figure 5.2: Time stamping of break index markings

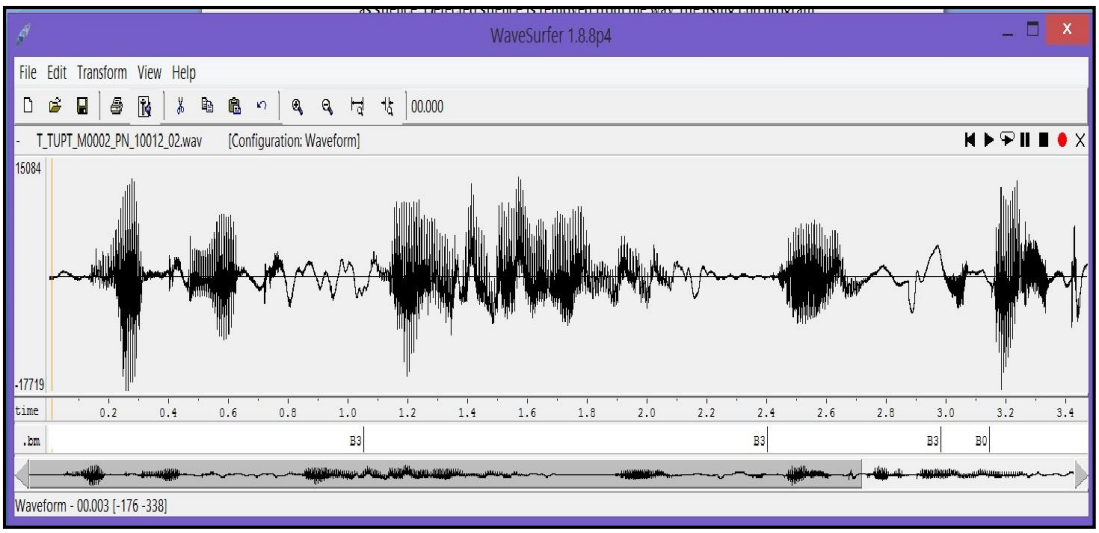


Figure 5.3: Break index markings corresponding to wave signal

5.2.3 Pitch Accent Marking

This work has been carried out with the objective to mark the various pitch labels on the wave file segment. Firstly, in the whole speech, voiced and unvoiced regions are detected so that only the voiced regions are pitch marked. In a particular voiced segment of speech, pitch accent may have 7 different marks, namely, low to high (*LH*), high to low (*HL*), flat (*F*), *i.e.*, no change in pitch, very low to high (*VLH*), very high to low (*VHL*), low to very high (*LVH*) and high to very low (*HVL*). Zero frequency filtering technique is used to segment the speech into voiced and unvoiced region. Then, pitch marking is done for each voiced region. For pitch marking, sampling rate of the signal should not be more than 8000 Hz.

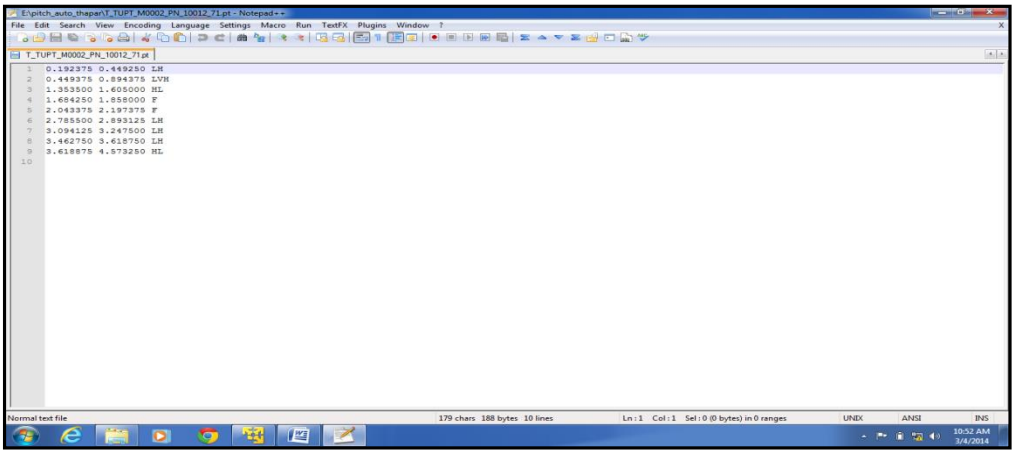


Figure 5.4: System generated pitch markings

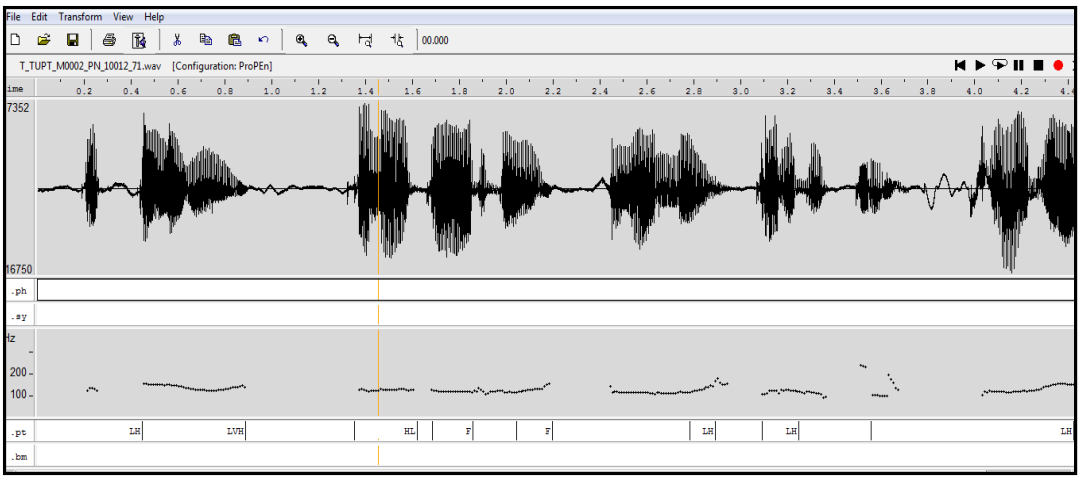


Figure 5.5: System generated pitch markings corresponding to wave signal

5.3 Performance Evaluation of different modes of speech

PE has been trained for all the modes of speech: Read mode speech, Lecture mode speech and Conversational mode speech, as mentioned earlier. The total duration of each kind of data considered for PE automation is given in Table 5.1.

Table 5.1: Total duration of each Mode of Speech

Mode	Total Duration(in Minutes)
Read Speech Mode	186.34
Lecture Speech Mode	124.41
Conversational Speech Mode	20.22

Phonetic Engine has been trained and tested for all three speech modes: read speech mode, lecture mode, and conversational mode. Different test cases have been generated for different modes of speech depending upon the nature of speech data collected. For read speech mode, the PE has been trained and tested for each gender and for each speaker. Total speech data set for each speech mode has been divided into two parts training data set and testing data set. For training purpose, 75.0% of total speech data has been reserved for training and rest 25.0% of data has been used for testing purpose. After recognition process, a confusion matrix has been generated as a result. This matrix is the resultant of comparison of expected results of transcription with the actual results of transcription. As there were various categories has been created for each kind of speech, therefore for each case, confusion matrix has been generated

separately. This confusion matrix 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 is 30 as the total number of unique phonemes including silence we have used are 30 and total number of columns are 30 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;

$$Accuracy = (N - S - D - I) / N$$

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

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

$$\% \text{ Correctness} = (H / N) * 100$$

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

5.3.1 Performance Evaluation for Read mode of Speech

Following are the results for the testing accuracy of the PE developed in this work. Figure 5.6 shows the confusion matrix generated after testing the PE in Read speech with 30 phones.

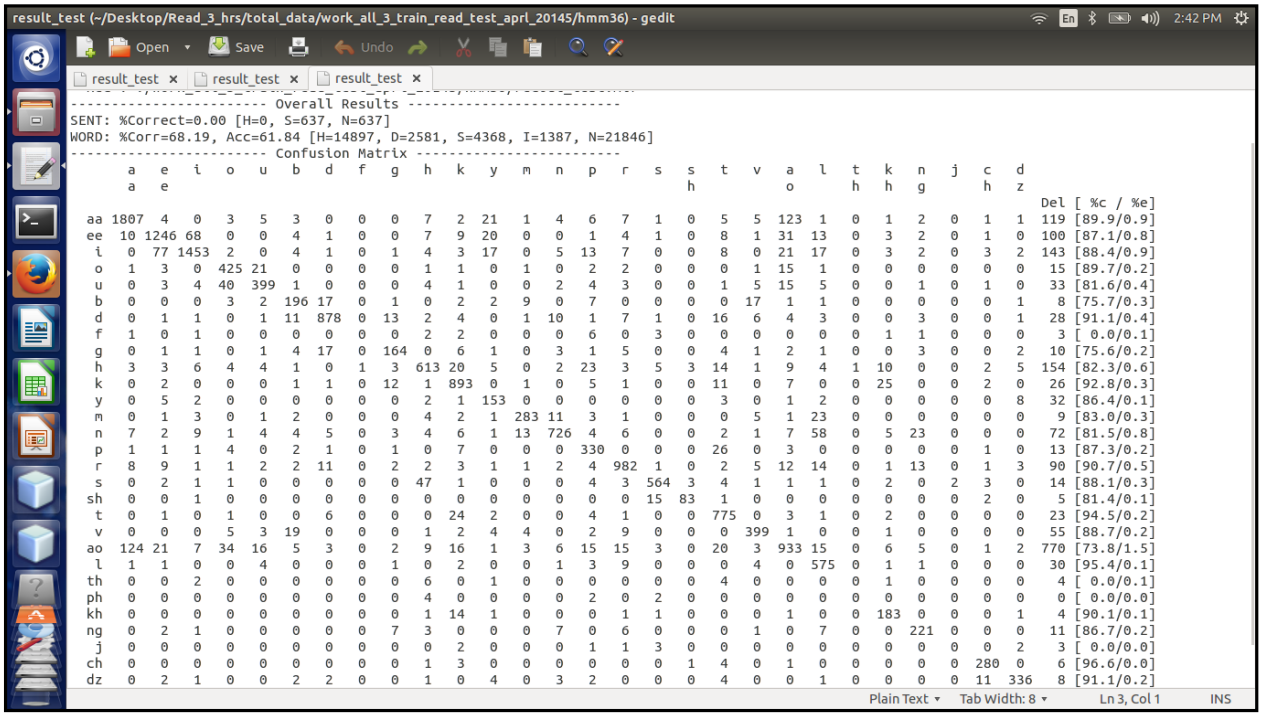


Figure 5.6: Confusion matrix of phonetic engine trained for Read speech mode with 30 phones

In Figure 5.6, each cell represents the 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 file. 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 68.2% and accuracy percentage is 61.8%. Total number of phonemes substituted (*S*) are 4368, total number of phonemes deleted (*D*) are 2581, number of phones inserted (*I*) are 1387, total number of phonemes correctly recognized (*H*) are 14897 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*).

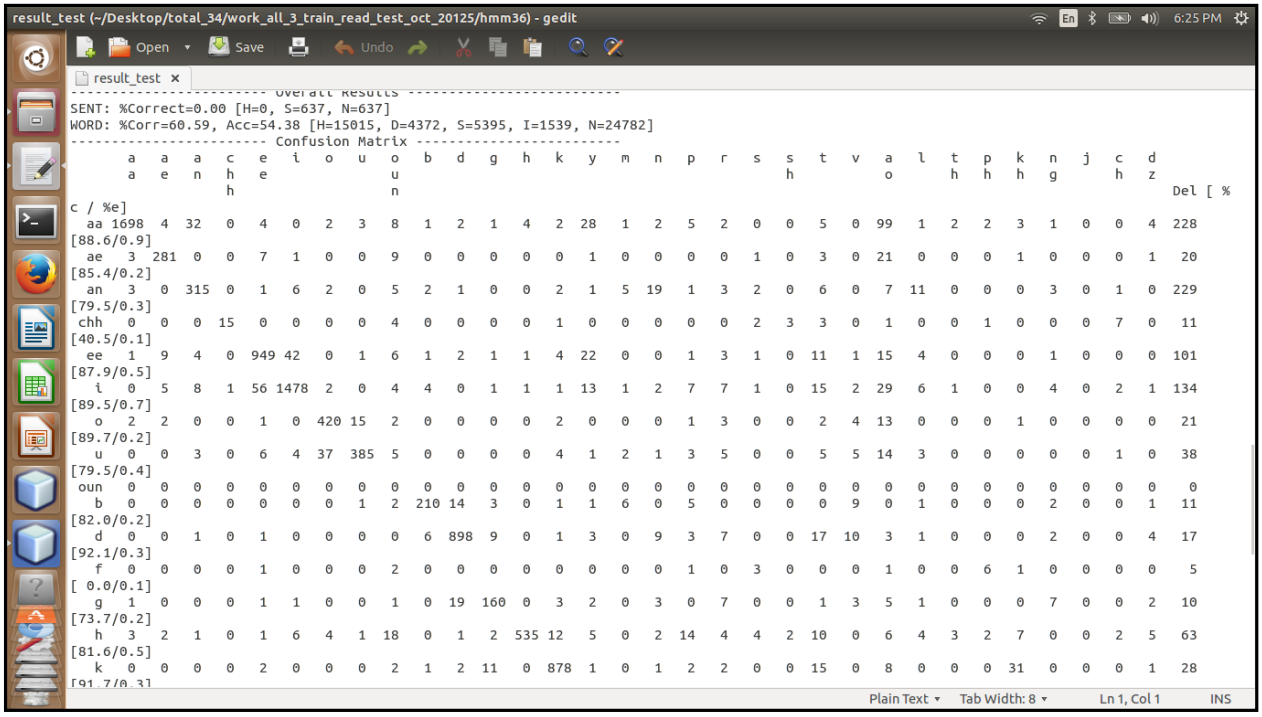


Figure 5.7: Confusion matrix of phonetic engine trained for read speech with 34 phones

First line shows the sentence level accuracy based on total number of label files which are identical to transcription file. It is 0.0% in this mode. The second line shows the word accuracy based on matches between label files and transcriptions. In this mode, *i.e.*, read speech mode with 34 phones, correctness percentage is 60.6% and accuracy percentage is 54.4%. Total number of phonemes

substituted (*S*) are 5395, total number of phonemes deleted (*D*) are 4372, number of phones inserted (*I*) are 1539, total number of phonemes correctly recognized (*H*) are 15015 and total number of phonemes (*N*) are 24782.

Read Speech Data has also been trained for each gender. This process has been exercised for PE both for, 30 phones based as well as 34 phones based. Figure 5.8 and 5.9 shows the confusion matrix for total males of read speech mode with 30 phones and 34 phones, respectively.

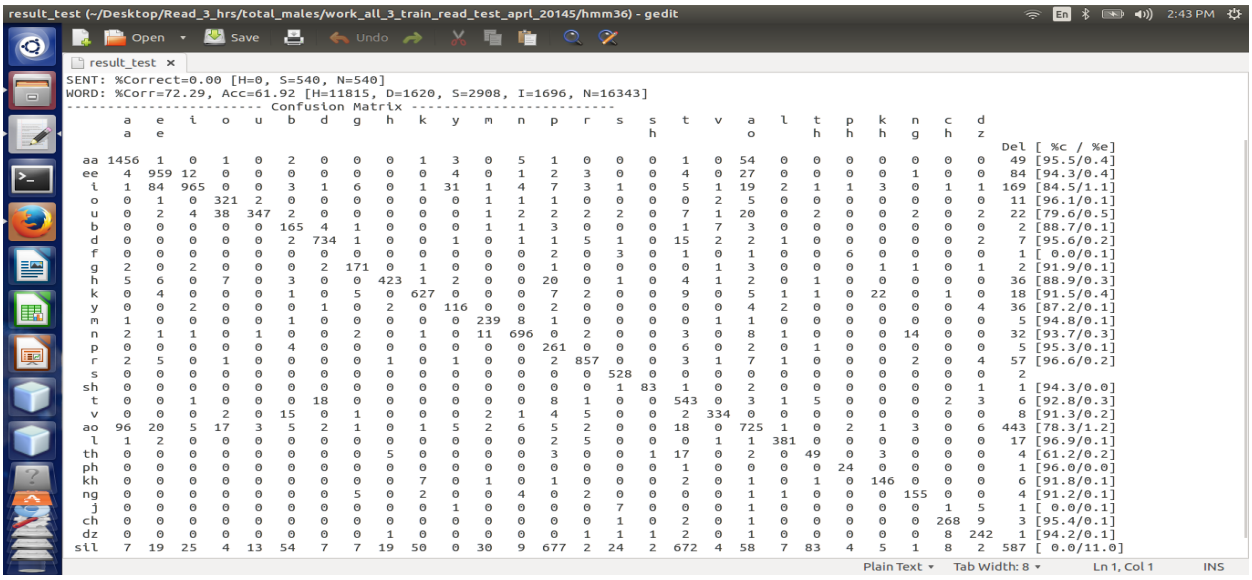


Figure 5.8: Confusion matrix for total males of Read Speech Mode with 30 Phones

Figure 5.8 shows the confusion matrix which is clearly mentioning the accuracy of PE for total males is 61.9% and correctness is 72.3%. The total submitted phonemes correctly recognized (*H)* are 11815, deleted phones *(D)* are 1620, inserted phones *(I)* are 1696 and substituted (*S)* are 3008, and the total number of phonemes, processed *(N)* are 16343.

