B.Sc. in Computer Science and Engineering Thesis

Automatic Word Recognition for Bangla Spoken Language

Submitted by

Sara Binte Zinnat
201014045

Md. Imamul Hossain
201014049

Razia Marzia Asheque Siddique
201014060

Supervised by

Dr. Mohammad Nurul Huda
Professor, United International University(UIU)

Department of Computer Science and Engineering
Military Institute of Science and Technology

CERTIFICATION

This thesis paper titled **“Automatic Word Recognition for Bangla Spoken Language”**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering on December 2013.

Group Members:

Sara Binte Zinnat
Md. Imamul Hossain
Razia Marzia Asheque Siddique

Supervisor:

————————————-

Dr. Mohammad Nurul Huda
Professor
United International University(UIU)

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis paper is the outcome of the investigation and research carried out by the following students under the supervision of Dr. Mohammad Nurul Huda, Professor, United International University(UIU), Dhaka, Bangladesh.

It is also declared that neither this thesis paper nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

——————————————-

Sara Binte Zinnat
201014045

——————————————-

Md. Imamul Hossain
201014049

——————————————-

Razia Marzia Asheque Siddique
201014060

ACKNOWLEDGEMENT

We are thankful to Almighty Allah for his blessings for the successful completion of our thesis. Our heartiest gratitude, profound indebtedness and deep respect go to our supervisor Dr. Mohammad Nurul Huda, Professor, United International University(UIU), Dhaka, Bangladesh, for his constant supervision, affectionate guidance and great encouragement and motivation. His keen interest on the topic and valuable advices throughout the study was of great help in completing thesis.

We are especially grateful to the Department of Computer Science and Engineering (CSE) of Military Institute of Science and Technology (MIST) for providing their all out support during the thesis work.

Finally, we would like to thank our families and our course mates for their appreciable assistance, patience and suggestions during the course of our thesis.

Dhaka
December 2013

Sara Binte Zinnat
Md. Imamul Hossain
Razia Marzia Asheque Siddique

ABSTRACT

Automatic speech recognition (*ASR*) known as speech recognition is a computer technology that enables a device to recognize and understand spoken words, by digitizing the sound and matching its pattern against the stored patterns. In short, it is the conversion of spoken words to text. Currently available devices are largely speaker-dependent and can recognize discrete speech better than the normal (continuous) speech. In our research, we have used a system which is speaker independent (recognize speech of indefinite multiple people) and can detect continuous speech. Their major applications are in assistive for helping people in working around their disabilities.

Our proposed Bangla word system, based on *LF-25* is a new approach towards the field of Bangla ASR system. For this thesis work, we have prepared a Bangla word recognition system of Bangla *ASR*. Most of the Bangla *ASR* system uses a small number of speakers, but 40 speakers selected from a wide area of Bangladesh, where Bangla is used as a native language, are involved here. In the experiments, Mel-Frequency Cepstral Coefficients (*MFCCs*) and Local Features (*LFs*) are inputted to the Hidden Markov Model (*HMM*) based classifiers for obtaining word recognition performance.

Other than the traditional *MFCC* triphone model; a new method that have used LF based triphone model had been experimented to get better *ASR* performance. We used k-mean clustering for the proposed method. From the experimental results, word correct rate and word accuracy for male and female voices distinctly provide much better result for *LF-25* than *MFCC-38* as well as *MFCC-39*. So, our proposed system is in favor of gender independent fact. For male and female voices collectively, sometimes *MFCC-39* based model and sometimes *LF-25* based model shows better word accuracy and correct rate.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATION

AF	: Articulatory Features
AM	: Acoustic Model
ASR	: Automatic Speech Recognition
ATR	: Advanced Telecommunication Research Institute International
BPF	: Band Pass Filter
DCT	: Discrete Cosine Transform
DSR	: Distributed Speech Recognition
EM	: Expected Maximization
FFT	: Fast Fourier Transform
GMM	: Gaussian Mixture Model
GI	: Gender-Independent
HNN	: Hybrid Neural Network
HMM	: Hidden Markov Model
HL	: Hidden Layer
In/En	: Inhibition/Enhancement
BNAS	: Bangla Newspaper Article Sentences
LF	: Local Feature
LM	: Language Model
LR	: Linear Regression
MFCC	: Mel-Frequency Cepstral Coefficient
OL	: Output Layer
OOV	: Out-of-vocabulary
PCR	: Phoneme Correct Rate
PRA	: Phoneme Recognition Accuracy
SPINE	: Speech recognition In Noisy Environments
SNR	: Signal-to-noise Ratio

LIST OF SYMBOLS

π	: Initial state distribution.
Φ	: Probability distribution.
S	: State Sequence.

CHAPTER 1
INTRODUCTION

Almost all the major spoken languages in the world have Automatic speech recognition *(ASR)* systems, but for Bangla (can also be termed as Bengali) too little research has been performed. The lack of proper speech corpus is a major difficulty to research in Bangla *ASR*. To develop Bangla speech corpus to build a Bangla text to speech system [17] the lack of proper speech corpus is a major issue. However, this effort is a part of developing speech databases for Indian Languages, where Bangla is one of the parts and it is spoken in the eastern area of India (West Bengal and Kolkata as its capital). But in Bangladesh most of the natives of Bangla (more than two thirds) reside, Here bangla is the official language. Although both the countries have same written characters of Standard Bangla, there are some sound that are produced variably in different pronunciations of Standard Bangla. So, there is a need to do research on the main stream of Bangla *ASR*, which is spoken in Bangladesh.

Bangla ASR research or Bangla speech processing can be found in [12, 11, 16, 14, 22, 13]. For example, using *Hidden Markov Models (HMMs)* recognition of isolated and continuous Bangla speech on a small dataset is described in [11]; Bangla vowel characterization is done in [12]; development of Continuous Bangla speech recognition system is in [22], where [13] shows a brief overview of Bangla speech synthesis and recognition. As a whole, most of these works are mainly focused on the on the frequency distributions of different vowels and consonants or simple recognition task on a very small database.

1.1 Contribution

In this work, a medium speech corpus which is based on ASR systems is used for the designing of triphone models. Two stages comprises the method. To catch context of both sides, the first stage designs triphone models; the second stage use Hidden Markov Model based classifier to output word strings based on triphone models. The purpose of this research is to help to build a medium vocabulary triphone based continuous speech recognizer for Bangla language.

In Order to solution some problems in Bangla Speech Recognition, this thesis consentrates on context sensitive triphone model. The problems on which attention is focused are:

(a) Co articulation fact,
(b) Correction Rate of Word and Sentence,
(c) Reduction of mixture component for desired result.

1.2 Organization of the paper

Chapter 2: Exposition of the main purpose of this thesis as well as elucidation of the benifits and drawbacks of current *ASR*.
Chapter 3: Topical outline of Bangla Phonology and Bangla *IPA* Schema.
Chapter 4: Exploration of the establishment of the *HMM*-classifier Vitebri search and Baum-welch Algorithm.
Chapter 5: A brief discusssion on Context Dependent Triphone Models and basic concepts of Triphone Model.
Chapter 6: A discussion on feature extraction using MFCC and LF.
Chapter 7: A concept on building ASR system using Hidden Markov Model Toolkit(*HTK*).
Chapter 8: Illustrate the experiment data as well as environment and examine the experimental results.
Chapter 9: Concludes the paper.

CHAPTER 2
AUTOMATIC SPEECH RECOGNITION

Conversion of human voice into text is the purpose of Automatic Speech Recognition(*ASR*), which is also termed as Computer Speech Recognition. The task of translating speech is simplified if the voice of the speaker is properly recognized. The recognition systems that must be trained to a particular speaker is the main task of most speech recognition software which is referred to the term “Voice Recognition”.

Again in an expanded sense, the process of enabling a computer to identify and respond to the sounds produced in human speech without being targeted at single speaker such as live tv show on phone request can recognize random voices. This sense can be represented by the term Speech Recognition.

Speech recognition applications include voice user interfaces such as call routing (e.g. “I would like to make a collect call”), voice dialing (e.g. “Call home”), domotic appliance control, simple data entry(e.g. entering a credit card number), search (e.g. find a podcast where particular words were spoken), preparation of structured documents (e.g. a radiology report), speech-to-text processing (e.g. word processors or emails), and aircraft (usually termed Direct Voice Input).

2.1 History

While AT&T Bell Laboratories developed a primitive device that could recognize speech in the 1940s, researchers knew that the widespread use of speech recognition would depend on the ability to accurately and consistently perceive subtle and complex verbal input. In 1952 the first speech recognizer is developed and it comprises of a device to recognize single spoken digits [3]. Next in 1964 at New York World’s Fair, IBM Shoebox another early device was displayed.

Thus, in the 1960s, researchers turned their focus towards a series of smaller goals that would aid in developing the larger speech recognition system. As a first step, developers created a device that would use discrete speech, verbal stimuli punctuated by small pauses. However, in the 1970s, continuous speech recognition, which does not require the user to pause between words, began. This technology became functional during the 1980s and is still being developed and refined today.

Speech Recognition Systems have become so advanced and mainstream that business and health care professionals are turning to speech recognition solutions for everything from providing telephone support to writing medical reports. Technological advances have made speech recognition software and devices more functional and user friendly, with most contemporary products performing tasks with over 90 percent accuracy.

According to figures provided by industry. Satisfying the needs of consumers and businesses by simplifying customer interaction, increasing efficiency, and reducing operating costs, speech recognition is used in a wide range of applications.

In speech recognition automatic transcription, the constraint which is behind it's backwardness is the lacking in the software. The judgment that may be provided by a real human but not yet by an automated system is mostly required as the nature of narrative dictation is highly interpretive. In addition, the requirement of a long period of time to train the software by the user and/or system provider is another visible constraint in this context.

In *ASR* a comparison is made, to differentiate between "artificial syntax systems"and "natural language processing". The first types of systems stated above are usually domain-specific and the second type of processing stated above are basically language-specific application. Each of these types of application represents its own specific goals and challenges.

2.2 Basics of Speech Recognition

Speech recognition is the way to identify spoken words by using a computer (or other type of machine). In short, it is the interaction between human and computer with the purpose of making it correctly recognizes the words of human voice. A general solution of speech recognition shows in Figure 2.1.

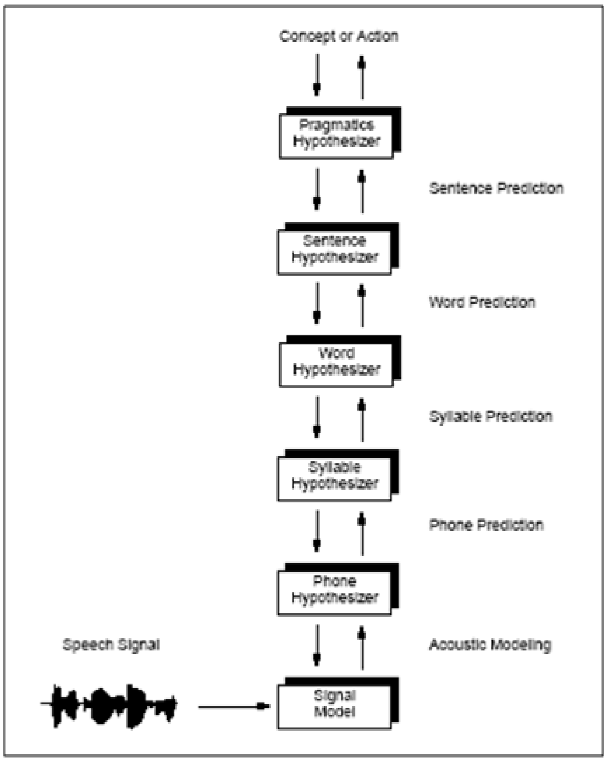


Figure 2.1: General Solution.

Some definitions which are the basics to understand the speech recognition technology are presented below:

Utterance

Vocal expression can be termed as utterance. It is the act or process of producing sounds with the voice that represents a single meaning to the computer. Utterances can be a single word, a few words, a sentence, or even multiple sentences.

Speaker Dependence

It is an Acoustic Model that has been tailored to recognize a particular person's speech. Such Acoustic Models are usually trained using audio from a particular person's speech.

A Speaker Independent Acoustic Model can recognize speech from a person who did not submit any speech audio that was used in the creation of the Acoustic Model.

The reason for the distinction is that it takes much more speech audio training data to create a Speaker Independent Acoustic Model than a Speaker Dependent Acoustic Model.

Vocabularies

Vocabularies (or dictionaries) are collection of words or utterances to be recognized by the *SR* system. Usually, larger vocabularies are more difficult to recognize than smaller vocabularies. Here, each entry doesnt have to be a single word. They can be as long as a sentence or two. Smaller vocabularies can have as few as 1 or 2 recognized utterances (e.g. "carry on"), while very large vocabularies can have a hundred thousand or more.

Accurate

By measuring the accuracy or well recognition utterance, the ability of a recognizer can be examined. This includes not only correct identification of an utterance but also the identification of spoken utterance if it is not in the recognizers vocabulary. Good *ASR* systems have an accuracy of 98% or more. The acceptable accuracy of a system really depends on the application.

Training

It is the process by which speech recognizer is taught the skills that are needed for the recognition of the speech of a speaker. An ASR system is trained by having the speaker repeat standard or common phrases and adjusting it's comparison algorithms to match that particular speaker. Training basically work for the improvement of accuracy of the recognizer.

Training can also be used by speakers that have difficulty speaking, or pronouncing certain words. As long as the speaker can consistently repeat an utterance, ASR systems with training should be able to adapt.

The speech recognition process is represented in Figure 2.2 and will be explained in more detail in the following sections.

The speech waveform first undergoes a signal processing step which produces a representation in spectral feature vectors. Phone likelihoods are subsequently estimated, after which a decoding step can finish the recognition process.

2.3 Types of Speech Recognition

With the description of what types of utterances recognizers have the ability to recognize, speech recognition systems can be separated in several different classes. These classes are based on the fact that one of the difficulties of ASR is the ability to determine when a speaker

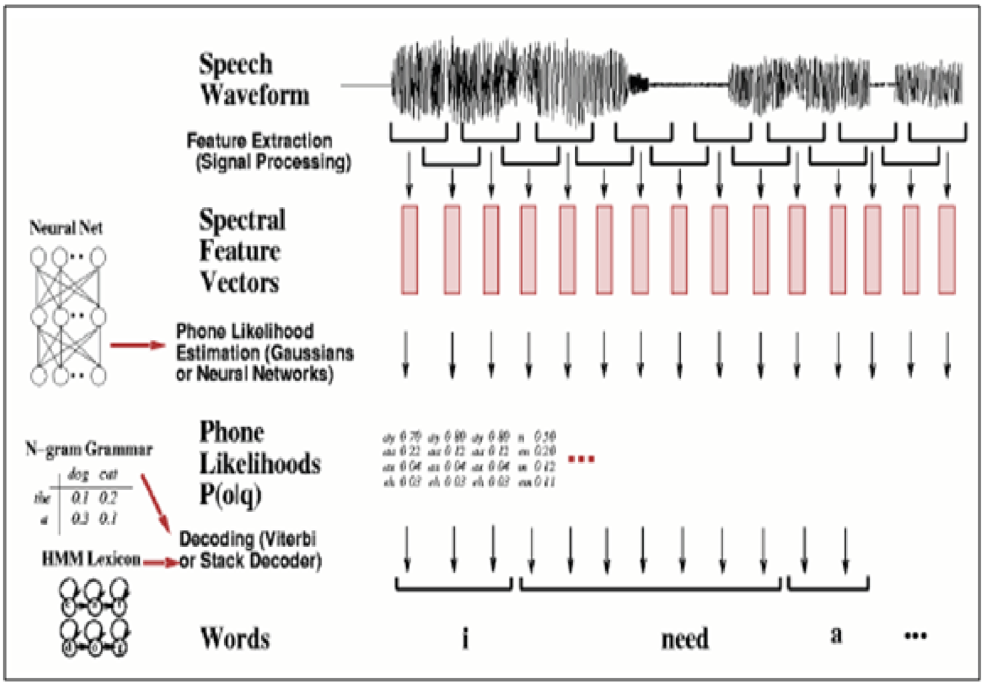


Figure 2.2: Schematic architecture for a simplified speech recognizer.

starts and finishes an utterance. Most packages can fit into more than one class, depending on which mode they're using.

Isolated Words

Isolated word recognizers usually require each utterance to have quiet (lack of an audio signal) on both sides of the sample window. It doesnt mean that it accepts single words, but does require a single utterance at a time. Often, these systems have "Listen/Not Listen"states, where they require the speaker to wait between utterances (usually doing processing during the pauses). Isolated Utterance might be a better name for this class.

Connected Words

Connect word systems or connected utterances are similar to Isolated words, but accept separate utterances to be 'uttered together'with a short pause between them.

Continuous Speech

Recognizers which has the capability with continuous speech are some of the most difficult to create because they must utilize special methods to determine utterance boundaries. Continuous speech recognizers allow users to speak almost naturally, while the computer determines the content. It can be suggested as computer dictation.

Spontaneous Speech

There appears to be a variety of definitions for what spontaneous speech actually is. At a basic level, it can be thought of as speech that is natural sounding and not rehearsed. An ASR system with spontaneous speech ability should be able to handle a variety of natural speech features such as words being run together, “ums”and “ahs”, and even slight stutters.

The recognition of spontaneous speech can be improved by taking into consideration the effects of the filled pauses while performing the recognition process by:

(1)Either deleting such pauses or by accepting them as words to be added to the dictionary of the *ASR* system.

(2) Recognizing hesitations and restarts.

(3) By developing the model accuracy at both the acoustic level and at the language model.

(4) Increasing the amount of training data and the lexicon size. This could reduce the error rate without increasing the search complexity.

At the end, for improving the performance of the existing recognizers it is needed to understand the properties of human auditory perception that are relevant for decoding the speech signal and to improve the performance of *ASR* in different environments is necessary. Also, using longer acoustic units (for example, syllables) instead of using short term speech segments followed by post processing techniques or using dynamic features is promising for the evolution of ASR. Moreover, rich prosodic cues (e.g. fundamental frequency, energy, duration, etc.) that permit successful understanding, which are ignored by state of the art ASR systems, must be considered for better performance. Again, the use of language independent acoustic models and variable ngram language models will enhance the performance further.

CHAPTER 3
BANGLA PHONETIC SCHEME

The International Phonetic Alphabet (IPA) is an alphabetic system of phonetic notation based primarily on the Latin alphabet. It was devised by the International Phonetic Association as a standardized representation of the sounds of spoken language. The IPA is used by foreign language students and teachers, linguists, speech pathologists and therapists, singers, actors, lexicographers, artificial language enthusiasts (conlangers), and translators.

The IPA is designed to represent only those qualities of speech that are distinctive in spoken language: phonemes, intonation, and the separation of words and syllables. To represent additional qualities of speech such as tooth gnashing, lisping, and sounds made with a cleft palate, an extended set of symbols called the Extensions to the IPA may be used.

IPA symbols are composed of one or more elements of two basic types, letters and diacritics. For example, the sound of the English letter t may be transcribed in IPA with a single letter, [t], or with a letter plus diacritics, [t^h], depending on how precise one wishes to be. Occasionally letters or diacritics are added, removed, or modified by the International Phonetic Association. As of 2008, there are 107 letters, 52 diacritics, and four prosodic marks in the IPA .

In 1886, a group of French and British language teachers, led by the French linguist Paul Passy, formed what would come to be known from 1897 onwards as the International Phonetic Association (in French, l' Association phontique internationale). Their original alphabet was based on a spelling reform for English known as the Romic alphabet, but in order to make it usable for other languages, the values of the symbols were allowed to vary from language to language.

Since its creation, the IPA has undergone a number of revisions. After major revisions and expansions in 1900 and 1932, the IPA remained unchanged until the IPA Kiel Convention in 1989. A minor revision took place in 1993, with the addition of four letters for mid-central vowels and the removal of letters for voiceless implosives. The alphabet was last revised in May 2005, with the addition of a letter for a labiodentals flap. Apart from the addition and removal of symbols, changes to the IPA have consisted largely in renaming symbols and categories and in modifying typefaces.

Extensions of the alphabet are relatively recent; “Extensions to the IPA”was created in 1990 and officially adopted by the International Clinical Phonetics and Linguistics Association in 1994.

3.1 Bangla Script

The Bengali script (Bengali: bangla lipi) is the writing system for the Bengali language. It is also used, with some modifications, for Assamese, Meitei, Bishnupriya Manipuri, Kokborok, Garo and Mundari languages. All these languages are spoken in the eastern region of South Asia. Historically, the script has also been used to write the Sanskrit language in the same region. From a classificatory point of view, the Bengali script is an abugida, i.e. its vowel graphemes are mainly realized not as independent letters like in a true alphabet, but as diacritics attached to its consonant graphemes. It is written from left to right and lacks distinct letter cases. It is recognizable by a distinctive horizontal line running along the tops of the letters that links them together, a property it shares with two other popular Indian scripts: Devanagari (used for Hindi, Marathi and Nepali) and Gurumukhi (used for Punjabi). The Bengali script is, however, less blocky and presents a more sinuous shaping.

The Bengali script evolved from the Eastern Nagari script, which belongs to the Brahmic family of scripts, along with the Devanagari script and other written systems of the Indian subcontinent. Both Eastern Nagari and Devanagari were derived from the ancient Nagari script. In addition to differences in how the letters are pronounced in the different languages, there are some minor typographical differences between the version of the script used for Assamese and Bishnupriya Manipuri as well as Maithili languages, and that used for Bengali and other languages.

The Bengali script was originally not associated with any particular language, but was often used in the eastern regions of Medieval India. It was standardized and modernized by Ishwar Chandra under the reign of the British East India Company. The script was originally used to write Sanskrit, which for centuries was the only written language of the Indian subcontinent in addition to Tamil. Epics of Hindu scripture, including the Mahabharata or Ramayana, were written in older versions of the Bengali script or Mithilakshar/Tirhuta script in this region. After the medieval period, the use of Sanskrit as the sole written language gave way to Pali, and eventually to the vernacular languages we know now as Maithili, Bengali, and Assamese. Srimanta Sankardeva used it in the 15^{th} and 16^{th} centuries to compose his oeuvre in Assamese and Brajavali the language of the Bhakti poets. There is a rich legacy of Indian literature written in this script, which is still occasionally used to write Sanskrit today.