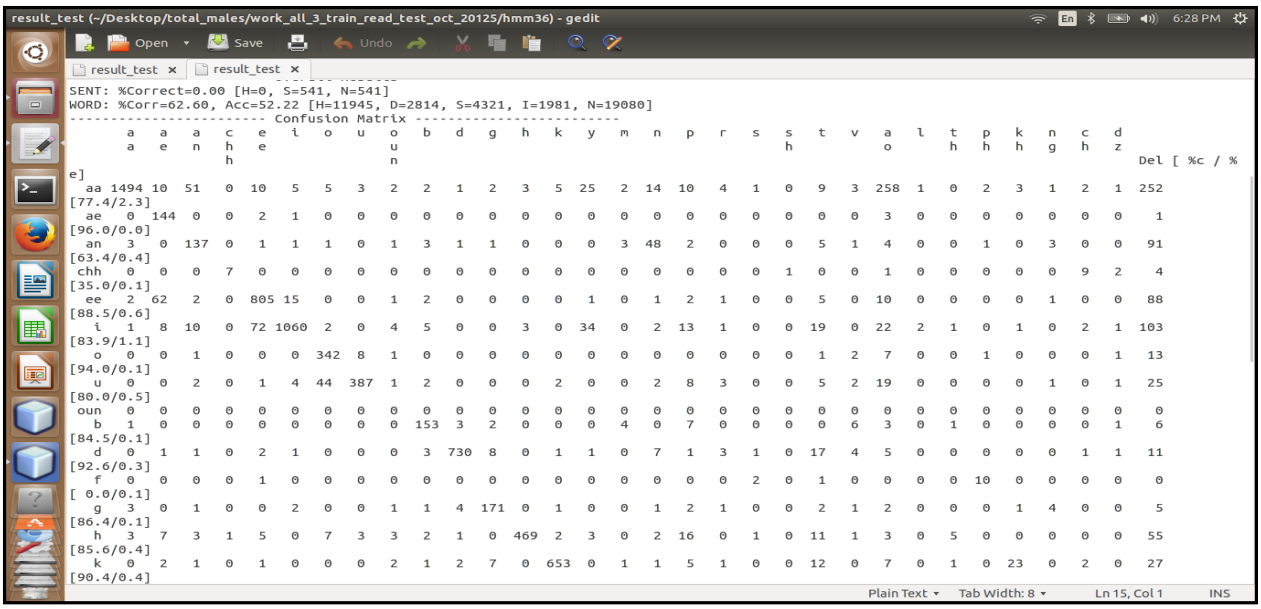


Figure 5.9: Confusion matrix for total males of read speech with 34 phones

Figure 5.9 shows the confusion matrix that PE has obtained 52.2% of accuracy and 62.6% of correctness for total males of read speech with 34 phones including silence. The total submitted phonemes correctly recognized *(H)* are 11945, deleted phones *(D)* are 2814, inserted phones *(I)* are 1981 and substituted *(S)* are 4321, and the total number of phonemes, processed *(N)* are 19880. Figure 5.10 and 5.11 shows the confusion matrix for PE that is trained and tested for all female speakers of read speech with 30 phones and 34 phones (including silence), respectively.

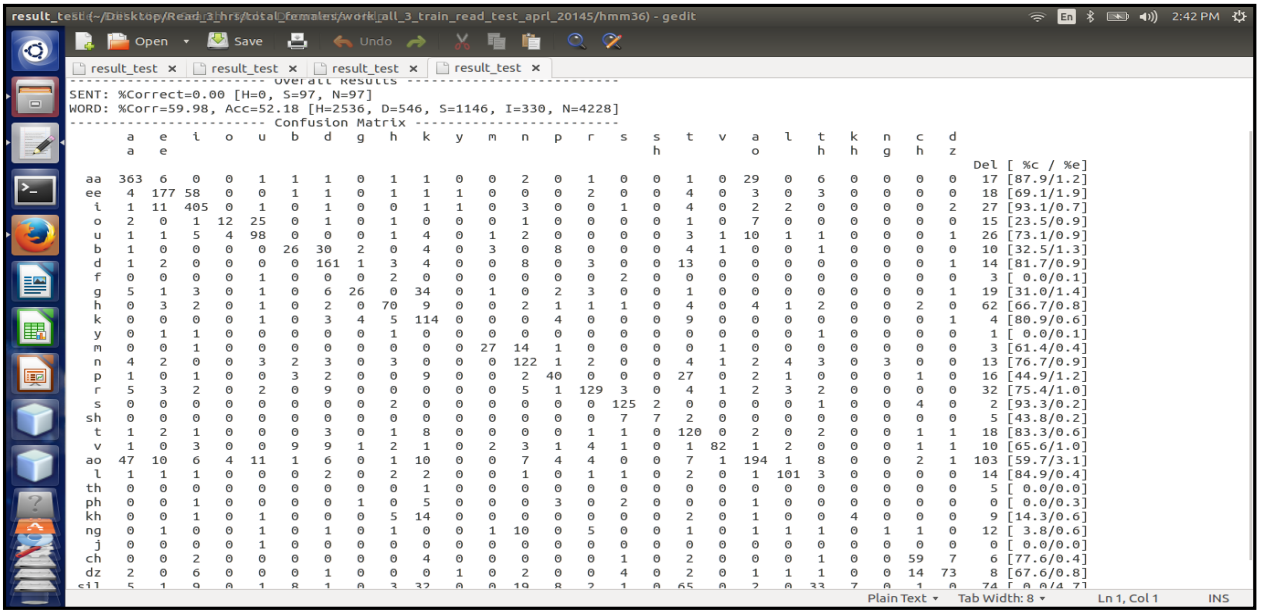


Figure 5.10: Confusion matrix for PE, trained for total females of read speech with 30 phones

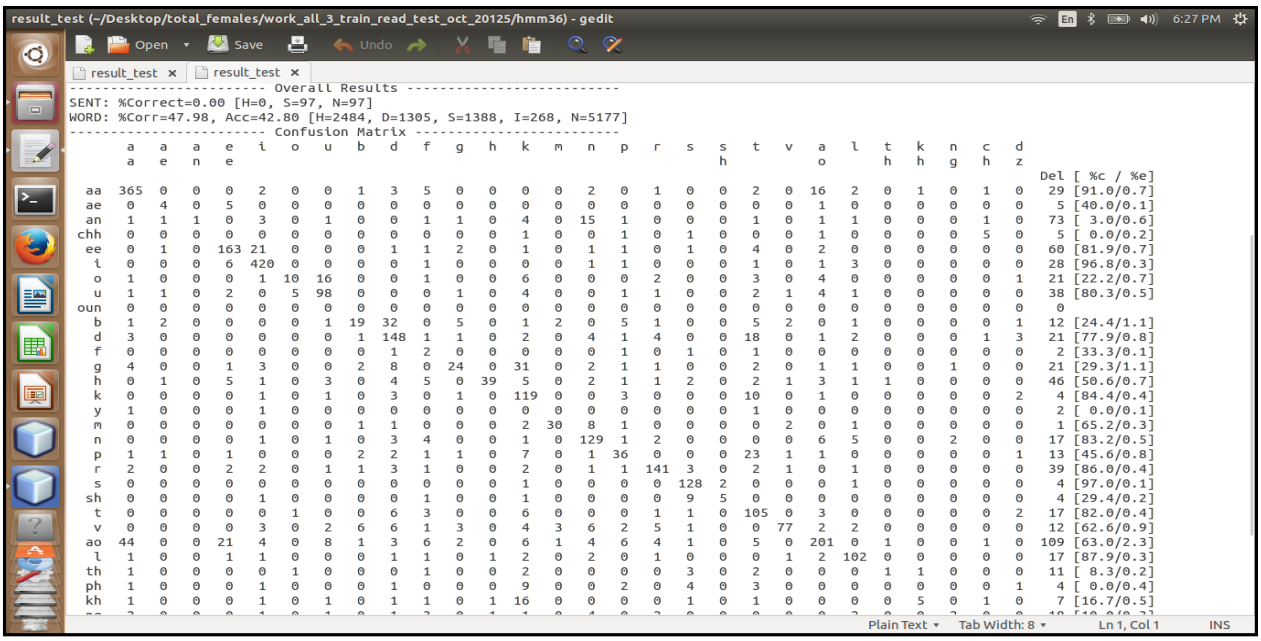


Figure 5.11: Confusion matrix for PE trained for total females of read speech with 34 phones

Figure 5.10 shows the confusion matrix that depicts the correctness as well as accuracy of PE for total females in read speech mode with 30 phones (including silence). The accuracy of the system is 52.2% and correctness is 60.0%. Figure 5.11 shows that the confusion matrix of Phonetic Engine for total females with 34 phones. The accuracy of the system is 42.8% and correctness is 48.0%. Table 5.2 clearly displays the details of experiments, carried out with read speech data. The PE has been trained with total data as well as for both the genders, male and female. This experiment has been done for 30 phonemes (including silence) as well as for 34 phonemes (including silence).

Table 5.2: Testing accuracy of PE with 30 phonemes (including silence) for each gender

Read Speech Data	No. of Speakers	Training Data (in Minutes)	Testing Data (in Minutes)	Accuracy (in %)	Correctness (in %)
Total Data	4	138.20	48.13	61.8	68.2
Total Female	2	18.48	6.35	52.2	60.0
Total Male	2	122.41	38.65	61.9	72.3

Table 5.3: Testing accuracy of PE with 34 phonemes (including silence) for each gender

Read Speech Data	No. of Speakers	Training Data (in Minutes)	Testing Data (in Minutes)	Accuracy (in %)	Correctness (in %)
Total Data	4	138.20	48.13	54.4	60.6

Total Female	2	18.48	6.35	42.8	48.0
Total Male	2	122.41	38.65	52.2	62.6

PE has also been trained for each individual person, for read speech data. This process has also been carried out with 30 phones as well as 34 phones. Figure 5.12 and 5.13 displays the correctness % and accuracy % of the phonetic engine, trained for individual person of female gender. As previous process, total duration of female data divided into each female speaker first. Then 75.0% of data of each individual's speech duration has been used for training purpose and rest 25.0% of data has been used for testing purpose.

Figure 5.12 clearly displays that PE trained for female_01 for read speech data with 30 phones obtained the 48.8% of accuracy and 55.7% of correctness. Figure 5.13 displays that PE trained for female_02 for read speech data with 30 phones obtained the 64.7% of correctness and 56.8% of correctness.

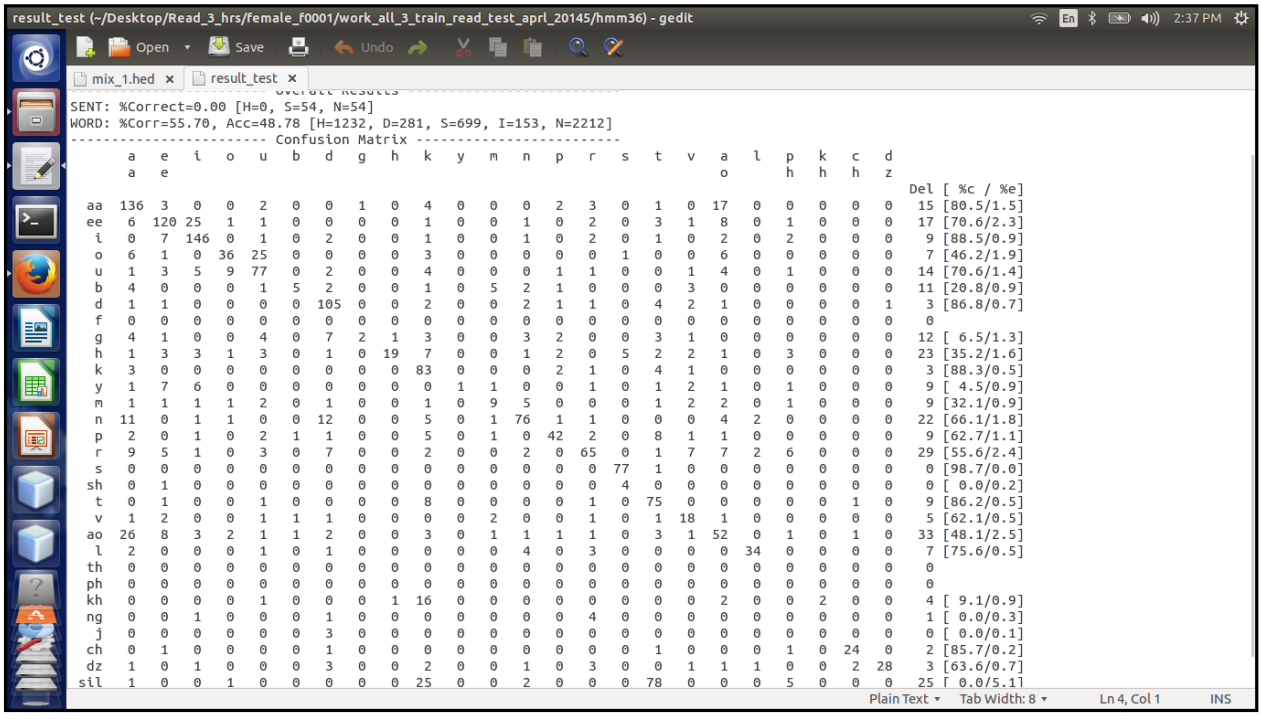


Figure 5.12: Confusion matrix for PE, trained for female_01 of read speech with 30 phones

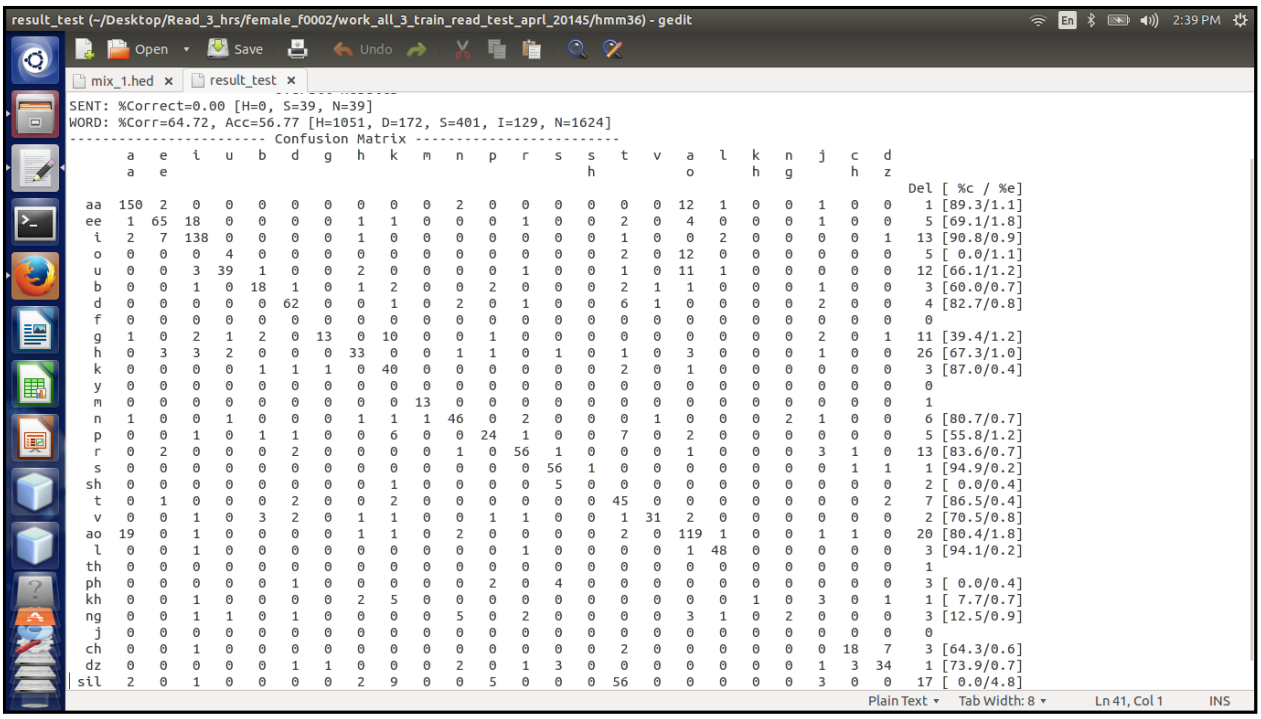


Figure 5.13: Confusion matrix for PE, trained for female_02 - read speech with 30 phones

Table 5.4 presents the testing accuracy and correctness of PE for each individual female speaker of read speech.

Table 5.4: Testing accuracy of PE with 30 phonemes (including silence) for each female individual

Read Speech Data	Training Data (in Minutes)	Testing Data (in Minutes)	Accuracy (in %)	Correctness (in %)
Female_01	10.35	3.45	48.8	55.7
Female_02	7.47	2.4	56.8	64.7

PE has also been trained for speech data of both female speakers of read speech with 34 phonemes. Figure 5.14 and Figure 5.15 displays the confusion matrix for both of the speakers, respectively.