3.2 Bangla IPA Table

The first IPA chart was prepared in 1888 by the earliest form of the International Phonetic Association and it has gone through many changes since then. The 1888 chart was rather a list of symbols and their descriptions.

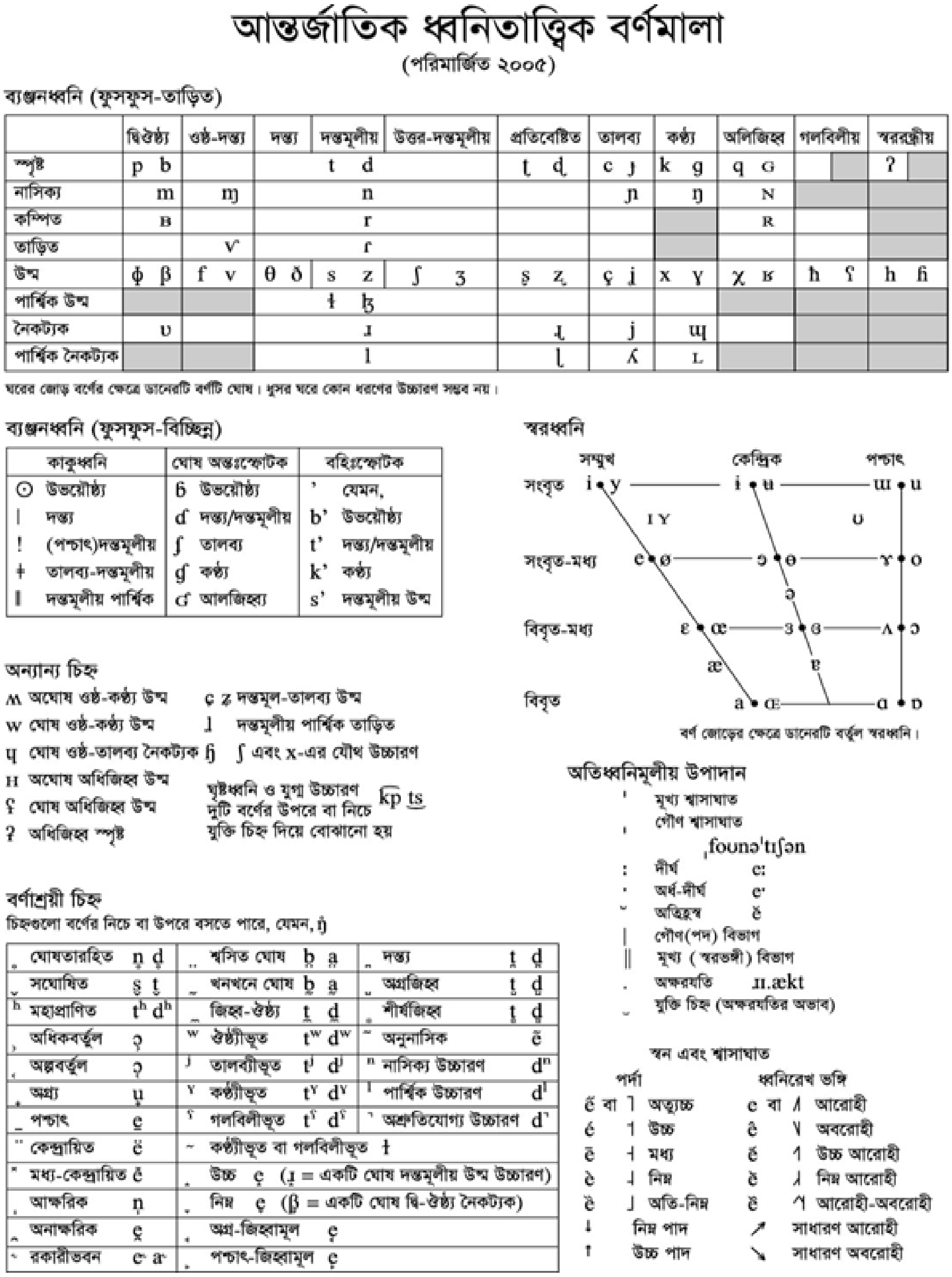
The latest, revised 2005, in Bengali shown in figure 3.1.

3.3 Bangla Phoneme Schemes

The Phonetic inventory of Bangla consists of 14 vowels, including seven nasalized vowels, and 29 consonants. Native Bangla words do not allow initial consonant clusters: the maximum syllable structure is CVC (i.e. one vowel flanked by a consonant on each side). Sanskrit words borrowed into Bangla possess a wide range of clusters, expanding the maximum syllable structure to CCCVC. English or other foreign borrowings add even more cluster types into the Bangla inventory.

Table 3.1 lists some Bangla words with their written forms and the corresponding IPA.

In the Bengali script, clusters of consonants are represented by different and sometimes quite irregular characters; thus, learning to read the script is complicated by the sheer size of the full set of characters and character combinations, numbering about 350. While efforts at standardizing the script for the Bengali language continue in such notable centers as the Bangla Academies (unaffiliated) at Dhaka (Bangladesh) and Kolkata (West Bengal, India), it is still not quite uniform as yet, as many people continue to use various archaic forms



আন্তর্জাতিক ধ্বনিতাত্ত্বিক বর্ণমালা

(পরিমার্জিত ২০০৫)

ব্যঞ্জনধ্বনি (ফুসফুস-তাড়িত)

	দ্বিঔষ্ঠ্য	ওষ্ঠ-দন্ত্য	দন্ত্য	দন্তমূলীয়	উত্তর-দন্তমূলীয়	প্রতিবেষ্টিত	তালব্য	কণ্ঠ্য	অলিজিহ্ব	গলবিলীয়	স্বরযন্ত্রীয়
স্পৃষ্ট	p b			t d		ʈ ɖ	c ɟ	k g	q ɢ		ʔ
নাসিক্য	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
কম্পিত	ʙ			r					ʀ		
তাড়িত		ⱱ		ɾ		ɽ					
উষ্ম	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
পার্শ্বিক উষ্ম				ɬ ɮ							
নৈকট্যক	ʋ			ɹ		ɻ	j	ɰ			
পার্শ্বিক নৈকট্যক				l		ɭ	ʎ	ʟ			

ঘরের জোড় বর্ণের ক্ষেত্রে ডানেরটি বর্ণটি ঘোষ। ধুসর ঘরে কোন ধরণের উচ্চারণ সম্ভব নয়।

ব্যঞ্জনধ্বনি (ফুসফুস-বিচ্ছিন্ন)

কাকুধ্বনি	ঘোষ অন্তঃস্ফোটক	বহিঃস্ফোটক
ʘ উভয়ৌষ্ঠ্য	ɓ উভয়ৌষ্ঠ্য	ʼ যেমন,
ǀ দন্ত্য	ɗ দন্ত্য/দন্তমূলীয়	bʼ উভয়ৌষ্ঠ্য
ǃ (পশ্চাৎ)দন্তমূলীয়	ʄ তালব্য	tʼ দন্ত্য/দন্তমূলীয়
ǂ তালব্য-দন্তমূলীয়	ɠ কণ্ঠ্য	kʼ কণ্ঠ্য
ǁ দন্তমূলীয় পার্শ্বিক	ʛ আলজিহ্ব্য	sʼ দন্তমূলীয় উষ্ম

অন্যান্য চিহ্ন

ʍ	অঘোষ ওষ্ঠ-কণ্ঠ্য উষ্ম	ɕ ʑ	দন্তমূল-তালব্য উষ্ম
w	ঘোষ ওষ্ঠ-কণ্ঠ্য উষ্ম	ɺ	দন্তমূলীয় পার্শ্বিক তাড়িত
ɥ	ঘোষ ওষ্ঠ-তালব্য নৈকট্যক	ɧ	ʃ এবং x-এর যৌথ উচ্চারণ
ʜ	অঘোষ অধিজিহ্ব উষ্ম		
ʢ	ঘোষ অধিজিহ্ব উষ্ম		
ʡ	অধিজিহ্ব স্পৃষ্ট		

ঘৃষ্টধ্বনি ও যুগ্ম উচ্চারণ দুটি বর্ণের উপরে বা নিচে যুক্তি চিহ্ন দিয়ে বোঝানো হয় k͡p t͜s

বর্ণাশ্রয়ী চিহ্ন

চিহ্নগুলো বর্ণের নিচে বা উপরে বসতে পারে, যেমন, ŋ̊

̥	ঘোষতারহিত	n̥ d̥	̤	শ্বসিত ঘোষ	b̤ a̤	̪	দন্ত্য	t̪ d̪
̬	সঘোষিত	s̬ t̬	̰	খনখনে ঘোষ	b̰ a̰	̺	অগ্রজিহ্ব	t̺ d̺
ʰ	মহাপ্রাণিত	tʰ dʰ	̼	জিহ্ব-ঔষ্ঠ্য	t̼ d̼	̻	শীর্ষজিহ্ব	t̻ d̻
̹	অধিকবর্তুল	ɔ̹	ʷ	ঔষ্ঠীভূত	tʷ dʷ	̃	অনুনাসিক	ẽ
̜	অল্পবর্তুল	ɔ̜	ʲ	তালব্যীভূত	tʲ dʲ	ⁿ	নাসিক্য উচ্চারণ	dⁿ
̟	অগ্র্য	u̟	ˠ	কণ্ঠ্যীভূত	tˠ dˠ	ˡ	পার্শ্বিক উচ্চারণ	dˡ
̠	পশ্চাৎ	e̠	ˤ	গলবিলীভূত	tˤ dˤ	̚	অশ্রুতিযোগ্য উচ্চারণ	d̚
̈	কেন্দ্রায়িত	ë	̴	কণ্ঠীভূত বা গলবিলীভূত ɫ				
̽	মধ্য-কেন্দ্রায়িত	e̽	̝	উচ্চ e̝ (ɹ̝ = একটি ঘোষ দন্তমূলীয় উষ্ম উচ্চারণ)				
̩	আক্ষরিক	n̩	̞	নিম্ন e̞ (β̞ = একটি ঘোষ দ্বি-ঔষ্ঠ্য নৈকট্যক)				
̯	অনাক্ষরিক	e̯	̘	অগ্র-জিহ্বামূল e̘				
˞	রকারীভবন	ɚ a˞	̙	পশ্চাৎ-জিহ্বামূল e̙				

স্বরধ্বনি

বর্ণ জোড়ের ক্ষেত্রে ডানেরটি বর্তুল স্বরধ্বনি।

অতিধ্বনিমূলীয় উপাদান

ˈ	মুখ্য শ্বাসাঘাত	
ˌ	গৌণ শ্বাসাঘাত	ˌfoʊnəˈtɪʃən
ː	দীর্ঘ	eː
ˑ	অর্ধ-দীর্ঘ	eˑ
̆	অতিহ্রস্ব	ĕ
|	গৌণ(পদ) বিভাগ	
‖	মুখ্য (স্বরভঙ্গী) বিভাগ	
.	অক্ষরযতি	ɹi.ækt
‿	যুক্তি চিহ্ন (অক্ষরযতির অভাব)	

স্বন এবং শ্বাসাঘাত

পর্দা			ধ্বনিরেখ ভঙ্গি		
e̋ বা	˥	অত্যুচ্চ	ě বা	˩˥	আরোহী
é	˦	উচ্চ	ê	˥˩	অবরোহী
ē	˧	মধ্য	e᷄	˦˥	উচ্চ আরোহী
è	˨	নিম্ন	e᷅	˩˨	নিম্ন আরোহী
ȅ	˩	অতি-নিম্ন	e᷈	˧˦˧	আরোহী-অবরোহী
↓		নিম্ন পাদ	↗		সাধারণ আরোহী
↑		উচ্চ পাদ	↘		সাধারণ অবরোহী

Figure 3.1: A chart of the full International Phonetic Alphabet.