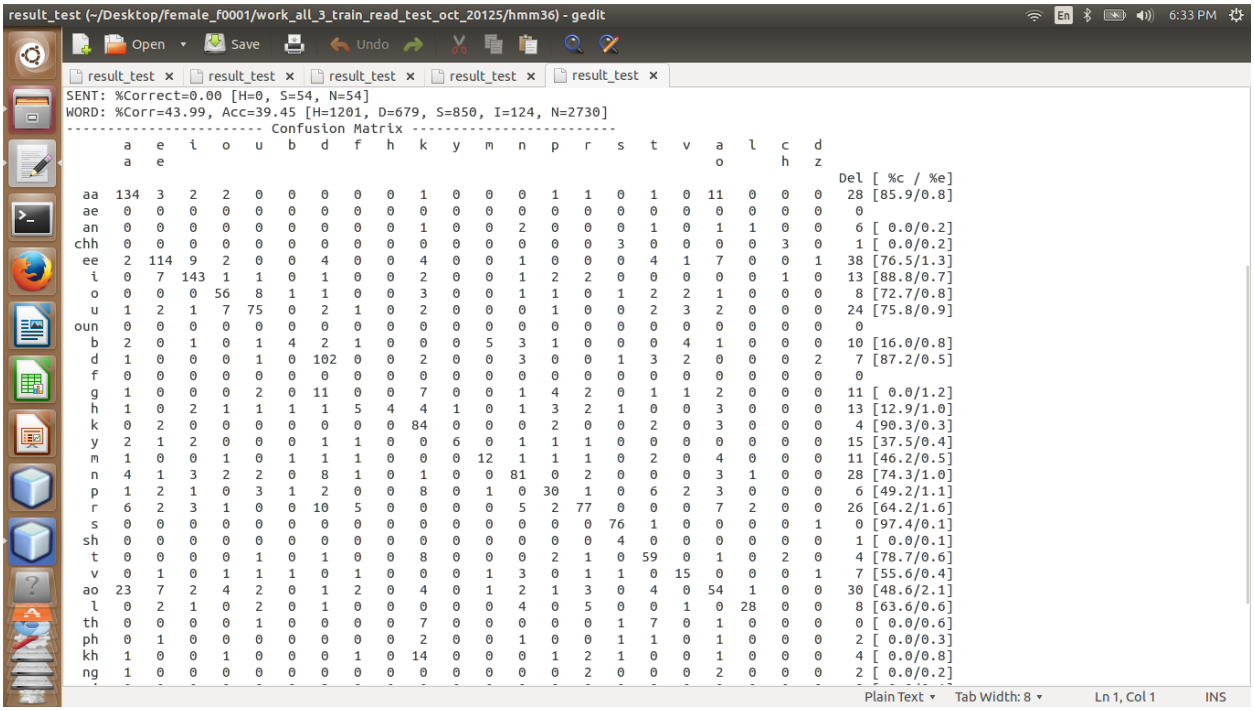


Figure 5.14: Confusion matrix for PE, trained for female_01 of read speech with 34 phones

Figure 5.14 dictates that accuracy and correctness of PE trained and tested for female_01 speaker of read speech with 34 phones is 39.4% and 43.9% respectively.

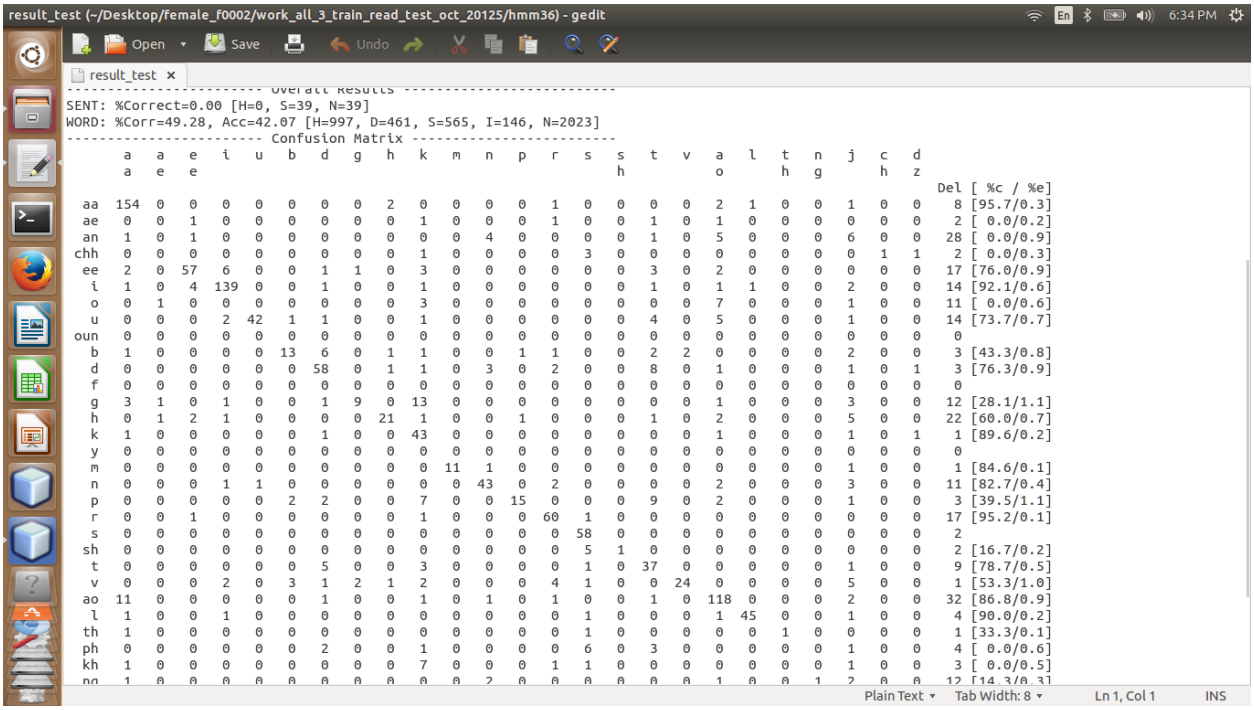


Figure 5.15: Confusion matrix for PE trained for female_02 of read speech with 34 phones

Figure 5.15 dictates that accuracy and correctness of PE trained and tested for female_02 speaker of read speech with 34 phones is 42.0% and 49.3% respectively.

Table 5.5 displays the accuracy and correctness of PE for both female speakers of read speech, trained and tested with 34 phonemes.

Table 5.5: Testing accuracy of PE with 34 phonemes (including silence) for each individual female

Read Speech Data	Training Data (in Minutes)	Testing Data (in Minutes)	Accuracy (in %)	Correctness (in %)
Female_01	10.35	3.45	39.4	44.0
Female_02	7.47	2.4	42.0	49.0

PE has also been built for individual male speakers of read speech, both for 30 phonemes and 34 phonemes. Figure 5.16 and Figure 5.17 shows the confusion matrix for PE trained and tested for individual male speakers, with 30 phones.

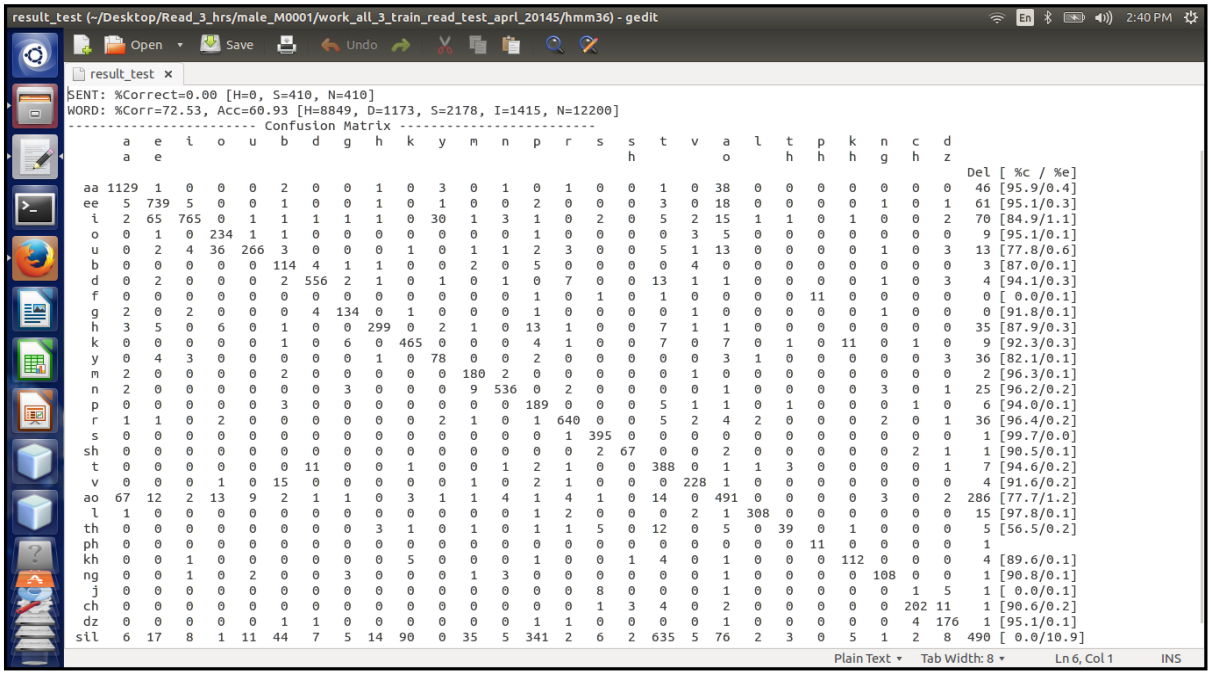


Figure 5.16: Testing accuracy of PE, trained for male_01 speaker of read speech with 30 phonemes

Figure 5.16 displays that the PE for male_01 speaker of read speech with 30 phones obtained 60.9% of accuracy and 72.5% of correctness.

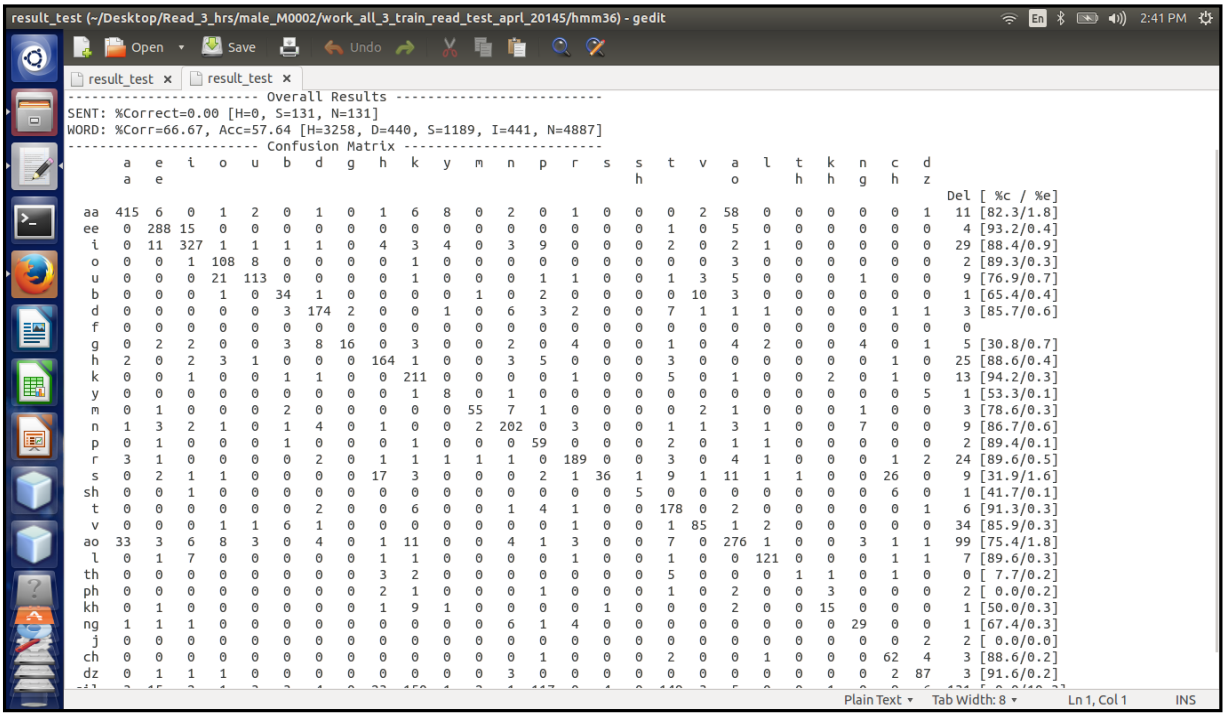


Figure 5.17: Testing accuracy of PE trained for male_02 speaker of read speech with 30 phonemes

Figure 5.17 displays that the PE for male_02 speaker of read speech with 30 phones obtained 57.6% of accuracy and 66.6% of correctness.

Table 5.6 clearly dictates the duration of speech of each male individual, its division in training and testing data set for 30 phonemes, and accuracy and correctness of each speaker.

Table 5.6: Testing accuracy of PE with 30 phonemes (including silence) for each male individual

Read Speech Data	Training Data (in Minutes)	Testing Data (in Minutes)	Accuracy (in %)	Correctness (in %)
Male_01	92.34	30.25	60.9	72.5
Male_02	30.5	9.98	57.6	66.7

The same process has been revised to train and test the PE with 34 phonemes (including silence).

Figure 5.18 and Figure 5.19 displays the Confusion matrix for PE trained for both the male speakers of read speech with 34 phones.

Figure 5.18: Confusion matrix for PE trained for male_01 of read speech with 34 phonemes

Figure 5.18 displays the confusion matrix for PE trained for male_01 speaker of read speech which depicts that the PE has obtained the 65.3% of correctness and 52.9% of accuracy.

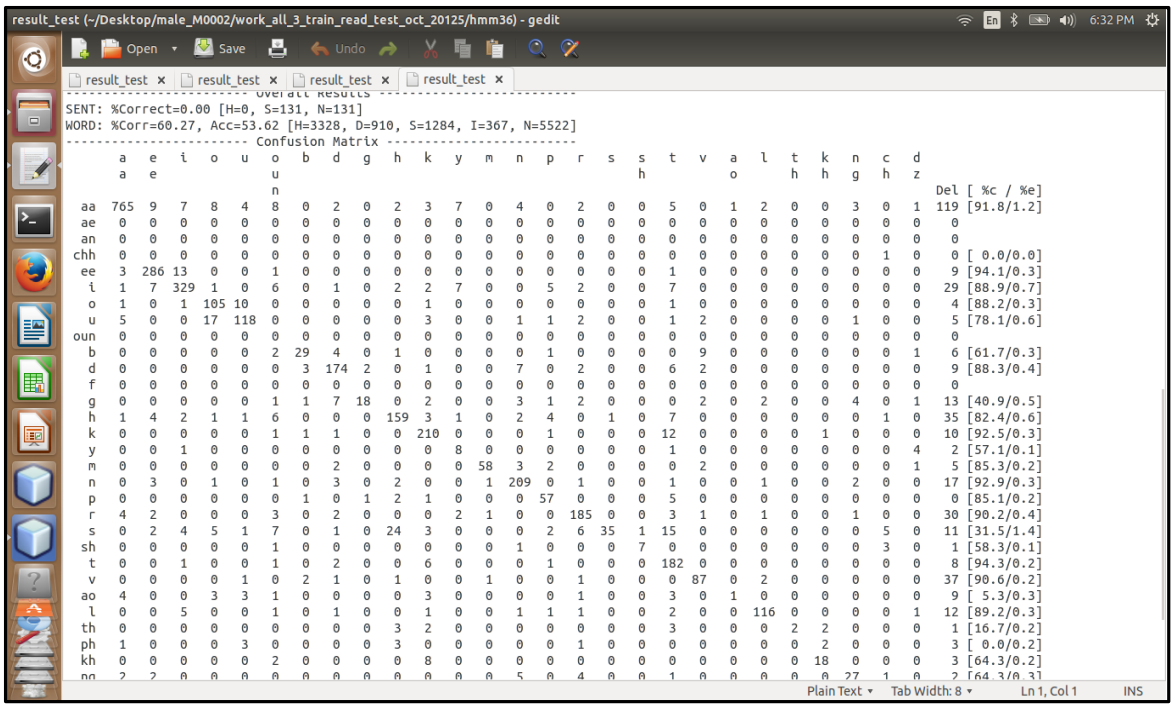


Figure 5.19: Confusion matrix for PE trained for male_02 of read speech with 34 phonemes

Figure 5.19 displays the Confusion matrix for PE trained for male_02 of read speech with 34 phones, which clearly dictates the accuracy and correctness of PE is 53.6% and 60.3%, respectively.

Table 5.7: Testing Accuracy of PE with 34 phonemes (including silence) for each male individuals

Read Speech Data	Training Data (in Minutes)	Testing Data (in Minutes)	Accuracy (in %)	Correctness (in %)
Male_01	92.34	30.25	52.90	65.3
Male_02	30.5	9.98	53.62	60.3

Table 5.7 clearly dictates the duration of speech of each male individual, its division in training and testing data set for 34 phonemes, and accuracy and correctness of each speaker.

5.3.2 Performance Evaluation for Lecture mode of Speech

PE has been developed for Lecture speech mode for both the categories, 30 phonemes and 34 phonemes. PE for lecture speech mode has been trained and tested speaker wise and transcriber wise. Figure 5.20 and 5.21 displays the testing accuracy of PE trained for total data of lecture speech with both, 30 phonemes and 34 phonemes.