Table 3.1: Some Bangla words with their orthographic transcriptions and IPA

English WORD	IPA Pronunciation	Our Symbol
AAMRA	/a m r a/	/aa m r ax/
AACHORON	/a tʃ r n/	/aa ch ow r aa n/

of letters, resulting in concurrent forms for the same sounds. Among the various regional variations within this script, only the Assamese and Bengali variations exist today in the formalized system. It seems likely that the standardization of the script will be greatly influenced by the need to typeset it on computers. The large alphabet can be represented, with a great deal of ingenuity, within the ASCII character set, omitting certain irregular conjuncts. Work has been underway since around 2001 to develop Unicode fonts, and it seems likely that it will split into two variants, traditional and modern.

In this and other articles on Wikipedia dealing with the Bengali language, a Romanization scheme used by linguists specializing in Bengali phonology is included along with IPA transcription. A recent effort by the government of West Bengal focused on simplifying Bengali spellings in primary school texts.

CHAPTER 4
HMM-BASED CLASSIFIER

4.1 About HMM

HMM phoneme models basically consists of three emitting states and a simple left-to-right topology as illustrated in Figs. 4.1 and 4.2 [21]. To ease joining of the models together the entry and exit states are provided. The exit state of one phoneme model can be connected with the entry state of another to form a composite *HMM*. This allows phone models to be joined together to form words and words to be joined together to cover complete utterances.

4.2 Modeling of HMM

Generally, an HMM is specified by a five-tuple [21] [9]:

(S, O, π, A, B)

(1) $S = \{1, 2, ..., N\}$, set of hidden states

N: the number of states.

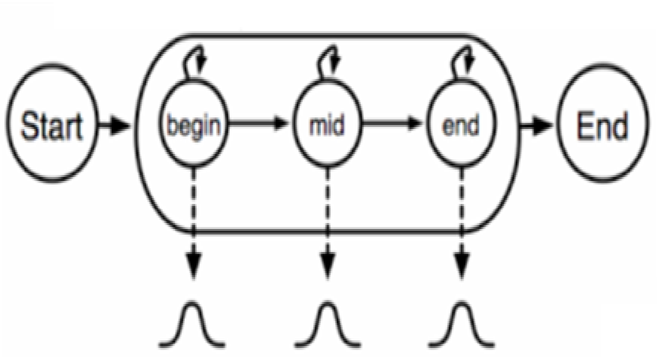


Figure 4.1: Standard HMM phoneme model (without Gaussian mixtures).

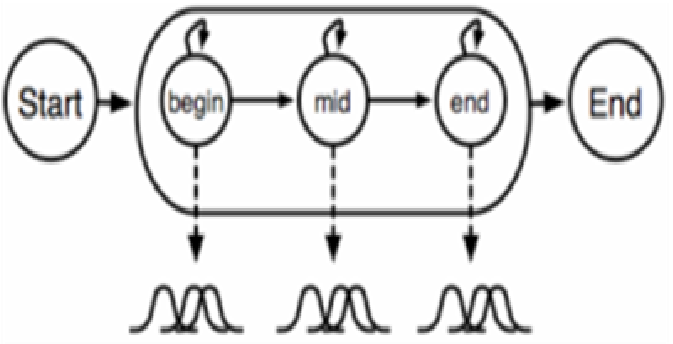


Figure 4.2: Standard HMM phoneme model (using Gaussian mixtures).

S_t: the state at time t.

(2) $O = \{o_1, o_2, ..., o_M\}$, set of observation symbols

M: the number of observation symbols

(3) $\pi = \{\pi_i\} \quad \pi_i = P(s_0 = i) \quad 1 \leq i \leq N$, the initial state distribution

(4) $A = \{a_{ij}\} \quad a_{ij} = P(s_t = j | s_{t-1} = i), \quad 1 \leq i, j \leq N$

State transition probability distribution.

(5) $B = \{b_j(k)\} \quad b_j(k) = P(X_t = o_k | s_t = j) \quad 1 \leq j \leq N, 1 \leq k \leq M$

Observation symbol probability distribution in state.

To sum up, a complete specification of an HMM includes:

(i) two constant-size parameters: N and M (representing the total number of states and the size of observation symbols)

(ii) three sets of probability distribution: $\Phi = (A, B, \pi)$

4.3 Three basic problems of HMM

HMM has an issue of facing three major problem. They are :

1. The evaluation problem : deals with the probability of the model.

2. The decoding problem : deals with the most likely state sequence.

3. The learning problem : deals with the adjustment of the model parameter to maximize the joint probability.

A brief definition of these problems are given below :

(i) The evaluation problem:

Given a model Φ and a sequence of observation $X = (X_1, X_2, ..., X_T)$, what is the probability $P(X|\Phi)$; i.e, the probability of the model that generates the observations?

(ii) The decoding problem:

Given a model Φ and a sequence of observation $X = (X_1, X_2, ..., X_T)$, what is the most likely state sequence $S = (s_0, s_1, ..., s_T)$ in the model that produces the observations?

(iii) The learning problem:

Given a model Φ and a set of observations, how can we adjust the model parameter to maximize the joint probability $\prod P(X|\Phi)$?

4.4 Solutions of the problems in HMM

As solutions to each of the three problems in HMM, three basic algorithms are used. These algorithms are :

1. Forward algorithm : define forward probability

2. Vitebri algorithm : define the best path

3. Baum-Welch algorithm : iteratively recomputes the model parameters to increase the likelihood of the training data at each iteration.

The above algorithms stated above are described below :

(i) First problem solution [Forward algorithm]

Define forward probability,

$\alpha_t(i) = P(X_1^t, s_t = i|\Phi)$

$\alpha_t(i)$ is the probability that the HMM is in state i having generated partial observation $X_1^t(namelyX_1, X_2, ..., X_t)$

(ii) Second problem solution: [Viterbi Algorithm]

Instead of summing up probabilities from different paths coming to the same destination state, the Viterbi algorithm picks and remembers the best path.

Define the best path probability,

$V_t(i) = P(X_1^t, S_1^{t-1}, s_t = i|\Phi)$

$V_t(i)$ is the probability of the most likely state sequence at time t, which generates the observation X_1^t (until time t) and ends in state i.

(iii) Third problem solution: [Baum-Welch Algorithm]

Baum-Welch Algorithm or so-called Forward-Backward algorithm in essence is an EM (Expectation Maximization) algorithm. The basic idea of EM algorithm is to iteratively recomputed the model parameters given their current estimates so as to increase the likelihood of the training data at each iteration.

CHAPTER 5
CONTEXT DEPENDENT TRIPHONE MODEL

Context-independent models or monophone models, assume that a phoneme in any context is equivalent to the same phoneme in any other context. Since the articulators do not move from one position to another immediately in most phoneme transitions this assumption is fundamentally proved incorrect. The transition duration is variable and depends on the phonemes, For example, for /v/ and /j/ the transitions are very long but from stop consonants to vowels the transition is significantly shorter but by no means discrete. Thus, neighboring phonemes are bound to have an effect on the examined phoneme. So, it should be considered that these co-articulation effects caused by context-dependency should be taken into account.

The terms precontext and postcontext are used to indicate the preceding and following neighbor of a phoneme. Figure 5.1 shows these terms for the triphone ch-aa+d.

Using word models is one way of dealing with the dependency. They are suitable for small vocabulary task and have been shown to be more accurate than phoneme models [2]. But for large recognizer they are not feasible. Instead, subword models are likely to produce better results.

In the past, a natural shift occurred from word models to syllable models. They were proposed already in the mid 70s by Fujimura [10]. The early syllable recognizers were non-HMM based. Now,demi-syllable and biphone models (a unit consisting of two consecutive

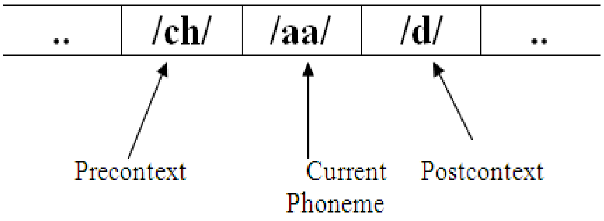


Figure 5.1: The definition of pre- and postcontext for the triphone ch-aa+d.

phonemes) have been used. Since phonemes are abstract super classes of phones, classification can be performed with different precision. As a result, some variations in the number of phonemes are caused. Syllable models have been popular especially in Mandarin, Cantonese, and Japanese recognizers but they are suitable for other languages as well. Diphones (phoneme units dependent on either pre- or postcontext) were introduced in the late 1970s and early 1980s [6, 24]. A few years after that, Schwartz et al. proposed the use of triphones in speech recognition [10]. This work concentrates on modeling triphones. This selection was made based on the fact that the most important coarticulatory effects on a phoneme are due to its immediate neighbors on either side. Moreover, triphones are commonly used in large vocabulary recognition in other languages, but not so much in Finnish.

5.1 Fundamental mechanism of triphones

To model the co-articulatory effects between phonemes and that there was nothing special about the units themselves, Schwartz [10] noted that all the subword units longer than phonemes (biphones, syllables, etc.) applied as units to speech recognition were merely trying. This motivated him to return to modeling phonemes. Only this time they were made context dependent, which led to the introduction of the concept of triphones - a model for a single phoneme conditioned on its preceding and following neighbor phoneme. The idea of triphones is used in any modern recognizer, and in [23] the actual results of the first recognizer utilizing triphones are presented. A triphone is simply a model of a single phoneme conditioned on its immediate neighbors, and not a structure of three phonemes. Similarly, a diphone is a model of a phoneme conditioned on either its left or right phoneme or a quinphone is conditioned on two neighboring phonemes on either side. Context-dependent models can be constructed in two ways: they can either be word internal or cross-word. For cross-word triphones the phonemes at the end or beginning of neighboring words are considered to affect the phoneme. On the contrary, when constructing word-internal models, context beyond the word borders are not considered. Usually, the number of cross-word triphones is considerably higher than the number of word-internal triphones.