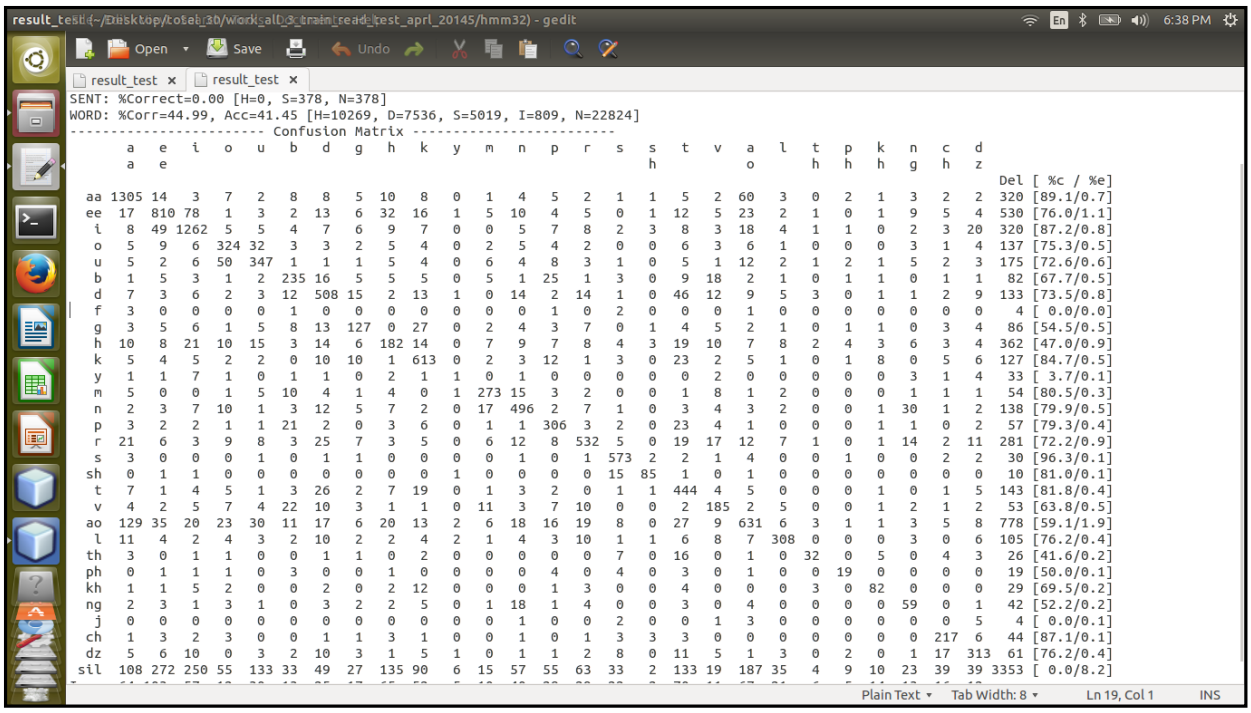


Figure 5.20: Confusion matrix for PE trained for total data lecture speech with 30 phonemes

Figure 5.20 displays the confusion matrix for PE trained and tested for total data of lecture speech with 30 phones. The matrix dictates that the PE has obtained 45.0% of correctness and 41.5% of accuracy.

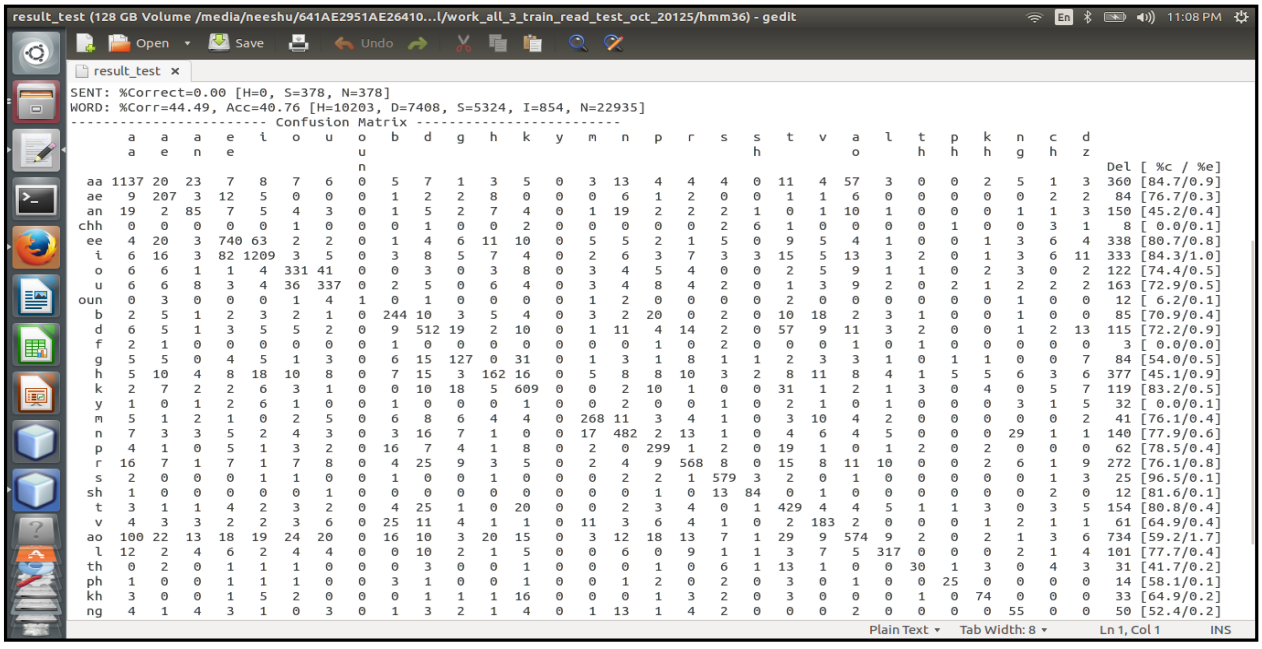


Figure 5.21: Confusion matrix for PE trained for total lecture speech data with 34 phonemes

Figure 5.21 displays the confusion matrix for PE trained and tested for total data of lecture speech with 34 phones. The matrix dictates that the PE has obtained 44.5% of correctness and 40.8% of accuracy.

Table 5.8: Testing accuracy of PE for total data of lecture speech

Lecture Speech Data	No. of Phones	Training Data (in Minutes)	Testing Data (in Minutes)	Accuracy (in %)	Correctness (in %)
Total_data	30	93.31	31.10	41.5	45.0
Total_data	34	93.31	31.10	40.8	44.5

Table 5.8 clearly dictates the testing accuracy of PE trained for total data of lecture speech mode with 30 phones as well as 34 phonemes, corresponding to their training and testing data.

PE has also been trained for various transcribers of lecture speech mode and the experiment has been done for both 30 phonemes and 34 phonemes. Figure 5.22, 5.23, 5.24 and 5.25 displays the testing accuracy of PE for each individual transcriber for 30 phonemes.

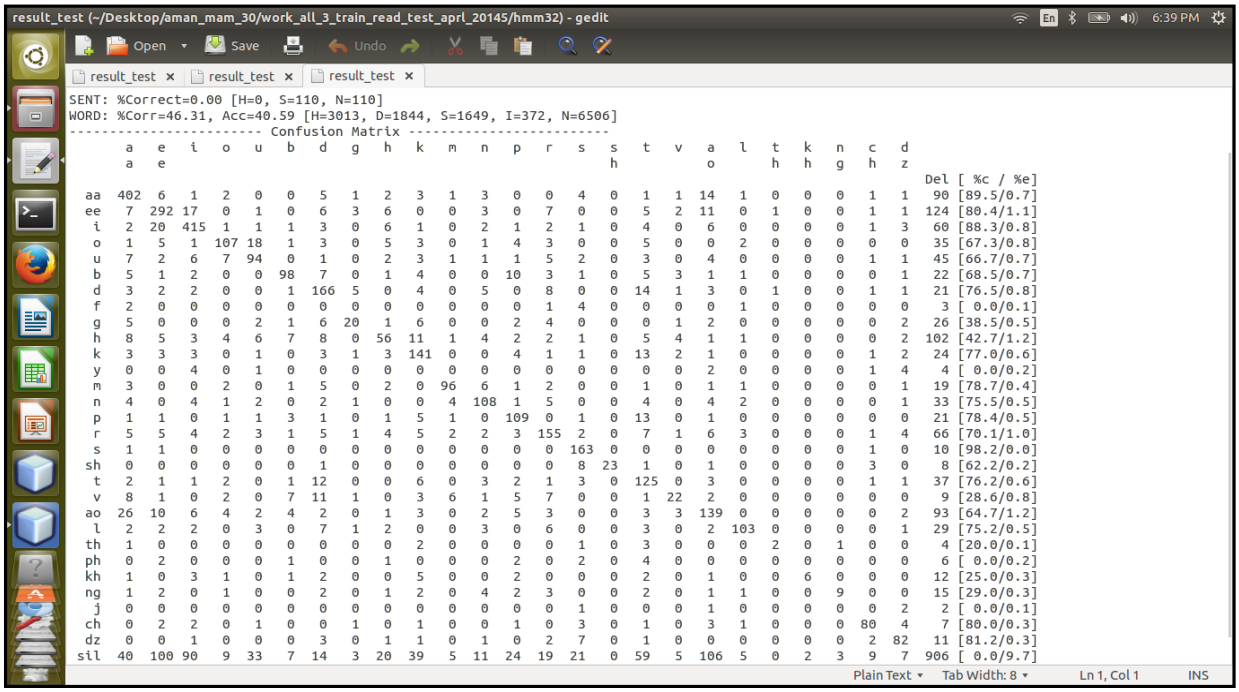


Figure 5.22: Confusion matrix for PE for transcriber_01 of lecture speech with 30 phonemes

Figure 5.22 displays the confusion matrix for PE trained and tested for transcriber_01 of lecture speech with 30 phones. The matrix dictates that the PE has obtained 46.3% of correctness and 40.6% of accuracy.

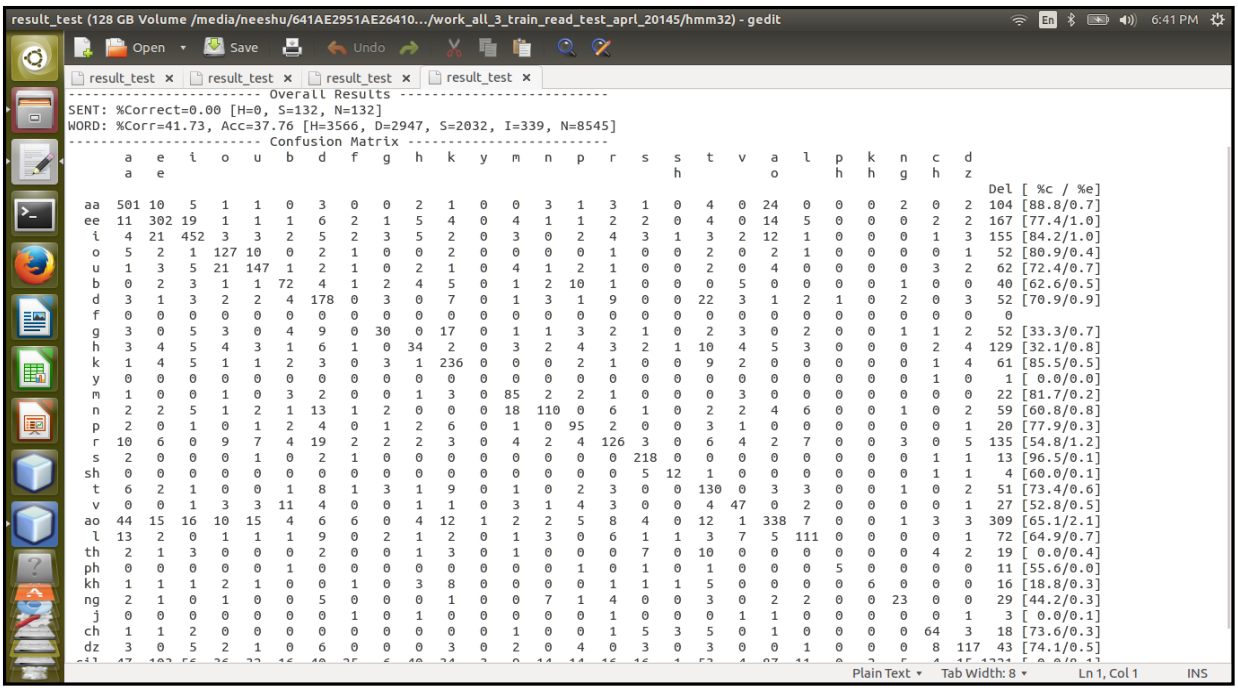


Figure 5.23: Confusion matrix for PE for transcriber_02 of lecture speech with 30 phonemes

Figure 5.23 displays the confusion matrix for PE trained and tested for transcriber_02 of lecture speech with 30 phones. The matrix dictates that the PE has obtained 46.3% of correctness and 40.6% of accuracy.

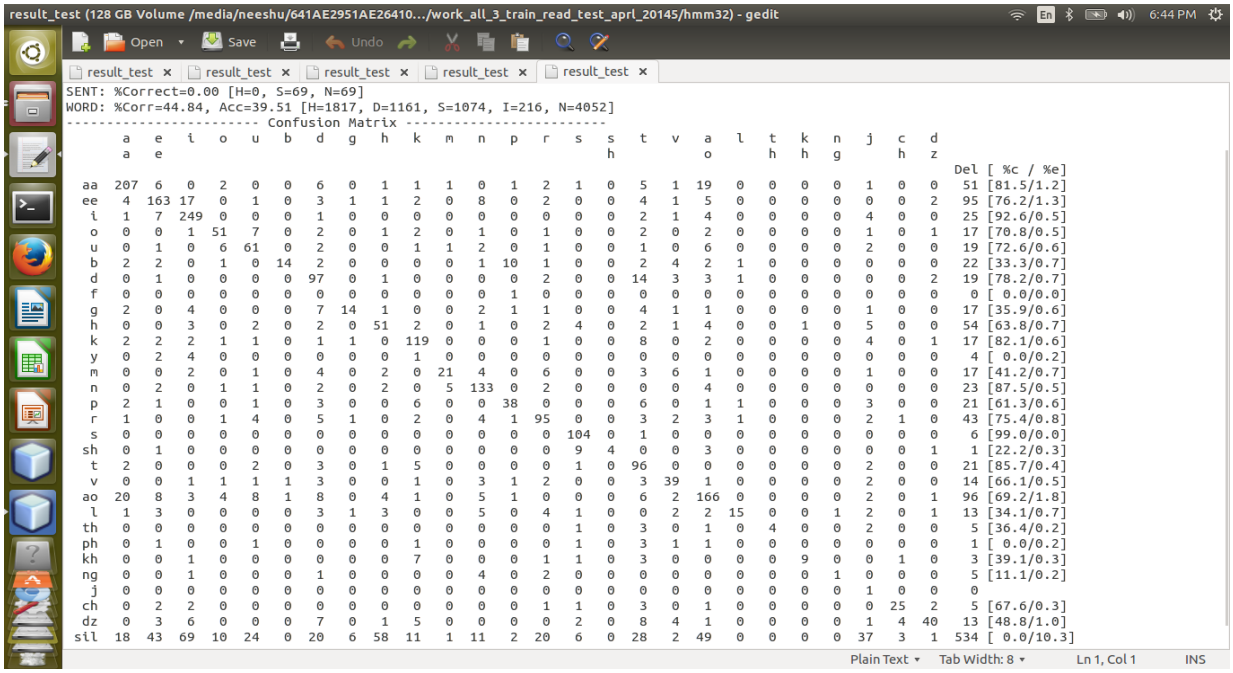


Figure 5.24: Confusion matrix for PE for transcriber_03 of lecture speech with 30 phonemes

Figure 5.24 displays the confusion matrix for PE trained and tested for transcriber_03 of lecture speech with 30 phones. The matrix dictates that the PE has obtained 44.8% of correctness and 39.5% of accuracy.

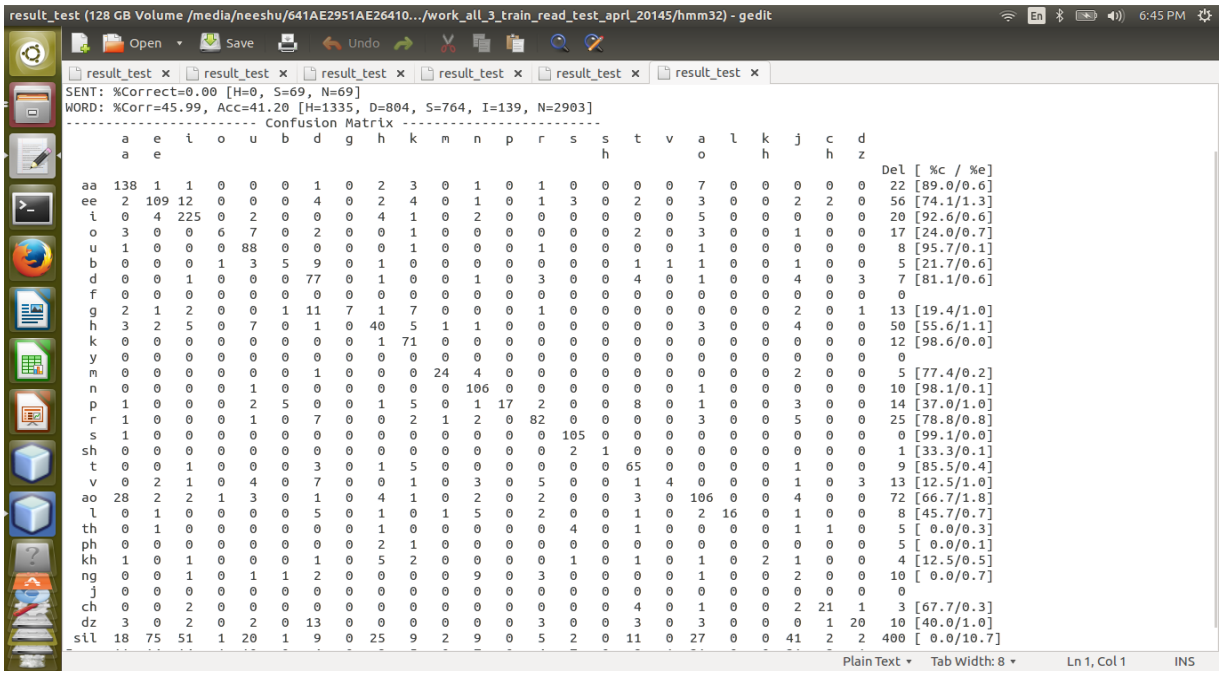


Figure 5.25: Confusion matrix for PE for transcriber_04 of lecture speech with 30 phones

Figure 5.25 displays the confusion matrix for PE trained and tested for transcriber_04 of lecture speech with 30 phones. The matrix dictates that the PE has obtained 46.0% of correctness and 41.2% of accuracy.

For comparative analysis, table 5.9 enlists the accuracy and correctness of each transcriber corresponding to their transcribed data.

Table 5.9: Testing accuracy of PE for various transcribers of lecture speech with 30 phonemes

Transcriber's ID	Training Data (in minutes)	Testing data (in minutes)	Correctness (in %)	Accuracy (in %)
Transcriber_01	35.25	11.76	46.3	40.6
Transcriber_02	32.45	10.81	41.7	37.8
Transcriber_03	15.36	4.12	44.8	39.5
Transcriber_04	15.34	4.13	46.0	41.2

The same experiment of calculating correctness and accuracy has been done for all the transcribers, corresponding to data transcribed by them, for 34 phonemes. Figure 5.26, 5.27, 5.28, and 5.30 shows the testing accuracy and correctness of confusion matrix of transcriber wise trained PE for lecture speech.

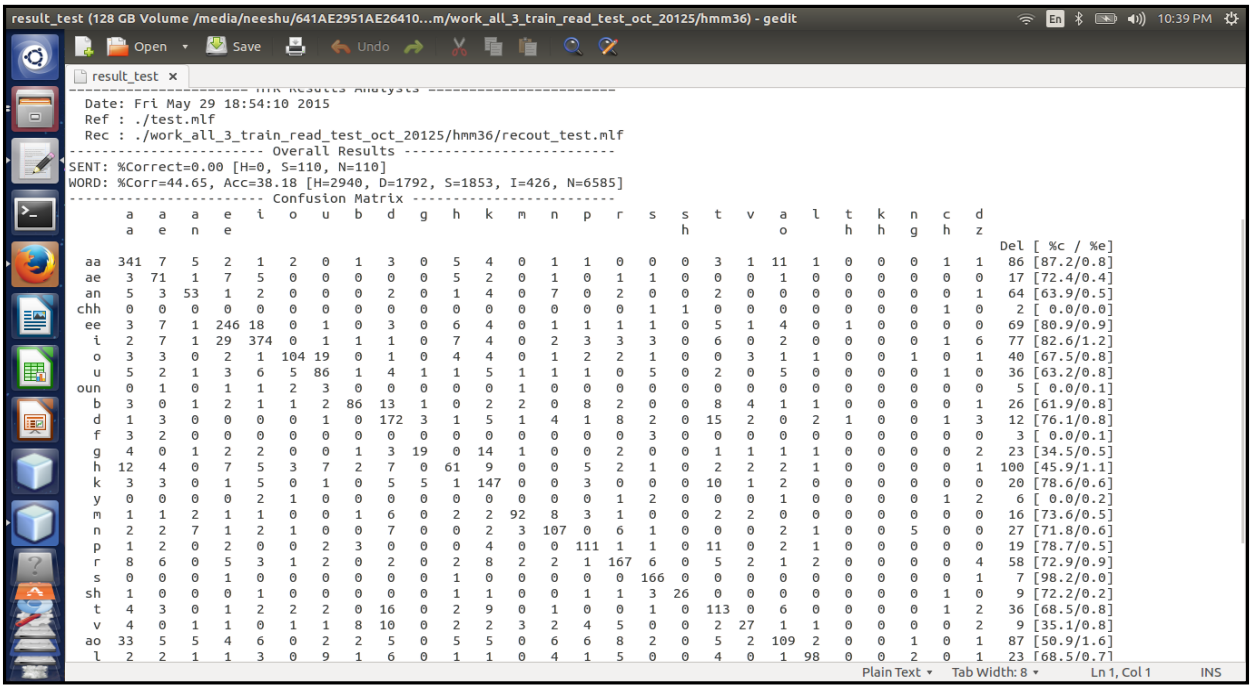


Figure 5.26: Confusion matrix for PE for transcriber_01 of lecture speech with 34 phonemes

Figure 5.26 displays the confusion matrix for PE trained and tested for transcriber_01 of lecture speech with 34 phones. The matrix dictates that the PE has obtained 44.7% of correctness and 38.2% of accuracy.

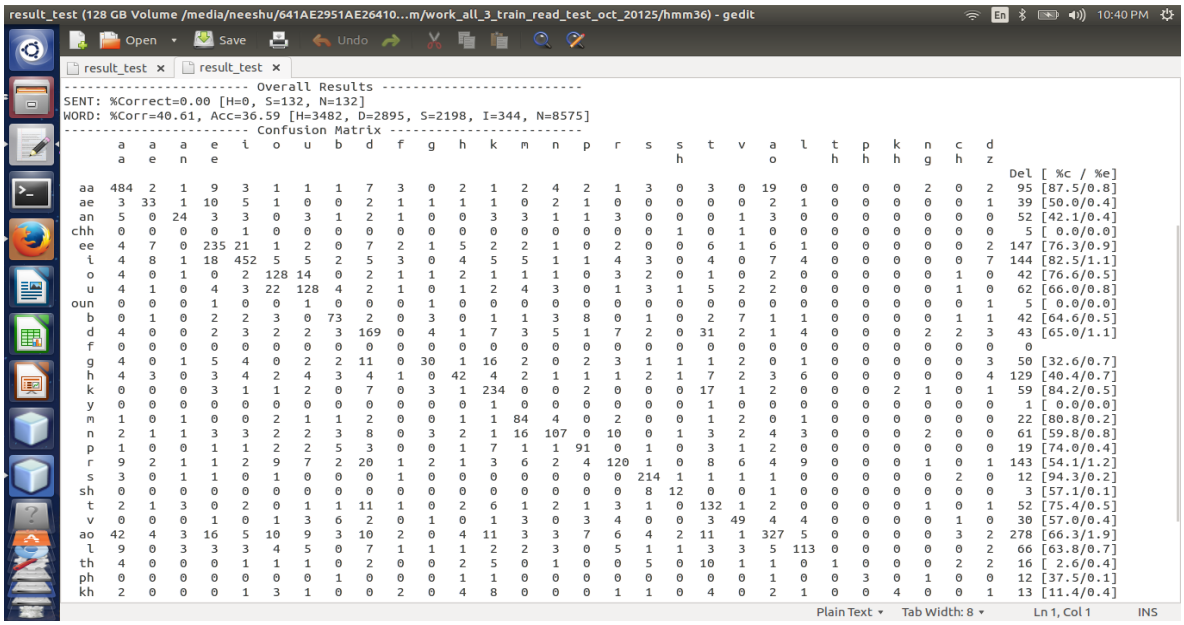


Figure 5.27: Confusion matrix for PE for transcriber_02 of lecture speech with 34 phonemes

Figure 5.27 displays the confusion matrix for PE trained and tested for transcriber_01 of lecture speech with 34 phones. The matrix dictates that the PE has obtained 40.6% of correctness and 36.6% of accuracy.

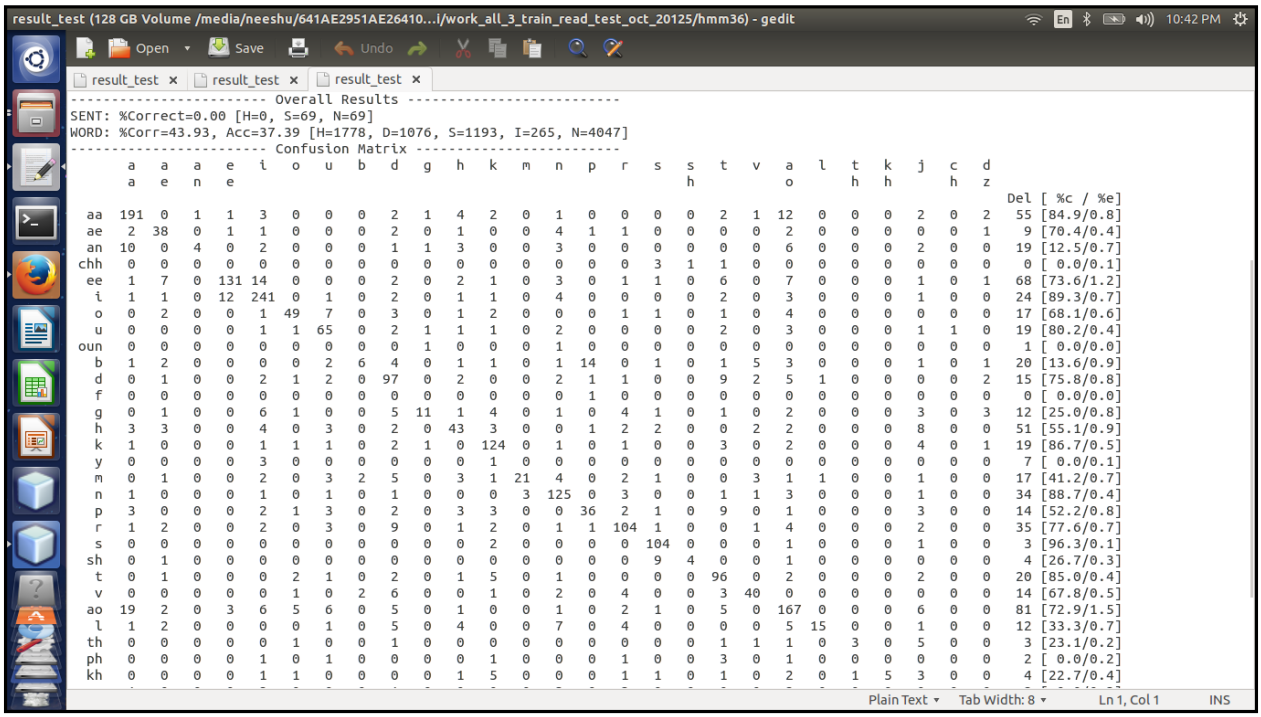


Figure 5.28: Confusion matrix for PE for transcriber_03 of lecture speech with 34 phonemes

Figure 5.28 displays the confusion matrix for PE trained and tested for transcriber_03 of lecture speech with 34 phones. The matrix dictates that the PE has obtained 43.9% of correctness and 37.4% of accuracy.

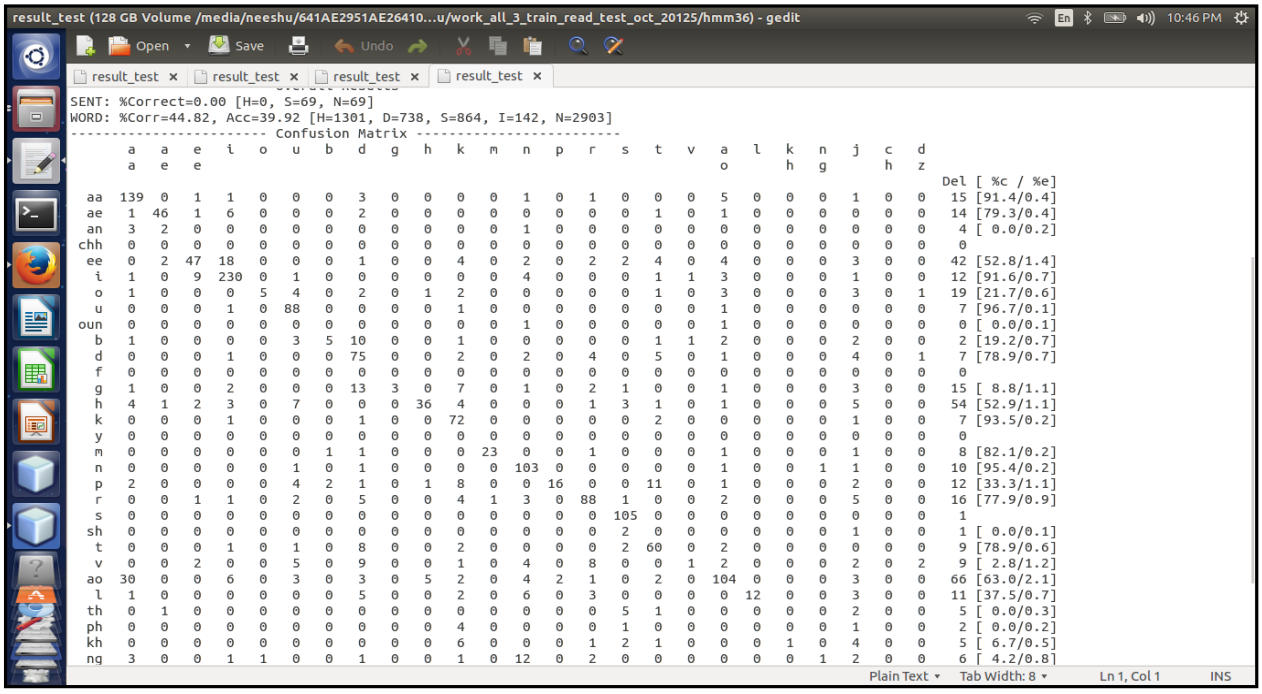


Figure 5.29: Confusion matrix for PE for transcriber_04 of lecture speech with 34 phonemes

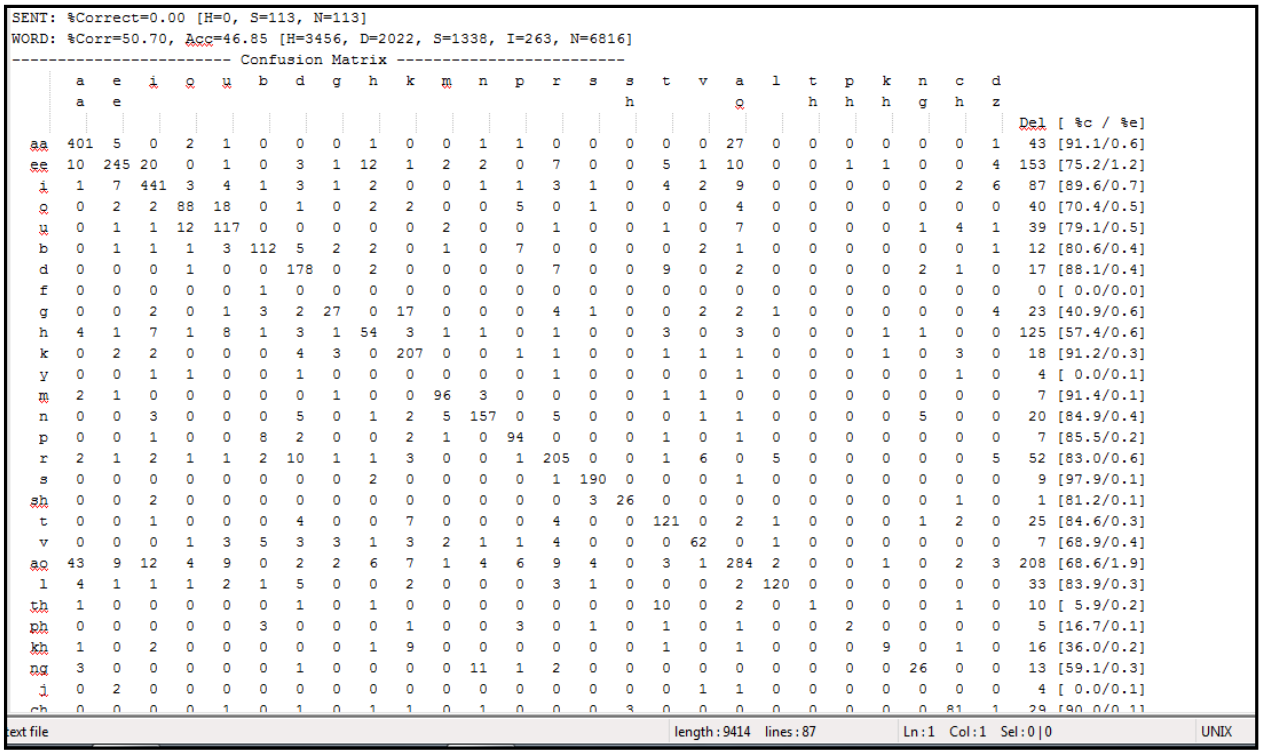
Figure 5.29 displays the confusion matrix for PE trained and tested for transcriber_04 of lecture speech with 34 phonemes. The matrix dictates that the PE has obtained 44.8% of correctness and 39.9% of accuracy.

For comparative analysis, table 5.10 represents the accuracy and correctness of each transcriber corresponding to their transcribed data.