The cross-word triphones are a natural choice for continuous speech recognition, since there are seldomly clear pauses between words in fluent speech. Actually, a stop consonant might introduce a longer pause than a word break. The problem, again, is the increasing number of models and the shortage of data for training them.

5.2 For context-dependent HMM's clustering mechanisms

In two cases a set of full triphones is immoderate. The first case arise in all practical cases there is not enough training material for many of the triphones. The second case is, despite co-articulatory effects some triphones are quite similar and had better be covered by the same model. With the minimization of the number of parameters the training data problem can be solved. It can be performed in several ways. The number of models or the number of states can be reduced by state or model clustering. Another approach is confining parameters inside the states or models, that is, forcing them to be equal for two different states (means and variances) or models (transition matrices). A straightforward way of reducing the number of parameters in a triphone model set could be to tie all the parameters of all models center states. The assumption that the center of each triphone (for the same phoneme) is similar could lead to this kind of an approach. Clustering mechanisms provide better results than this kind of direct tying. Some clustering algorithms are depicted in this section.

5.2.1 Data-driven (bottom-up) clustering

In the data driven clustering algorithm each state is initially located in it's respective cluster. Next, two clusters which form the smallest cluster are amalgamated together. This amalgamation of clusters is iteratively continued until the smallest cluster that would be constructed by combining any two clusters would be greater than some predefined limit.

The size of the cluster is represented as the longest distance between any two members of the cluster. The metric is termed as the Euclidean distance between the means of the two states. A constraint in this fact is its limitation to deal with invisible triphones (triphones not present in the training data), which are bound to occur in large vocabulary recognition with cross-word triphones. Therefore, diphone and monophone models are normally used to deal with this problem.

This algorithm is bottom-up since it starts with individual states and ends with clusters.

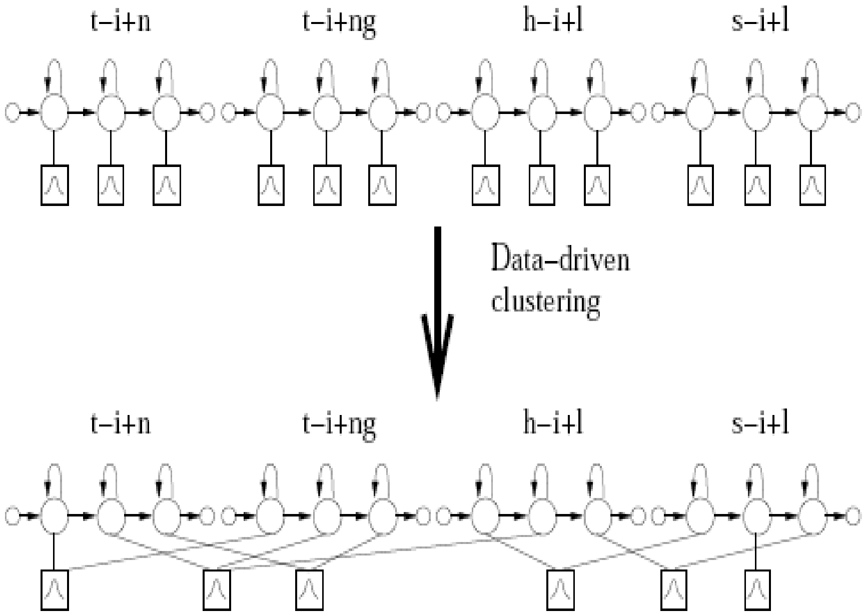


Figure 5.2: Data-driven state clustering for some triphones of the phoneme /i/.

An illustration of the algorithm is depicted in Figure 5.2. This clustering algorithm was introduced in [18].

5.2.2 Decision-tree (top-down) clustering

Binary decision trees [26] is another perspective for clustering states. Furthermore, to states and unlike data-driven clustering described above, this algorithm can be used to cluster entire models as well.

To split the clusters during the process a set of questions regarding phonemes context is needed. There is no specification regarding the number of questions. As an example a typical question might be : “Is the left context of this phoneme either an / a/ or an /o/?”.

A brief description of the algorithm stated below: initially in the root node of a tree, all states/models in a given list are placed. The nodes are iteratively split by selecting a question. Based on the answer, states/models in the state are placed either in the right or left child node of the current node. It is performed iteratively until the log likelihood increase of the states/models in the tree node obtained by the best question is below a predefined limit. At this stage, all the parameters in the state/model are tied. An illustration is in Figure 5.3.

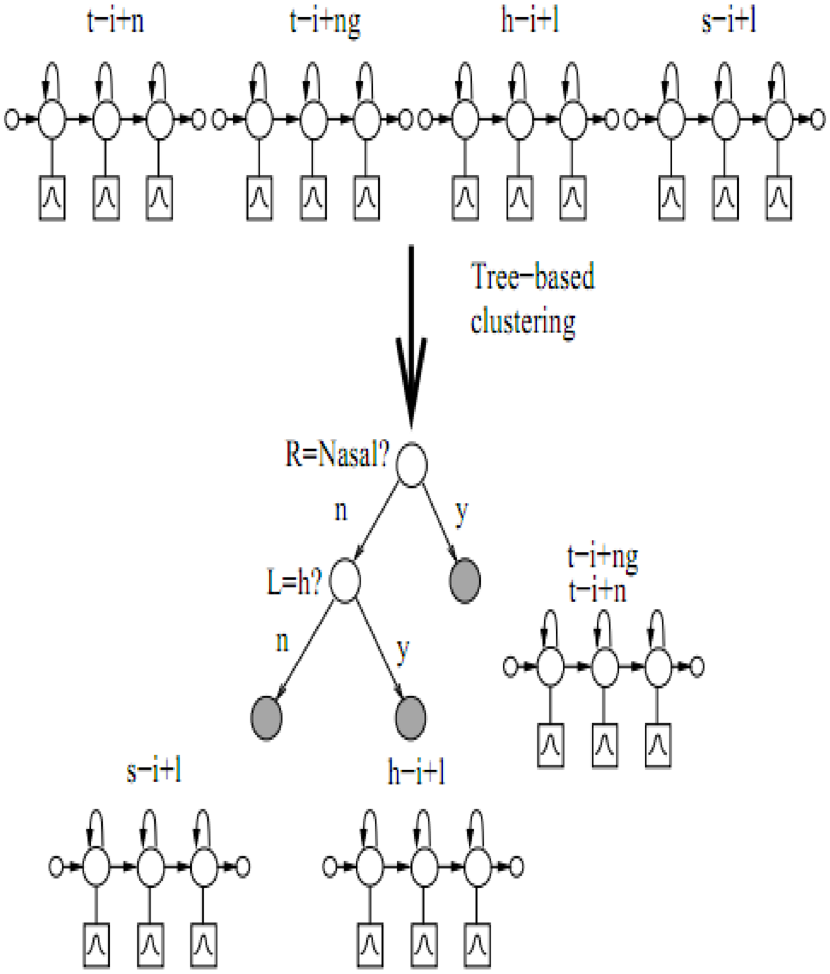


Figure 5.3: Part of the tree-based model clustering process of /i/-triphones. Leaf nodes are gray, and they form the final clusters.

The question used is chosen to maximize the likelihood of the training data given the final set of model/state tying. When the node is split, the likelihood of its child nodes is bound to increase since the number of parameters to describe the same data increases. The log likelihood can be calculated based on the statistics (means, variances, and state occupation counts) gathered from the training data, and based on that information the best question for each node can be chosen. Figure 5.3: Part of the tree-based model clustering process of /i/-triphones. Leaf nodes are gray, and they form the final clusters.

5.2.3 Classification based on articulatory facts

For the classification of triphone models one of the criterion is, to use decisions made a priori about the context for classifications. Basically, one decides classifications for phonemes and then classifies those triphones with contexts from the same phoneme classes to belong to the same broad class triphone (or cluster). The phoneme classes could be formed randomly but spontaneously it would be beneficial if there was some similarity between the members of a phoneme class. Furthermore, natural choices are based on articulatory facts. This type of approach has been suggested in [5] and [4]. Here, two different classifications were used: one is based on the type of the phoneme (short: ToP) and the other on the place of articulation (short: PoA). articulation (short: PoA).

CHAPTER 6
LF-BASED BANGLA ASR USING CONTEXT SENSITIVE TRIPHONE HMM

Automatic speech recognition (*ASR*) deals with the decoding of an acoustic signal of a speech utterance into corresponding text transcription, such as words, phonemes or other language units. Even after years of extensive research and development, accuracy in *ASR* remains a challenge to researchers. There are number of well known factors which determine accuracy. The prominent factors are those that include variations in context, speakers and noise in the environment. Therefore, research in *ASR* has many open issues with respect to small or large vocabulary, isolated or continuous speech, speaker dependent or independent and environmental robustness.

ASR for western languages like English and Asian languages like Chinese are well matured. But similar research in Bangla (widely used as Bengali) languages is still in its infancy stage. Another major hurdle in *ASR* for the Bangla language is resource deficiency. Annotated speech corpora for training and testing the acoustic models are scarce. Recently there is a growing interest in *ASR* for Bangla language [12, 11, 16, 14, 22, 13]. Continuous Bangla speech recognition system is developed in [14], while [8] presents a brief overview of Bangla speech synthesis and recognition. However, most of these researches have some problems: (i) deals with small scale speech corpus, (ii) use only time domain information and (iii) constructs triphone models [25, 19, 7, 15, 8] using *MFCC* features and consequently, better recognition performance is not obtained.

In this study, we have proposed an *ASR* system where information is based on time domain and frequency domain. The proposed method comprises three stages, where the first stage extracts phoneme probabilities from acoustic features, *LF*, the second stage designs triphone models to catch context of both sides and the last stage outputs word strings based on triphone models using hidden Markov model based classifier.

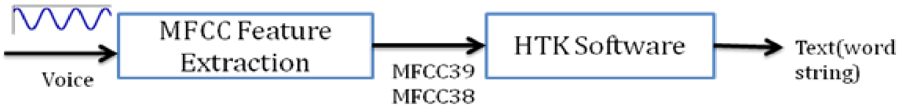


Figure 6.1: Conventional *MFCC* Based approach of word recognizer.

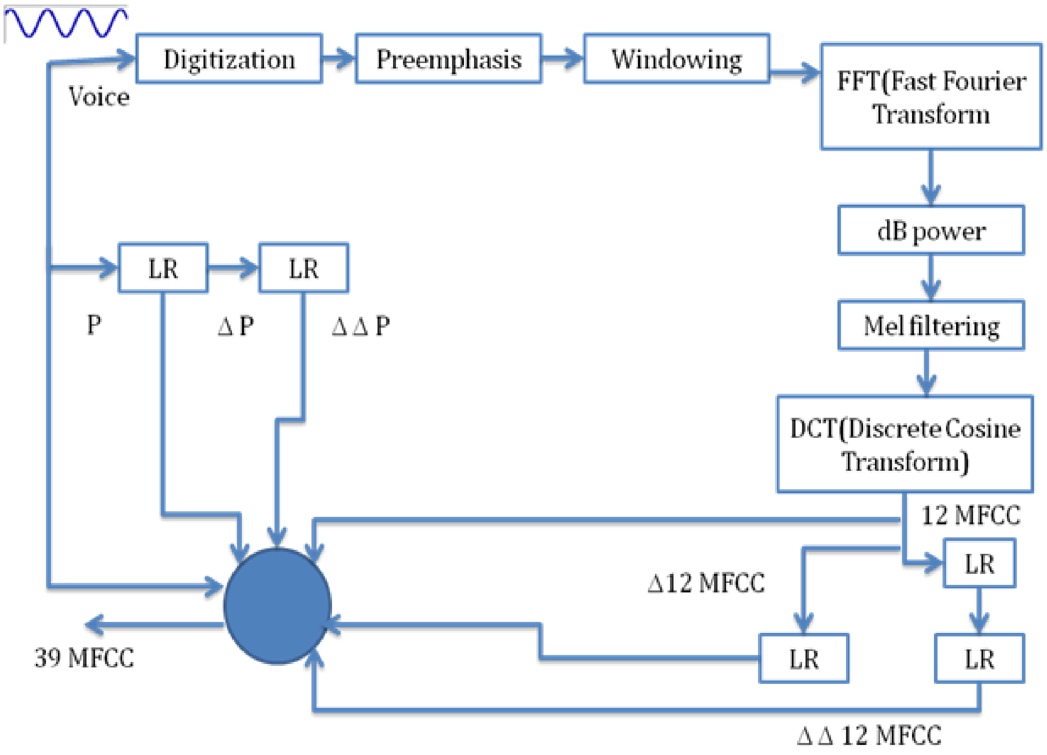


Figure 6.2: *MFCC* Feature Extraction.

6.1 Conventional MFCC Based Method

Traditional approach of *ASR* systems uses *MFCC* of 39 dimensions (*12-MFCC*, *12-ΔMFCC*, *12-ΔΔMFCC*, *P*, Δ*P* and ΔΔ*P*, where *P* stands for raw log energy of the input speech signal) as feature vector to be fed into a *HMM*-based classifier and the system diagram is shown in Figure 6.1. In Figure 6.2 *MFCC* feature Extraction is shown. Parameters (mean and diagonal covariance of hidden Markov model of each phoneme) are estimated, from *MFCC* training data, using Baum-Welch algorithm. For different mixture components, training data are clustered using the K-mean algorithm. Triphone models are configured using training data instead of monophone. During recognition phase, a most likely word for an input utterance is obtained using the Viterbi algorithm.

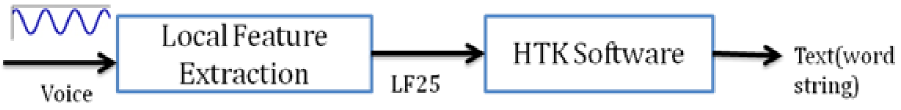


Figure 6.3: Proposed *LF* Based approach of word recognizer.

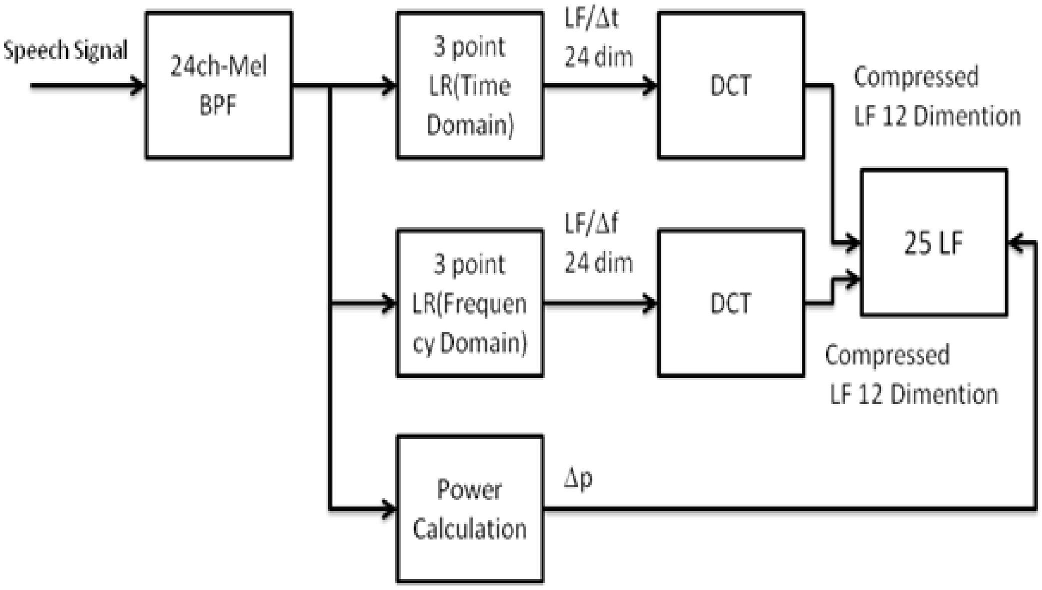


Figure 6.4: *LFs* extraction procedure.

6.2 Proposed LF Based Method

At the acoustic feature extraction stage, the input speech is first converted into *LFs* that represent a variation in spectrum along the time and frequency axes. Two *LFs* are then extracted by applying three-point linear regression (*LR*) along the time (t) and frequency (f) axes on a time spectrum pattern (TS), respectively. The following figure shows the *LF* extraction procedure. After compressing these two *LFs* with 24 dimensions into *LFs* with 12 dimensions using discrete cosine transform (*DCT*), a 25-dimensional (*12* Δt, *12* Δf, and ΔP, where P stands for the log power of a raw speech signal) feature vector called *LF* is extracted. Figure 6.3 shows the Local Feature based approach of word recognizer and figure 6.4 shows *LFs* extraction procedure.

CHAPTER 7
BUILDING THE RECOGNIZER WITH HTK

There are some steps which are used for feature extraction. This chapter deals with the steps, *HMM* building and training as well as testing. These steps are required by the *HTK* software package and for all stages, necessary *HTK* commands are shown. These are simplified forms of the actual commands used in this work. Here, debugging options were used for increased verbosity of the tools and the files resided in different directories making the actual commands very long. The functionality of the commands presented here is preserved. Many command lines option are available for all of the commands, some of which are the same for all of the tools. These are explained in table 7.1 and the options specific to each tool are explained when the tool is described. However, not all of the options are described here and further information may be found by consulting the *HTK* manual.

7.1 Feature extraction

The first task in building the recognition system was to calculate the cepstral coefficient files from the sound files. This process involves many different parameters. Feature extraction is accomplished by using the HCopy tool as follows:

```
HCopy -C hcopy.conf -S script.scp
```

Here, *hcopy.conf* defines the parameters to be used and *script.scp* contains simply a list of waveform files to process, one file per line. An example configuration file, used for standard MFCC calculation, is given below:

```
SOURCEFORMAT = WAV

TARGETKIND = MFCC_D_A_0

TARGETRATE = 100000.0
```

Table 7.1: Command line options common for all HTK tools

Command line option	Usage
-I <file.mlf>	<file.mlf> contains the (possibly segmented) labeling information for the files used in training/testing.
-i <file.mlf>	Output the recognized/edited labeling to <file.mlf>.
-C <file.conf>	Read configuration information from <file.conf>.
-M <dir>	Write the trained models to directory <dir>.
-H <macrofile>	Read HMM information from file <macrofile>.
-d <dir>	Read HMM information from separate files in directory <dir>.
-S <file.scp>	Instead of specifying the speech files used in training/recognition, read a list of them from the file <file.scp>.
-B	Save the models in binary instead of text files.

```
SAVECOMPRESSED = T
SAVEWITHCRC = T
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = F
```

7.2 Phoneme models

As the first stage of recognizer building, simple monophone models were built. Their creation process is explained in the following subsections.

7.2.1 Creating prototype models

At first, prototype models have to be created. They describe the structure of the models, while the actual values of the coefficients are unimportant. At this stage, the number of states and allowed transitions need to be determined and set. In this work, it was decided to use standard three state left-to-right HMM models. At an early stage of the tests, models with more states for long vowels and less for short vowels were tried, when trying to solve a problem where short vowels were intermingled with each other. However, this approach did not have any positive effect on recognition accuracy. The prototype models are equal for all phonemes and part of the prototype used in this work.

7.2.2 Initializing the prototype models

It is time to estimate more reasonable values for the prototype models after the prototype models for each phoneme are created. Two reasonable schemes are available in *HTK* for this. The first one is to use the tool HCompV, which calculates the global mean and variance of a set of training files and then uses these values for all models. It is referred to the flat start training scheme, as all the models receive the same values for the parameters. The other scheme is, the use of the tool HInit. It's principle relies on the idea of a *HMM* as a generator of speech vectors - it tries to find out which training data vectors were emitted by each state in the models and then calculates the means and variances for the corresponding states. This is done by continuously applying the Viterbi algorithm on the data and updating the parameters until the likelihood of each sentence falls below a defined value or a number of iterations has been reached. The HInit approach was chosen as it utilizes the segmentation information that was available and proved to provide slightly better results than the HCompv approach in preliminary tests. The number of iterations was chosen to be ten.

The initialization is made for each phoneme model separately with a command like

HInit -i 10 -I labels.mlf -l a -C hinit.conf -M hmm.0 -S train.scp\proto/a.

Here, the command line option -i specifies the maximum number of iterations, the option -l specifies the phoneme in question, and the final parameter proto/a specifies where the prototype file resides.

7.2.3 Model training

Model training is described by the following steps. HRest and HERest tools hepls to perform this. The working procedure of HRest is common to HInit. HRest expects the models to be initialized properly and uses the Baum-Welch algorithm instead of Viterbi training. This involves finding the probability of being in each state at each time frame using the forward backward algorithm. To form weighted averages for the *HMM* parameters this probability is then used. Whereas Viterbi training makes a hard decision as to which state each training vector was 'generated'by, Baum-Welch takes a soft decision. This can be helpful when estimating phone based *HMM*s since there are no hard boundaries between phones in real speech and using a soft decision may provide better results.

HERest, the tool for embedded training, simultaneously updates all of the *HMM*s in a system using all of the training data,As a result it is different from HInit and HRest. After loading the complete *HMM* definitions into memory, HERest processes each training file in turn. During this state, the segmentation information in the files is ignored and only the sequence of phonemes is of importance.

For each file, HERest constructs a composite HMM consisting of all the models corresponding to the phoneme sequence in the labeling of the utterance. The Forward-Backward algorithm is then applied as normal. When all of the training files have been processed, the new parameter estimates are formed from the weighted sums and the updated *HMM* set is output. In this work, HRest was called first for each phoneme:

HRest -I labels.mlf -v 0.001 -l a -C hrest.conf -M hmm.1 -S train.scp\hmm.0/a

The parameters are as in HInit except for -v, which sets the minimum variance to 0.001. This is necessary to avoid overtraining. After running HRest, HERest was called iteratively seven times:

HERest -C herest.conf -v 0.001 -d hmm.1 -M hmm.2 -I labels.mlf \ -t 250 150 1000 -S train.scp hmmlist

The option -t specifies the beam width in the backward phase of the Forward-Backward algorithm to 250. If pruning errors occur, the beam width is increased by 150, and the file in question is reprocessed using this new beam. This is repeated until no pruning error occurs or the upper limit for the beam - in this case 1000 - is reached. In the latter case the file is rejected and probably contains labeling errors.