Table 5.10: Testing accuracy of PE for various transcribers of lecture speech with 34 phonemes

Transcriber's ID	Training Data (in minutes)	Testing data (in minutes)	Correctness (in %)	Accuracy (in %)
Transcriber_01	35.25	11.76	46.3	40.6
Transcriber_02	32.45	10.81	41.7	37.8
Transcriber_03	15.36	4.12	44.8	39.5
Transcriber_04	15.34	4.13	46.0	41.2

As discussed earlier, PE has also been trained for each speaker of lecture speech. The experiment has been done for both the schemes, i.e. 30 phonemes and 34 phonemes (including silence). Figure 5.30, 5.31, and 5.32 shows the confusion matrix for PE with 30 phonemes, trained and tested for each speaker of lecture speech.



```
SENT: %Correct=0.00 [H=0, S=113, N=113]
WORD: %Corr=50.70, Acc=46.85 [H=3456, D=2022, S=1338, I=263, N=6816]
---------------------- Confusion Matrix -----------------------
      a   e   i   o   u   b   d   g   h   k   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 401   5   0   2   1   0   0   0   1   0   0   1   1   0   0   0   0   0  27   0   0   0   0   0   0   1   43 [91.1/0.6]
  ee  10 245  20   0   1   0   3   1  12   1   2   2   0   7   0   0   5   1  10   0   0   1   1   0   0   4  153 [75.2/1.2]
   i   1   7 441   3   4   1   3   1   2   0   0   1   1   3   1   0   4   2   9   0   0   0   0   0   2   6   87 [89.6/0.7]
   o   0   2   2  88  18   0   1   0   2   2   0   0   5   0   1   0   0   0   4   0   0   0   0   0   0   0   40 [70.4/0.5]
   u   0   1   1  12 117   0   0   0   0   0   2   0   0   1   0   0   1   0   7   0   0   0   0   1   4   1   39 [79.1/0.5]
   b   0   1   1   1   3 112   5   2   2   0   1   0   7   0   0   0   0   2   1   0   0   0   0   0   0   1   12 [80.6/0.4]
   d   0   0   0   1   0   0 178   0   2   0   0   0   0   7   0   0   9   0   2   0   0   0   0   2   1   0   17 [88.1/0.4]
   f   0   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]
   g   0   0   2   0   1   3   2  27   0  17   0   0   0   4   1   0   0   2   2   1   0   0   0   0   0   4   23 [40.9/0.6]
   h   4   1   7   1   8   1   3   1  54   3   1   1   0   1   0   0   3   0   3   0   0   0   1   1   0   0  125 [57.4/0.6]
   k   0   2   2   0   0   0   4   3   0 207   0   0   1   1   0   0   1   1   1   0   0   0   1   0   3   0   18 [91.2/0.3]
   y   0   0   1   1   0   0   1   0   0   0   0   0   0   1   0   0   0   0   1   0   0   0   0   0   1   0    4 [ 0.0/0.1]
   m   2   1   0   0   0   0   0   1   0   0  96   3   0   0   0   0   1   1   0   0   0   0   0   0   0   0    7 [91.4/0.1]
   n   0   0   3   0   0   0   5   0   1   2   5 157   0   5   0   0   0   1   1   0   0   0   0   5   0   0   20 [84.9/0.4]
   p   0   0   1   0   0   8   2   0   0   2   1   0  94   0   0   0   1   0   1   0   0   0   0   0   0   0    7 [85.5/0.2]
   r   2   1   2   1   1   2  10   1   1   3   0   0   1 205   0   0   1   6   0   5   0   0   0   0   0   5   52 [83.0/0.6]
   s   0   0   0   0   0   0   0   0   2   0   0   0   0   1 190   0   0   0   1   0   0   0   0   0   0   0    9 [97.9/0.1]
  sh   0   0   2   0   0   0   0   0   0   0   0   0   0   0   3  26   0   0   0   0   0   0   0   0   1   0    1 [81.2/0.1]
   t   0   0   1   0   0   0   4   0   0   7   0   0   0   4   0   0 121   0   2   1   0   0   0   1   2   0   25 [84.6/0.3]
   v   0   0   0   1   3   5   3   3   1   3   2   1   1   4   0   0   0  62   0   1   0   0   0   0   0   0    7 [68.9/0.4]
  ao  43   9  12   4   9   0   2   2   6   7   1   4   6   9   4   0   3   1 284   2   0   0   1   0   2   3  208 [68.6/1.9]
   l   4   1   1   1   2   1   5   0   0   2   0   0   0   3   1   0   0   0   2 120   0   0   0   0   0   0   33 [83.9/0.3]
  th   1   0   0   0   0   0   1   0   1   0   0   0   0   0   0   0  10   0   2   0   1   0   0   0   1   0   10 [ 5.9/0.2]
  ph   0   0   0   0   0   3   0   0   0   1   0   0   3   0   1   0   1   0   1   0   0   2   0   0   0   0    5 [16.7/0.1]
  kh   1   0   2   0   0   0   0   0   1   9   0   0   0   0   0   0   1   0   1   0   0   0   9   0   1   0   16 [36.0/0.2]
  ng   3   0   0   0   0   0   1   0   0   0   0  11   1   2   0   0   0   0   0   0   0   0   0  26   0   0   13 [59.1/0.3]
   j   0   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1   1   0   0   0   0   0   0   0    4 [ 0.0/0.1]
  ch   0   0   0   0   1   0   1   0   1   1   0   1   0   0   0   3   0   0   0   0   0   0   0   0  81   1   29 [90.0/0.1]
```

```
text file                    length : 9414   lines : 87        Ln : 1   Col : 1   Sel : 0 | 0                UNIX
```

Figure 5.30: Confusion matrix for PE for speaker_01 of lecture speech with 30 phonemes

Figure 5.30 displays the confusion matrix for PE trained and tested for speaker_01 of lecture speech with 30 phonemes. The matrix dictates that the PE has obtained 50.7% of correctness and 46.9% of accuracy.

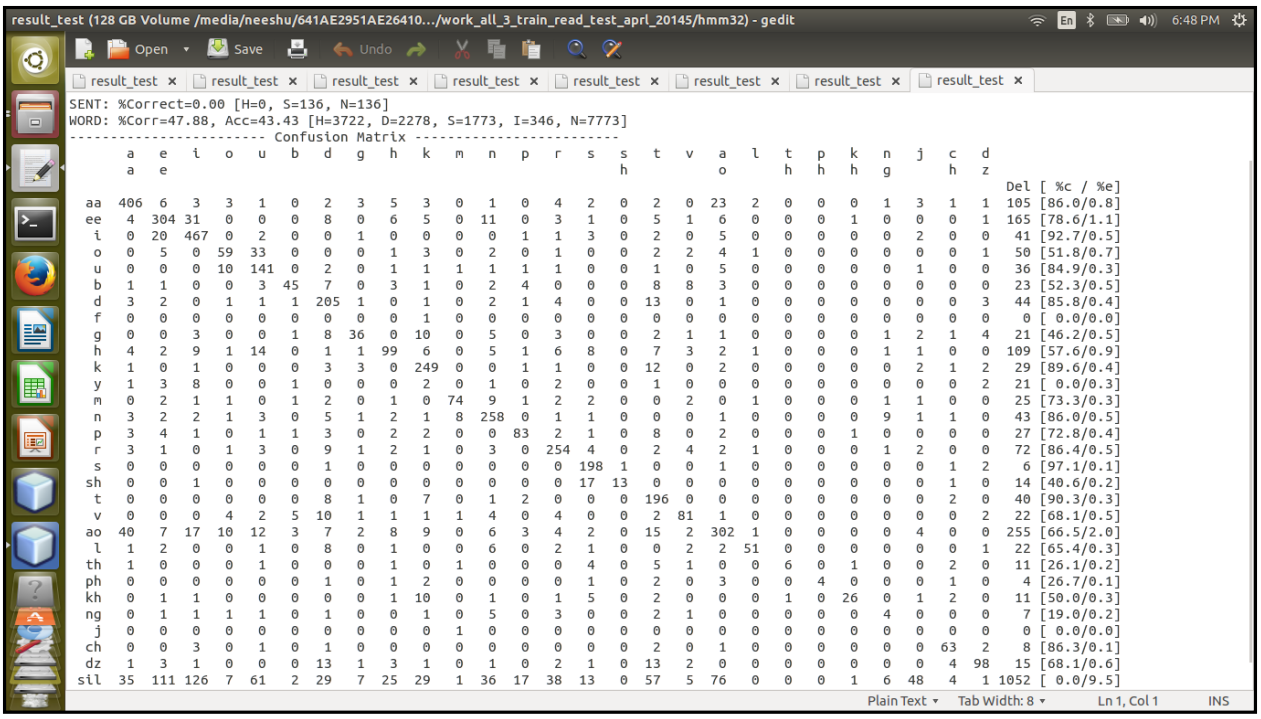


Figure 5.31: Confusion matrix for PE for speaker_02 of lecture speech with 30 phonemes

Figure 5.31 shows the confusion matrix for PE trained and tested for speaker_02 of lecture speech with 30 phonemes. The matrix dictates that the PE has obtained 47.9% of correctness and 43.4% of accuracy.

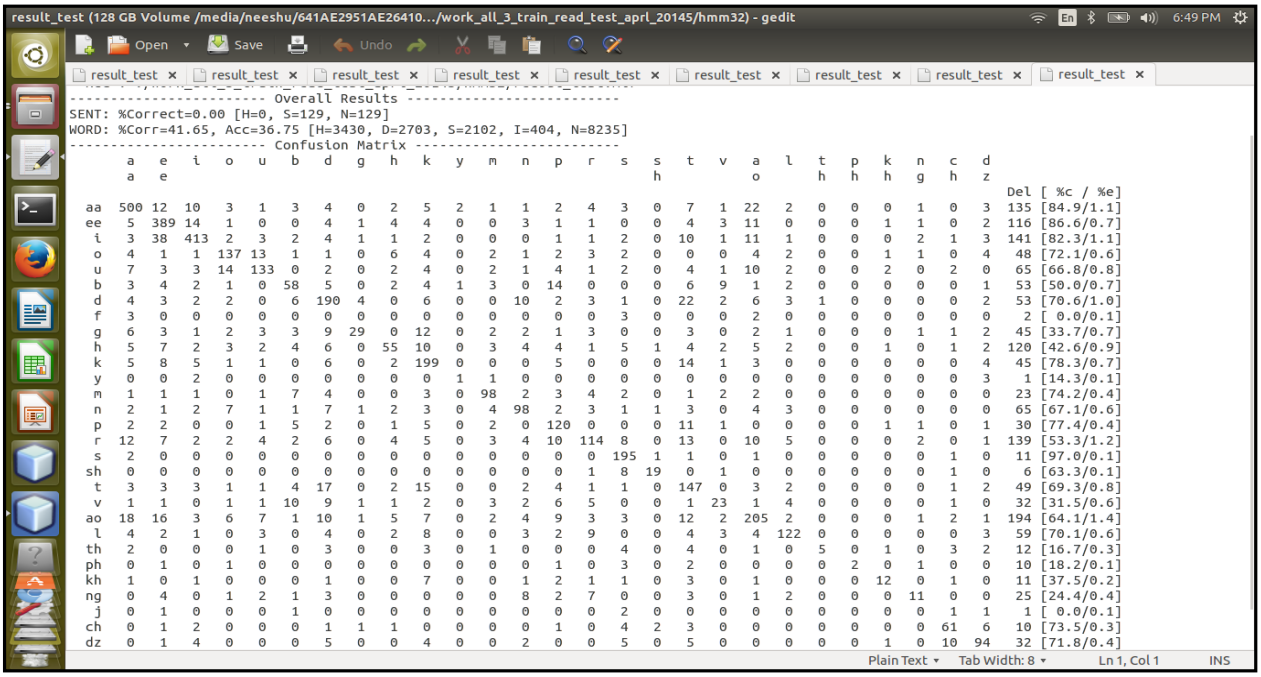


Figure 5.32: Confusion matrix for PE for speaker_03 of lecture speech with 30 phonemes

Figure 5.32 shows the confusion matrix for PE trained and tested for speaker_03 of lecture speech with 30 phonemes. The matrix dictates that the PE has obtained 41.7% of correctness and 36.8% of accuracy.

For comparative analysis, table 5.11 represents the accuracy and correctness of each speaker corresponding to their recorded data.

Table 5.11: Testing accuracy of PE for various speakers of lecture speech with 30 phonemes

speaker's ID	Training Data (in minutes)	Testing data (in minutes)	Correctness (in %)	Accuracy (in %)
Speaker_01	36.28	12.10	50.7	46.85
Speaker_02	25.23	9.03	47.88	43.43
Speaker_03	31.40	9.57	41.65	36.75

Phonetic Engine has also been trained for each speaker of lecture speech with 34 phonemes (including silence). Figure 5.34, 5.34 and 5.35 shows the confusion matrix for PE with 34 phonemes, trained and tested for each speaker of lecture speech.

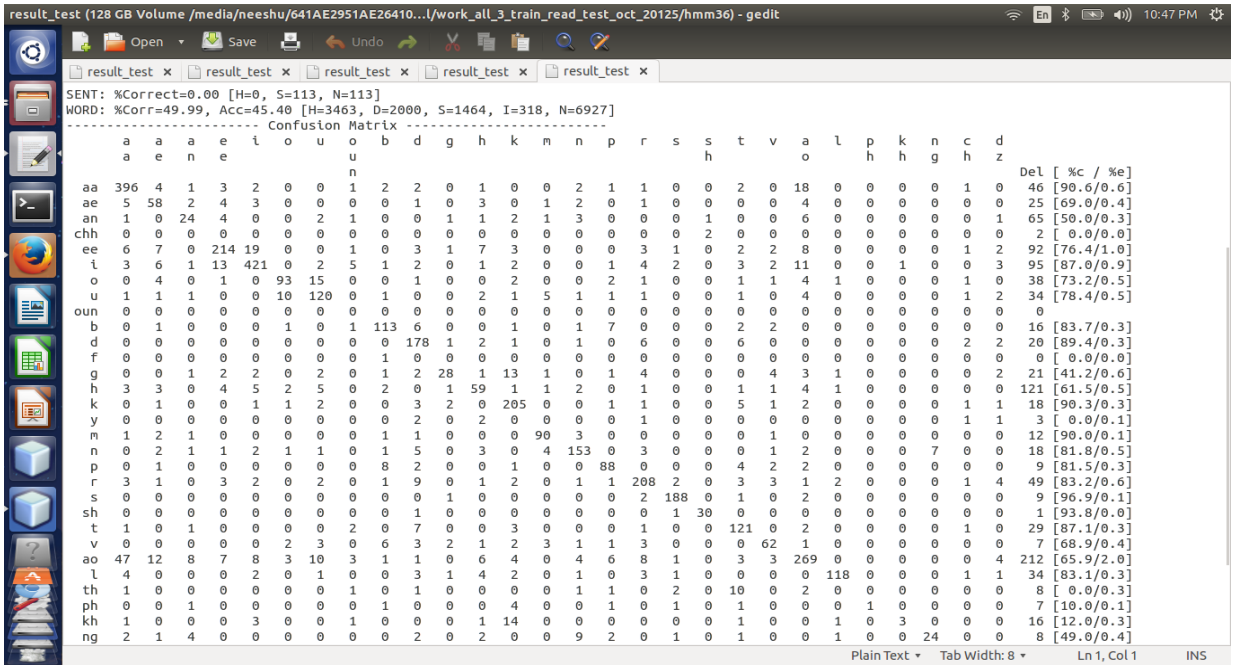


Figure 5.33: Confusion matrix for PE for speaker_01 of lecture speech with 34 phonemes

Figure 5.34 shows the confusion matrix for PE trained and tested for speaker_01 of lecture speech with 34 phonemes. The matrix dictates that the PE has obtained 50.0% of correctness and 45.4% of accuracy.

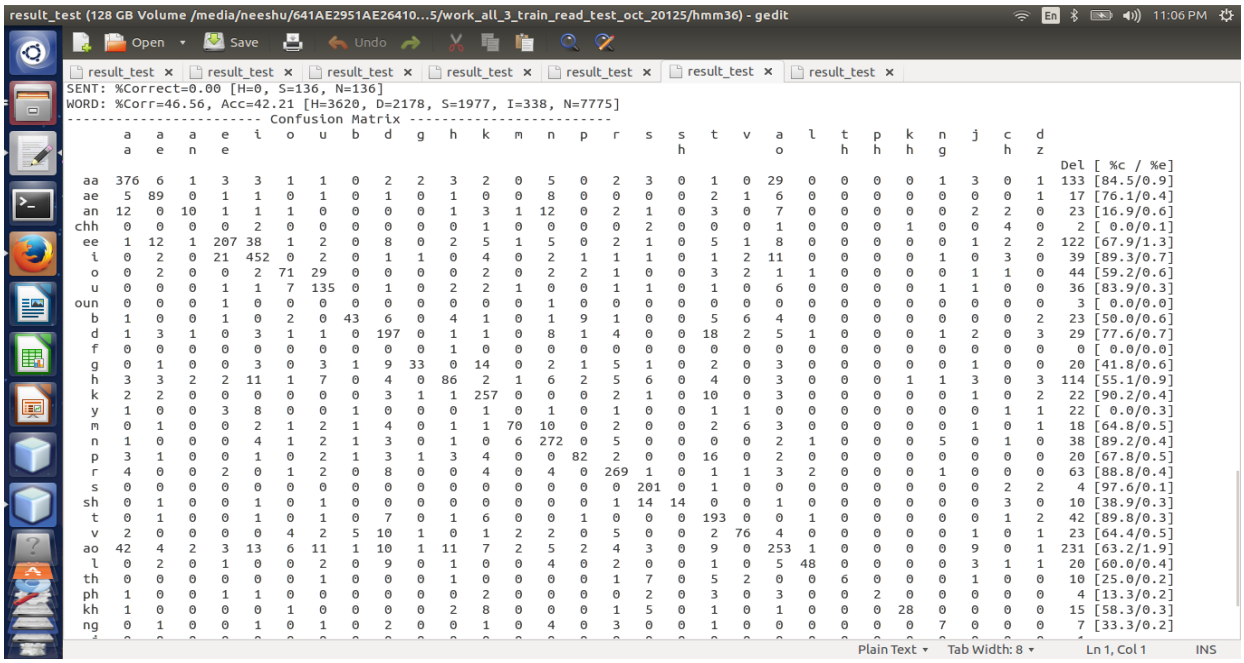


Figure 5.34: Confusion matrix for PE for speaker_02 of lecture speech with 34 phonemes

Figure 5.34 shows the confusion matrix for PE trained and tested for speaker_02 of lecture speech with 34 phonemes. The matrix dictates that the PE has obtained 46.6% of correctness and 42.2% of accuracy.