7.2.4 Fixing the silence model

Pauses of different lengths and nature may occur in speech occurrence. There exists shorter periods of silence inside a sentence and longer at sentence borders, mainly at punctuation marks. The pause is very short or even nonexistent between most words. In stop consonants there is a short silent section, and sometimes there are glottal stops in different locations of speech. Moreover, presence of some background noise should be handled by some models. Due to this, there requires the necessity of various kinds of silence models.

In this work, two different silence models are used. The silence model sil was until now handled as any other model. Some modifications are made to it so that it will better absorb silences of different lengths, and another short pause model (sp) model is added for the word break silences.

For the sil model, transitions from the state 2 to state 4 and from state 4 to state 2 (from the first emitting state to the last and back) are added to allow the different states absorb the background noise without creating too many consecutive sil observations in the recognition results.

Additionally, a short pause model sp is created. It has only one emitting state (state number 2), which is tied (set equal) to the center state of the sil model. There is also a transition from the beginning state directly to the end state (from state 1 to state 3).

The sp model is useful in that it allows very short silence sections. For example, the period of silence between words in a sentence is very short and sometimes even nonexistent. At this point, the labeling was also changed so that in most places the sil labels were replaced by sp labels. Only at locations that suggest that there really is silence in the sentence, e.g. at punctuation marks, the sil label was preserved.

The sp model was created by copying the center state of the sil model and adding appropriate definitions for the rest of the model to it. Then, new transitions to the sil model were added and state 2 of the sp model was tied to state 3 of sil by the following HHEd script:

```
AT 2 4 0.2 {sil.transP}

AT 4 2 0.2 {sil.transP}

AT 1 3 0.3 {sp.transP}

TI silst {sil.state[3],sp.state[2]}
```

The needed HHEd command was

```
HHEd -C hhed.conf -d hmm.3 -M hmm.4 fixsil.hed hmmlist,
```

where fixsil.hed contains the lines above.

This silence model fixing was made between the second and third iteration of HERest. Therefore, for iterations three to seven the hmmlist should contain the short pause model as well. At this point the monophone models are ready. There are still many possibilities to improve their performance - adding mixtures being the most obvious - but since this thesis studies the use of triphones, no further attention was paid to developing the monophone models. The monophone models were used for comparison reasons for test data recognition. The recognition procedure is almost identical to the triphone case.

7.3 Triphone models

In theory, triphone models are clones of monophone models that have been renamed to include the left and right context of the phoneme and retrained with the occurrences of the triphone in question. As mentioned before, training a full triphone set is not feasible due to the large number of possible triphones and the small number of training examples for many of these in any realistic training data set. Therefore, the triphones need to be clustered. This section describes the steps required to create a triphone set from the monophones built above as well as different ways of clustering.

7.3.1 Making triphone labels from monophone labels

Initially, monophone label files need to be transformed into triphone form. That is, in the case of decision tree based triphone clustering, we want a phoneme string

sil h ae n sp s a n o i sil e tt ae sp h ... to be transformed into

sil+h sil-h+ae h-ae+n ae-n+s sp n-s+a s-a+n a-n+o n-o+i o-i+sil \ ***i-sil+e sil-e+tt e-tt+ae tt-ae+h sp ae-h+...***

If we want to cluster based on articulatory facts then the context of the triphones will be the broad class of the context phoneme. For example, instead of the triphone n-s+a we would have NA-s+BV, where NA denotes nasal and BV denotes a back vowel. As one can see from the example, the two kinds of silence are treated differently. The short pause is ignored when forming triphones. The sil label is considered as a normal context to other phonemes. Both of the silence models are in practice context-free, even though the contexts are included in the sil labels - all sil models are tied later on to one model. This approach was chosen because the short pauses, being short or even non-existent, do not usually affect the pronunciation of other phonemes, while the longer silence is usually realized in a longer pause in speech. Some recognizers consider the silence models as context-dependent while others do not, and the context-free approach was chosen for this work. The labels are transformed into triphone format by applying the following HLEd script to the label files.

NB sp

TC

Assuming that this script is saved into a file called mktri. led, it is applied to a label file with the command:

HLEd -C hled.conf -i new.mlf mktri.led new.mlf

This script seems to leave the silences in the very beginning and end without context. These need to be added to the labels by hand or a script so that recognition would work properly.

7.4 Recognizing the test data

HTK needs some kind of rules regarding allowed phoneme sequences in order to be able to recognize test utterances. Normally, these rules are provided by the lexicon and the language model, but since none were used in this work, the simplest possible rule set, a phoneme loop, was used. It states that any phoneme may follow any other without a limit for the phoneme string length.

This is defined by a network file such as:

```
$T1 = PROTHOM AALO;

$T2 = NOTUN BIDDUT UTPADON;

( SENT-START (($T1) — ($T2)) SENT-END )
```

Given appropriate configuration parameters, *HTK* expands the phonemes to match their context. This kind of approach requires that all possible logical triphones have a physical counterpart. That is, a line exists in the triphone list file for them. The network file above also defines that a sentence always has to begin and end with silence. This approach reduces the need of diphone models to only silence models. Furthermore, in this work, all sil-models are tied to a single physical model. The network file has to be transformed into a Standard Lattice Format (or SLF) file using the HParse tool:

```
HParse network lattice
```

After this, recognition is accomplished using the HVite tool:

```
HVite -C hvite.conf -i recout_test.mlf -w lattice -H \ hmm.5/newMacros -t 150 -p -30 -S
test.scp vocab tiedlist_rec
```

The “vocabulary” file vocab contains the “description pronunciation” for all words.

The -t 150 option enables beam searching such that any model whose maximum log probability token falls more than 300 below the maximum for all models is ignored in decoding. The -p -30 specifies the phoneme insertion log probability to be -30. If the value given with the -t option is too small, the most probable path through the states may not be found, resulting in all kinds of recognition errors. If the value is too large - or beam searching is disabled altogether by setting the option to 0 - recognition takes much time. A positive value for the option -p results in many falsely inserted phonemes while too small a value causes many phonemes to be deleted. Iterative methods were used to find suitable values for these options, and appropriate values depend on the model type. For monophone models the phoneme insertion penalty was -10 and the beam width was 300, while for triphone models the corresponding values were -30 and 120. The -p values were selected so that the number of insertion and deletion errors was roughly equal. A good value for the -t option was searched for by gradually decrementing it and looking for the smallest value with which the recognition performance still remained good.

7.5 Processing the output

Following the steps described in this chapter, the recognized transcription is produced into the file recout_test.mlf with one phoneme on each line. In addition to the phoneme name, its beginning and ending times are included as well as the score (total likelihood of the segment):

```
s #!MLF!#
“/data_dir/FILE3000.rec”
6300000 12600000 PORIKOLPONAY -5968.250000
12600000 17900000 EGIYE -4388.363770
17900000 24600000 BASTOBAYONE -5550.177246
24600000 30400000 PICHIYE -4365.925781
```

7.5.1 Alignment by HResults

HTK includes a performance analysis tool, called HResults. It compares the recognized label files to the correct reference files and calculates the number of insertion, deletion and substitution errors as well as accuracy and correctness figures. Furthermore, it can produce confusion matrices between phonemes and output the test and reference sentences time-aligned. By default, HResults does not print the time-aligned transcription for sentences with no errors. This was changed in order to simplify the further analysis process. Also, if the -e or -c flags are used to make two phonemes equivalent, the original phonemes are replaced in the output transcription. Both of these cases were undesired, and the behavior was modified by making some minor changes in the HResults source code. For utilizing the time-aligned transcription produced by HResults, a set of Perl scripts were written. They gather information about:

- Overall correctness and accuracy

- Sentence based results

- Confusion matrices

- Error chain lengths

- Overall results for the individual phonemes

- Results for the individual phonemes in different contexts

- Errors for all phonemes in different contexts

- Overall phoneme statistics

CHAPTER 8
EXPERIMENTS AND RESULTS

In order to achieve further insight into the performance of different HMM types the results were analyzed thoroughly. First, the overall insertion, deletion, and substitution errors were counted and the accuracy and correctness figures derived from these.

8.1 Performance figures

As explained earlier, there are three different types of recognition errors: substitution (S), deletion (D), and insertion errors (I). The total number of words is marked with N. The correctness (C) figure is calculated according to equation 8.1 and it describes the portion of words recognized correctly from all words.

$$\%WordCorrectRate = \frac{(N-S-D)}{N} \times 100\% \tag{8.1}$$

For Accuracy calculation it considers insertion(I). Equation for word accuracy can be written as follows:

$$\%WordAccuracy = \frac{(N-S-D-I)}{N} \times 100\% \tag{8.2}$$

For the case of Sentence correct rate it considers correctly recognized sentences(H) within total number of sentences(N) to calculate sentence correct rate.

$$\%SentenceCorrectRate = \frac{H}{N} \times 100\% \tag{8.3}$$

8.2 Speech DataBase

At present, a real problem to do experiment on Bangla phoneme *ASR* is the lack of proper Bangla speech corpus. In fact, such a corpus is not available in any of the existing literature. Therefore, we develop a medium size Bangla speech corpus, which is described below.

Hundred sentences from the Bengali newspaper “Prothom Alo”are uttered by 30 male speakers and 30 female speakers of different regions of Bangladesh. These male sentences (30×100) are used for training corpus and these female sentences are user for training corpus. On the other hand, different 100 sentences from the same newspaper uttered by 10 different male speakers (total 1000 sentences) and different 100 sentences from the same newspaper uttered by 10 different female speakers (total 1000 sentences) are used as test corpus. All of the speakers are Bangladeshi nationals and native speakers of Bangla. The age of the speakers ranges from 20 to 40 years. We have chosen the speakers from a wide area of Bangladesh.

Recording was done in a quiet room located at United International University (UIU), Dhaka, Bangladesh. A desktop was used to record the voices using a head mounted close-talking microphone. We record the voice in a place, where ceiling fan and air conditioner were switched on and some low level street or corridor noise could be heard.

Jet Audio 7.1.1.3101 software was used to record the voices. The speech was sampled at 16 kHz and quantized to 16 bit stereo coding without any compression and no filter is used on the recorded voice. For this study, stereo coding has been used in order to reduce the redundancy in stereo coding signals. One can achieve significant signal gains in stereo coding which can be utilized to either boost the quality of the reconstructed signal or to lower the bit rate whiling keeping the signal quality constant with respect to the original coder.

8.3 Experimental Setup

The frame length and frame rate are set to 25 ms and 10 ms (frame shift between two consecutive frames), respectively, to obtain acoustic features (MFCCs) from an input speech. MFCC comprised of 39 dimensional features. LFs are a 25-dimensional vector consisting of 12 delta coefficients along time axis, 12 delta coefficients along frequency axis, and delta coefficient of log power of a raw speech signal [20].