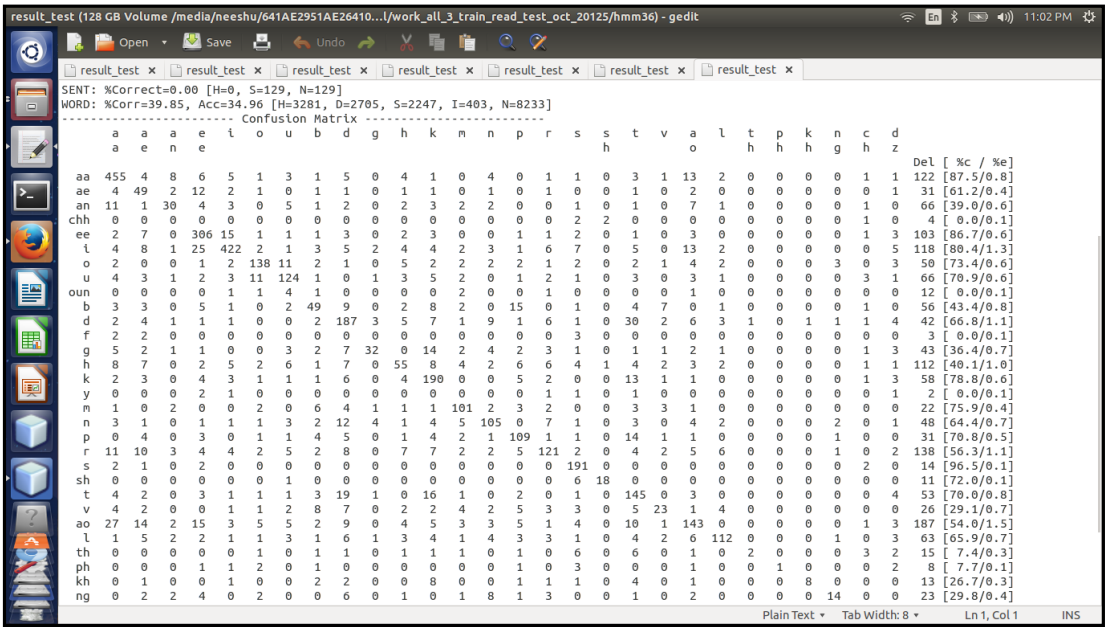


Figure 5.35: Confusion matrix for PE for speaker_03 of lecture speech with 34 phonemes

Figure 5.35 shows the confusion matrix for PE trained and tested for speaker_03 of lecture speech with 34 phonemes. The matrix dictates that the PE has obtained 39.9% of correctness and 35.0% of accuracy.

For comparative analysis, table 5.12 represents the accuracy and correctness of each speaker corresponding to their recorded data that has been processed with 34 phonemes scheme.

Table 5.12: Testing Accuracy of PE for various speakers of lecture speech with 34 phonemes

speaker's ID	Training Data (in minutes)	Testing data (in minutes)	Correctness (in %)	Accuracy (in %)
Speaker_01	36.28	12.10	50.0	45.4
Speaker_02	25.23	9.03	46.6	42.2
Speaker_03	31.40	9.57	39.9	35.0

5.3.3 Performance Evaluation for Conversational Mode of Speech

In this mode of speech, the confusion matrix for PE trained and tested for both 30 phonemes and 34 phonemes. In this mode of speech, more than 2 speakers are talking on a topic in their natural mode of speech in the open environment.

Approximately, 20 minutes of speech data has been processed. Figure 5.36 and Figure 5.37 shows the confusion matrix, which depicts the testing accuracy of trained PE for conversational mode of speech with 30 phonemes and 34 phonemes (including silence), respectively.

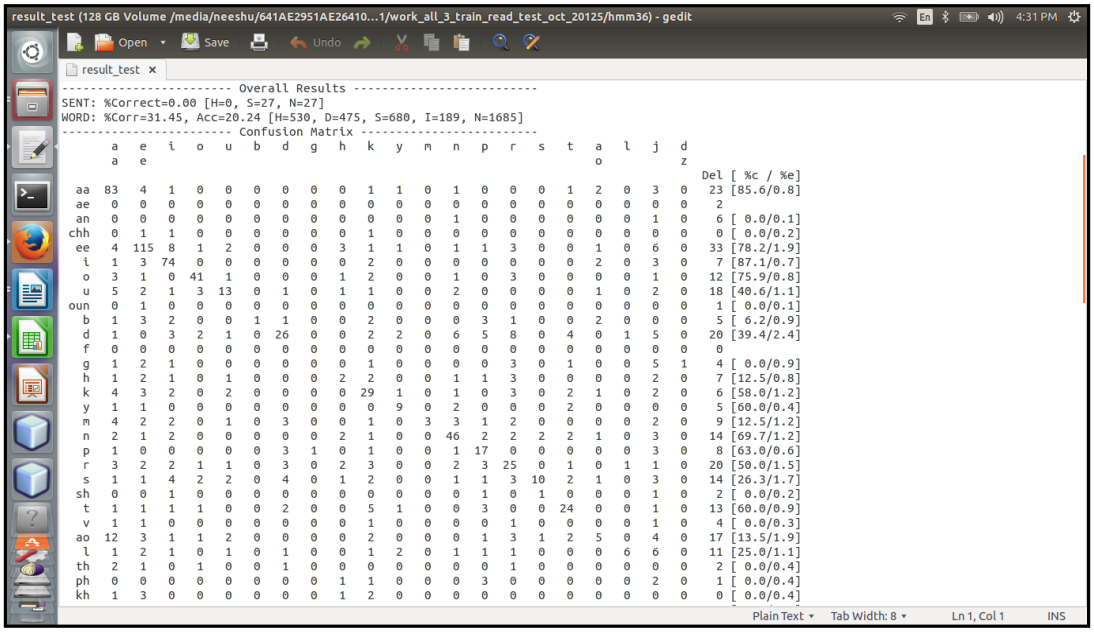


Figure 5.36: Confusion matrix for PE, trained with 30 phonemes for conversational speech

Figure 5.36 shows the confusion matrix for PE trained and tested for conversational speech with 30 phonemes. The system obtained 31.5% of correctness and 20.2% of accuracy.

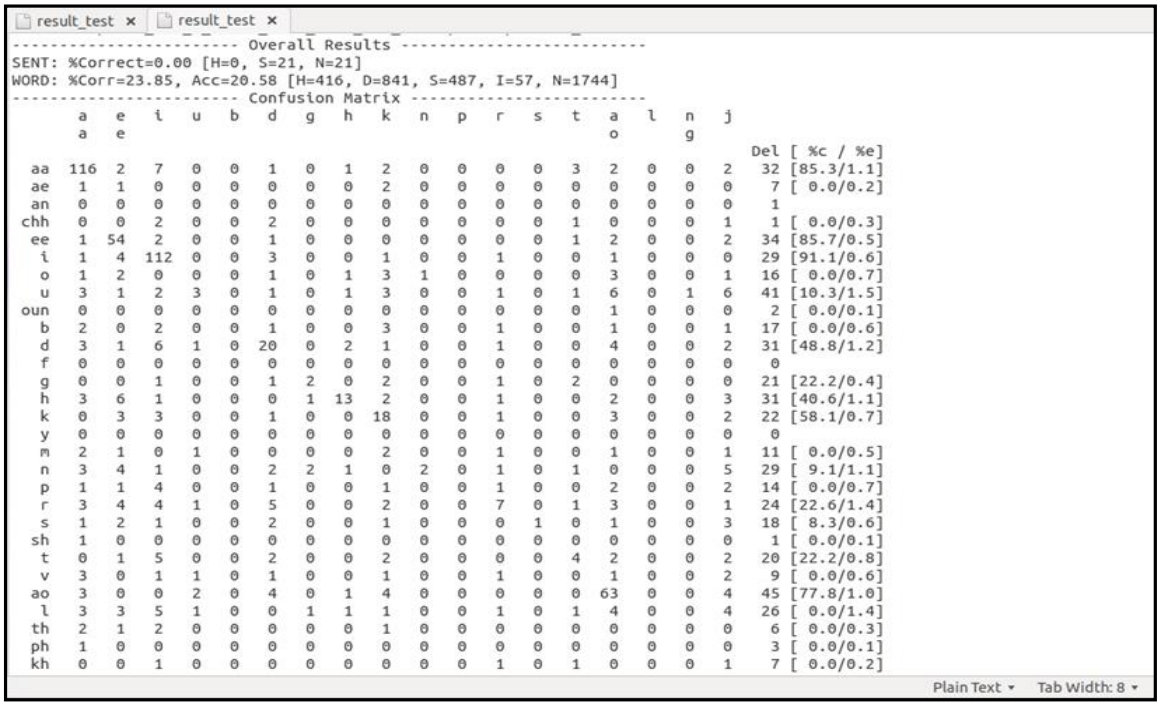


Figure 5.37: Confusion matrix for PE, trained with 34 phonemes for conversational speech

Figure 5.37 shows the confusion matrix for PE trained and tested for conversational speech with 30 phonemes. The system obtained 23.9% of correctness and 20.6% of accuracy.

For comparative analysis, table 5.13 represents the accuracy and correctness of conversational speech data that has been processed with 34 phonemes scheme.

Figure 5.13: Correctness and accuracy of PE trained for Conversational mode speech

Conversational Speech Data	No. of phones	Training Data (in Minutes)	Testing Data (in Minutes)	Correctness (in %)	Accuracy (in %)
Convsp_30	30	15.08	5.04	31.4	20.2
Convpsp_34	34	15.08	5.04	23.9	20.6

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

In this work, data has been collected and processed in three different modes: read speech mode, lecture mode and conversational 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 186.34 minutes, in lecture mode, collected from a radio channel, Punjabi radio USA of about 124.41 minutes and in conversational mode, the conversation and discussion of Punjabi speakers are recorded of about 20.12 minutes.

Collected data has been 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 with two different schemes. The first one is using 30 phones (including silence) for preparing MLF file and the second one is using 34 phones (including silence) for preparing the MLF file. This file is mandatory to train the system. Such type of process model provides the phoneme level recognition for continuous speech signal of Punjabi language.

Ass there is no particular standard has been defined yet, that consist a list of unique phonemes in Punjabi language due to its variation in speaking over a large geographical area, this work contributes 4 new phones ‘ae’, ‘an’, ‘chh’, and ‘oun’ towards the digital recognition of Punjabi language, which is one of the widely spoken language in the world.

HTK toolkit is used for training and testing of the PE and the platform is Ubuntu-14.04 64-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. For another case, 34 unique phonemes have been used in place of 30 phonemes. PE is trained separately for the following cases: read speech mode, lecture mode, conversation mode. In the case of read speech mode, PE is trained for each gender and for each individual Punjabi speaker. In the case of lecture speech mode, PE is trained for each Punjabi speaker and for each transcriber, who

transcribed the Punjabi spoken utterances. In each case, PE is trained with 75.0% of data and its performance was evaluated with 25.0% remaining data.

PE got an accuracy of 61.8% in read speech mode for 30 phonemes based PE, and 54.4% for 34 phonemes based PE; 41.5% accuracy in lecture mode for 30 phonemes based PE and 40.8% for 34 phonemes based PE; 20.2% accuracy in conversational mode for 30 phonemes based PE and 20.6% accuracy for 34 phones based PE.

6.2 Future Scope

The accuracy as well as correctness of PE can be increased by collecting more data such that PE can be trained with a large amount of data. The system can be develop with large number of MFCC features and more no. of Gaussian Mixtures for training.

This can also be extended to set a standard of uniquely existed phonemes for Punjabi language as well as for its variations.

The accuracy of the system can also be increased by increasing the number of phonemes as well as training data accordingly.

This work can be extended to syllable level recognition. As syllable are the sound unit that focus on the presence vowel surrounded by consonants; so it'll give a fascinating accuracy.

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 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 tri phone 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), 134-150.

[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.

[30] 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.

[34] 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] Tiwari, Vibha, (2010).”MFCC and its Applications in speaker recognition”, *International Journal on Emerging Technologies,* 1(1), 19-22.

[39] Muda, Lindasalwa, Begam, Mumtaj and Elamvazuthi, I., (2010). “Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques”, *Journal of Computing*, 2(3), 138-143.

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

[41] Kumar, K., and Aggarwal, R. K. (2011). "Hindi speech recognition system using HTK", *International Journal of Computing and Business Research*, *2*(2), 2230-6166.

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

[43] 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.

[44] 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.

[45] 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.

[46] Vimala, C. M., and Radha, V. (2012). "Speaker Independent Isolated Speech Recognition System for Tamil Language using HMM", *Procedia Engineering, 30*, 1097-1102.

[47] 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)".

[48] 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, OCOCOSDA/CASLRE,* 1-6.

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

[50] 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.*

[51] 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.

[52] 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.

[53] Chavan, Rupali S., and Sable, Ganesh S., (2013). "An Implementation of Text Dependent Speaker Independent Isolated Word Speech Recognition Using HMM", *International Journal of Engineering Sciences Research and Technology, 2(9), 2311-2318.*

[54] Dhingra, S. Dev, Nijhawan, Geeta, and Pandit, Poonam, (2013). "Isolated Speech Recognition Using MFCC and DTW", *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* 2(8), 4085-4093.

[55] 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.

[56] Suryawanshi, Umarani J., and Ganorkar, S. R., (2014). "Hardware Implementation of Speech

Recognition Using MFCC and Euclidean Distance", *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, 3(8), 11248-11254.

[57] Hebse, Vidwath R., and G., Anitha, (2014). "The Learning Method of Speech Recognition Based on HMM", *International Journal of Innovative Research in Computer and Communication Engineering*, 3(4), 3287-3403.

[58] Muda, Lindasalwa, Begam, Mumtaj and Elamvazuthi, I., (2010). "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques", *Journal of Computing*, 2(3), 138-143.

Video Link

http://bit.do/neeshu-punjabi-asr

APPENDIX

A GUI has been developed to realize the experiments, which have been done for various speech modes with various schemes, in real world. The GUI has been developed in Java-8 with the help of NetBeans 8.0 IDE in Ubuntu Linux-14.04 LTS environment. Snapshots of this GUI are as follows.

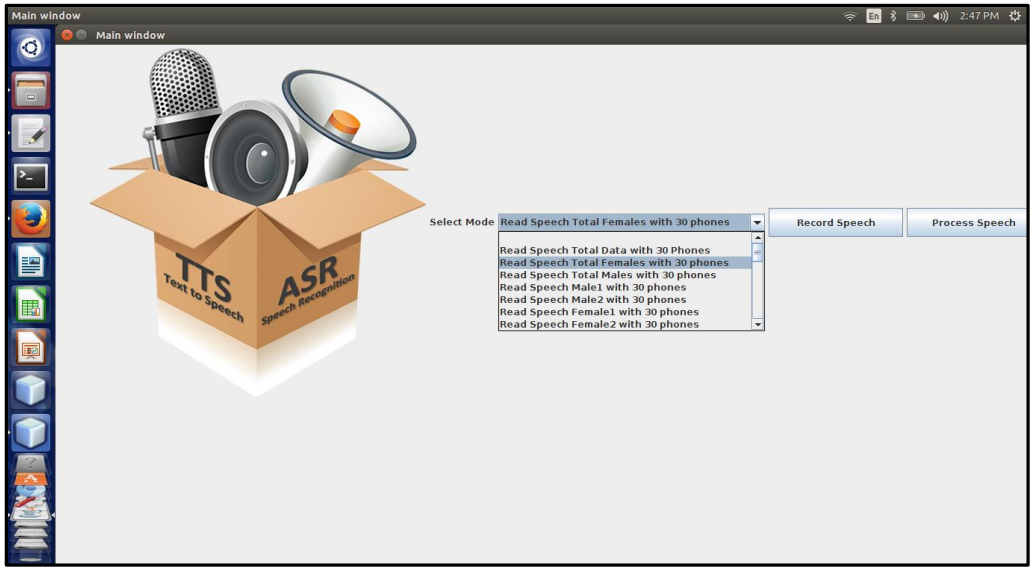


Figure A.1: Mode Selection Screen to select modes of experiment

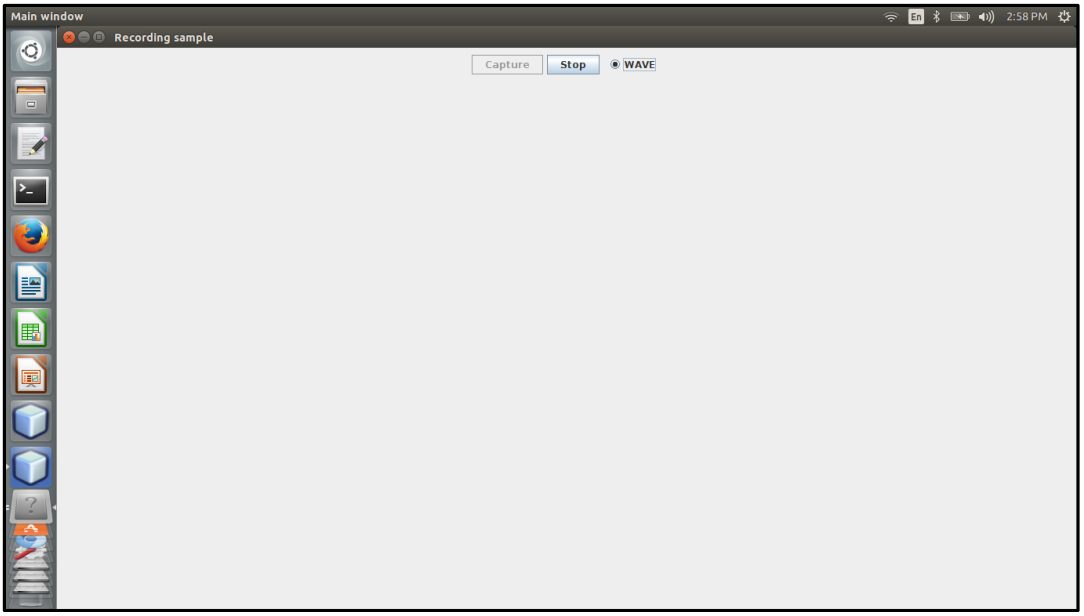


Figure A.2: audio recording progress window

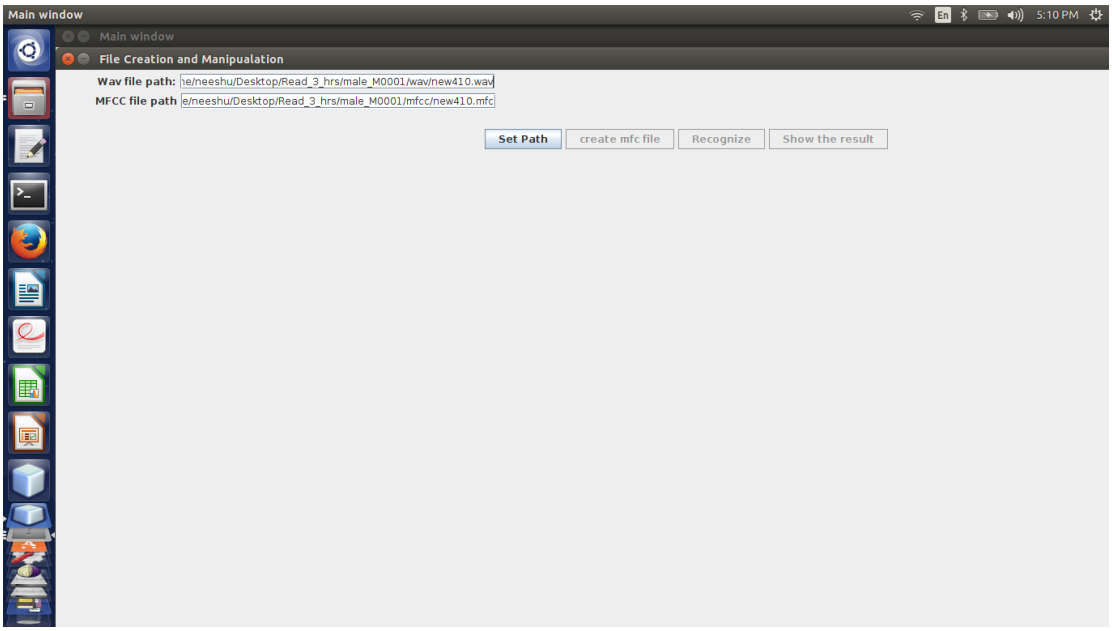


Figure A.3: Set path window for MFCC creation

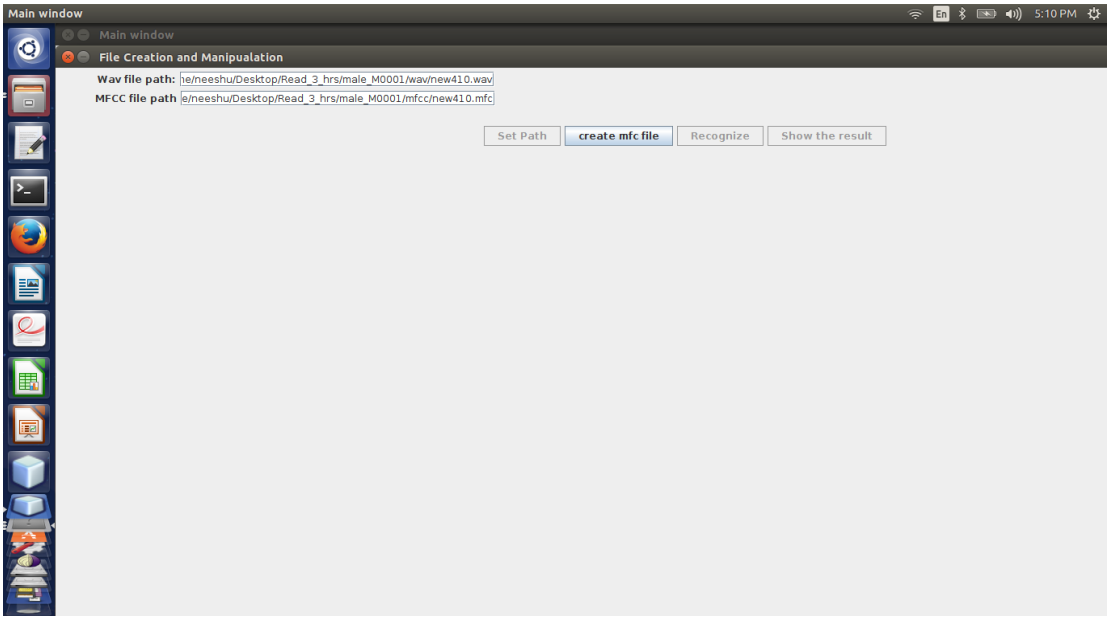


Figure A.4: MFCC generation window

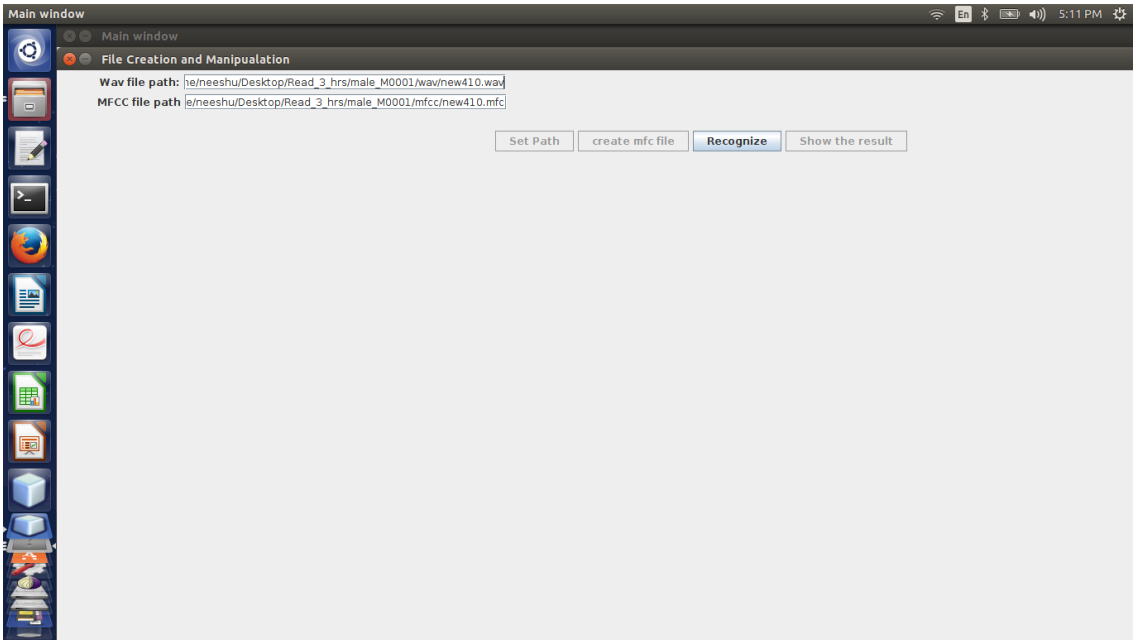


Figure A.5: MFCC done window

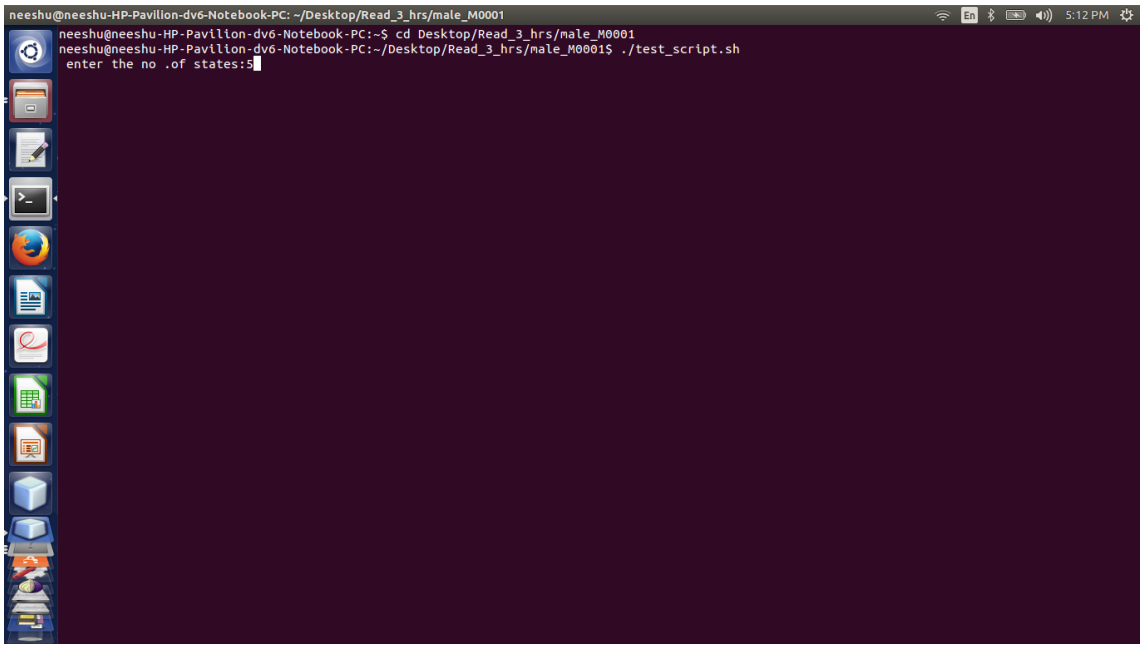


Figure A.6: Test script execution window

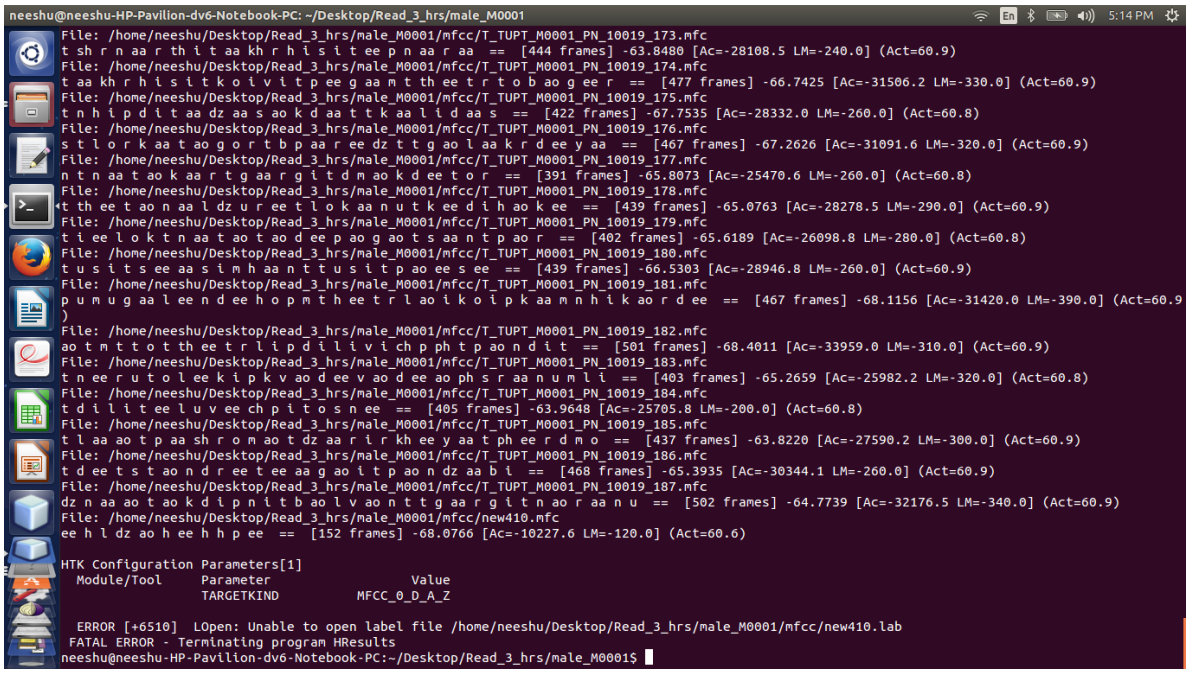


Figure A.7: Script running done window

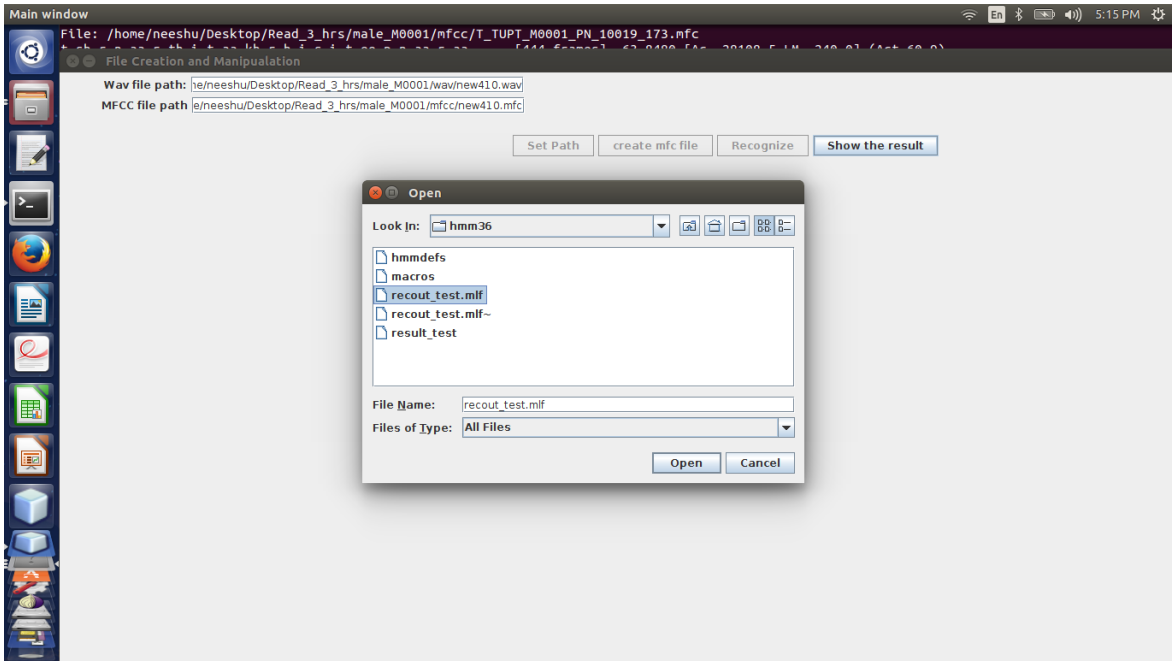


Figure A.8: Show results option window

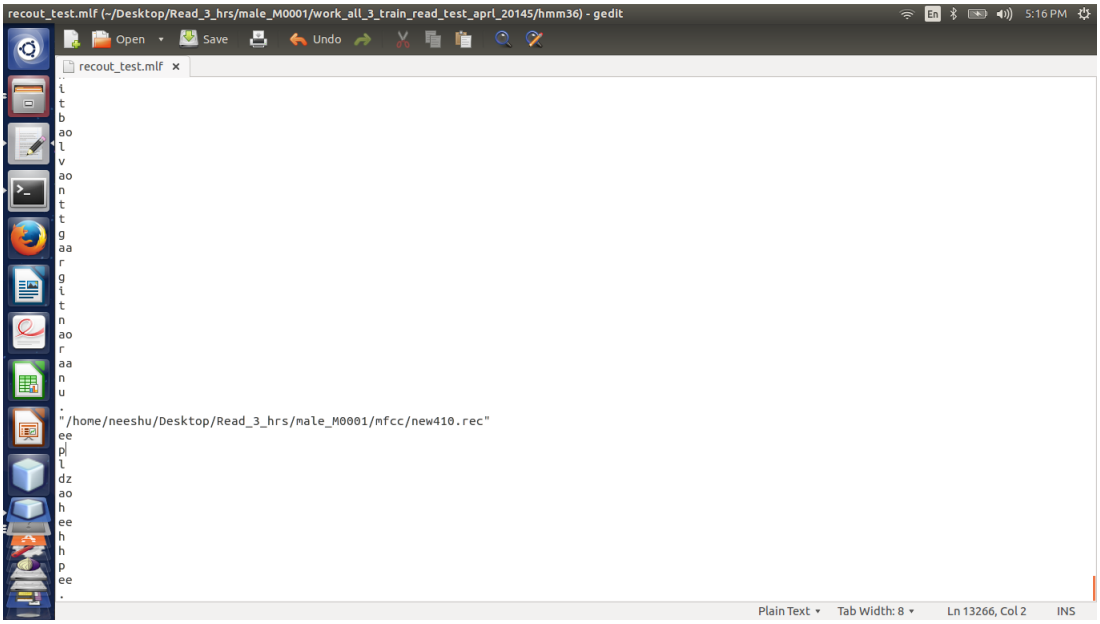


Figure A.9: Script result window