For designing an accurate continuous word recognizer, word correct rate (WCR) and word accuracy (WA) for test data set are evaluated using an HMM-based classifier. The train data set is used to design Bangla triphones HMMs with five states, three loops, and left-to-right models. Input features for the classifier are 39 dimensional MFCC. In the HMMs, the output probabilities are represented in the form of Gaussian mixtures, and diagonal matrices are used. The mixture components are set to one, two, four and eight.

To obtain the WCR and WA we have designed the following experiments for male , female and male+female test data sets:

(a) MFCC39+Triphone-HMM.

(b) MFCC38+Triphone-HMM.

(c) LF25+Triphone-HMM.

8.4 Experimental Results and Discussion

Figure 8.1 shows the comparison of *WCR* using male test data set among the systems, MFCC39+Triphone-HMM, MFCC38+Triphone-HMM and LF25+Triphone-HMM. On the other hand, Figure 8.2 gives corresponding *WA* for the methods investigated. It is also shown from this figure that similar types of results are obtained. Sentence correct rates (*SCRs*) for the investigated methods using the male test data set are shown in Figure 8.3.

The word recognition performance for the methods, MFCC39 + Triphone - HMM, MFCC38 + Triphone - HMM and LF25 + Triphone - HMM respectively using the male test data set are shown in Tables 8.1, 8.2 and 8.3. It is observed from the tables that highest number of recognized words, H by the methods (a), (c) and (b) are 3131, 3119 and 2898 at mixture components four, eight and four respectively for the total number of training words, 3290. Again, the lowest number of total deletions for the methods (c), (a) and (b) are 11, 34 and 52 in mixture component four, four and four respectively, which indicates that the method (c) exhibits the lowest deletions among the all methods investigated. Moreover, the methods,(b),(c) and (a) substitute 322, 156 and 125 words that represent lowest substitution at

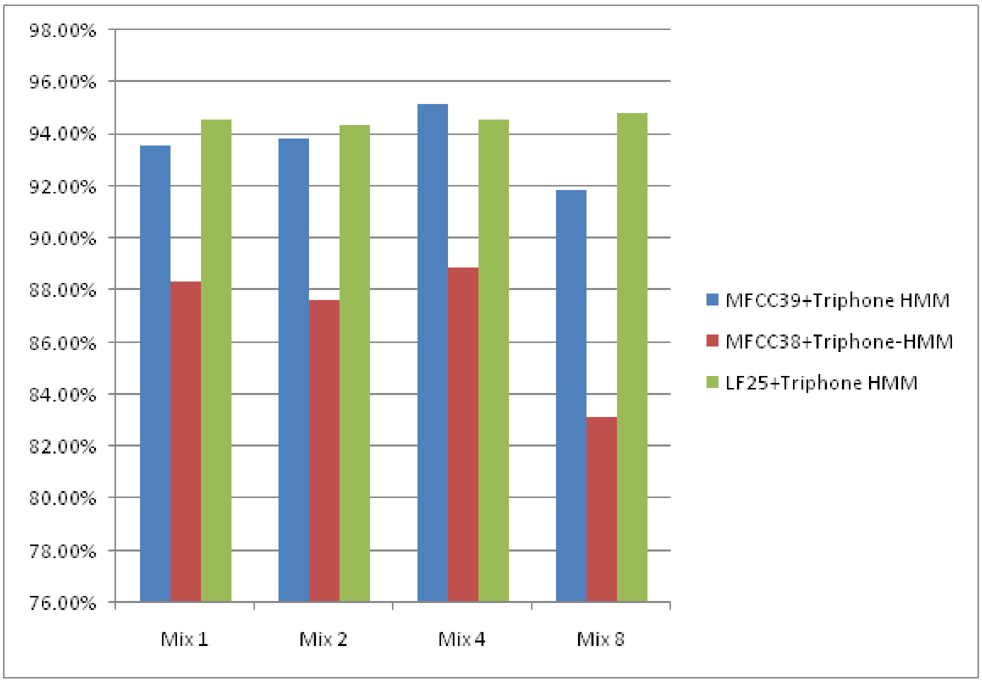


Figure 8.1: Word Correct Rate for male test data set.

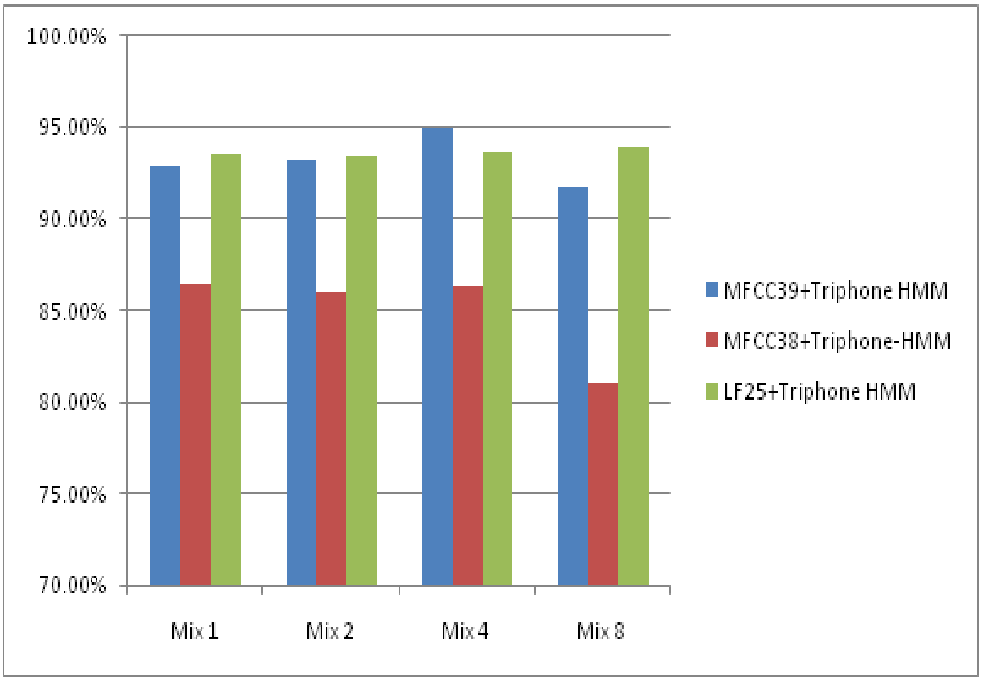


Figure 8.2: Word Accuracy for male test data set.

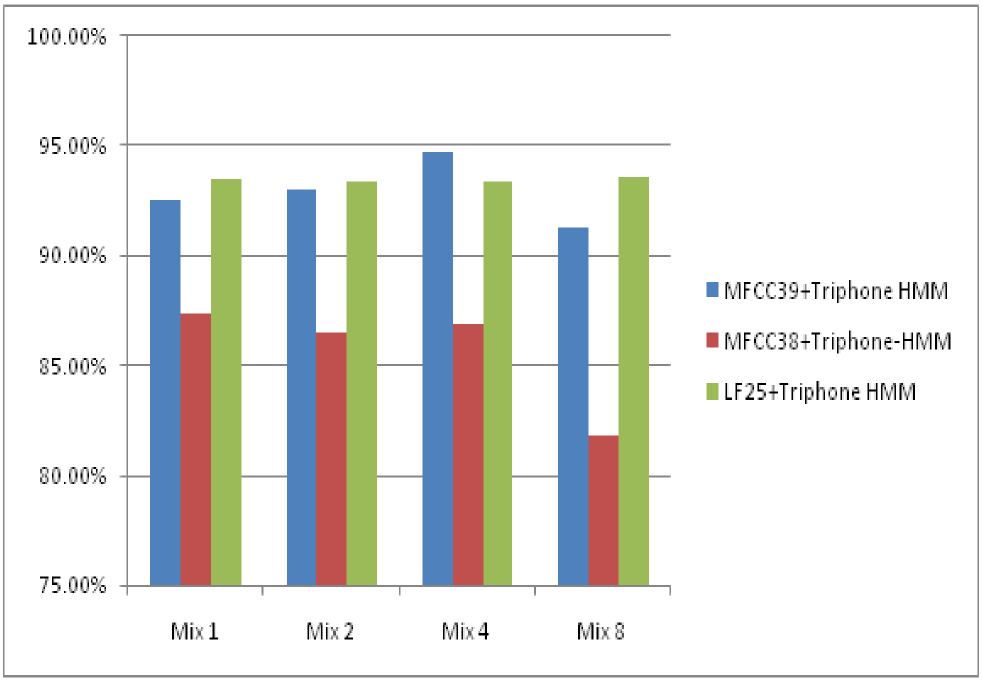


Figure 8.3: Sentence Correct rate for male test data set.

mixture component one ,eight and four respectively. Finally, the least number of insertions for the corresponding methods, which measures the word accuracy, are found 55, 29 and 7 for the mixture component one, eight and eight.

The sentence recognition performance for the methods (a), (b) and (c) are shown in tables 8.4, 8.5 and 8.6 respectively using the male test data set. The total numbers of correctly recognized sentences for the corresponding methods are 947 , 874 and 958 at mixture component four, one and eight out of 1000 test sentences, respectively. These recognized values indicate highest number of sentence correction rate among all the investigated mixture components.

WCRs using the female test data set for the methods, MFCC39 + Triphone - HMM, MFCC 38 + Triphone - HMM, LF25 + Triphone - HMM systems are depicted in Fig. 8.4. Beside the WCR, Fig. 8.5 illustrates WA for the corresponding methods investigated. Sentence correct rates (SCRs) for the investigated methods using the female test data set are shown in Fig. 8.6.

Word recognition performance for the methods, MFCC39 + Triphone - HMM, MFCC38 + Triphone - HMM and LF25 + Triphone - HMM using the female test data set are shown in Tables 8.7, 8.8 and 8.9. The methods,(c), (b) and (a) provide the lowest number of deletions,

Table 8.1: Word Recognition Performance for MFCC39 + TRIPHONE - HMM using Male test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	3079	3087	3131	3023
Deletion, D	37	40	34	63
Substitution, S	174	163	125	204
Insertion, I	23	21	8	7
Total, N	3290	3290	3290	3290

Table 8.2: Word Recognition Performance for MFCC38 + TRIPHONE - HMM using Male Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	2906	2883	2898	2736
Deletion, D	62	55	52	89
Substitution, S	322	352	340	465
Insertion, I	61	55	59	70
Total, N	3290	3290	3290	3290

Table 8.3: Word Recognition Performance for LF25+TRIPHONE-HMM using Male Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	3111	3104	3111	3119
Deletion, D	12	13	11	15
Substitution, S	167	173	168	156
Insertion, I	33	31	30	29
Total, N	3290	3290	3290	3290

Table 8.4: Sentence Recognition Performance for MFCC 39 + TRIPHONE - HMM using Male Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	925	930	947	913
Substitution, S	75	70	53	87
Total, N	1000	1000	1000	1000

Table 8.5: Sentence Recognition Performance for MFCC38 + TRIPHONE - HMM using Male Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	874	865	869	818
Substitution, S	126	135	131	182
Total, N	1000	1000	1000	1000

Table 8.6: Sentence Recognition Performance for LF 25 + TRIPHONE - HMM using Male Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	937	945	953	958
Substitution, S	63	55	47	42
Total, N	1000	1000	1000	1000

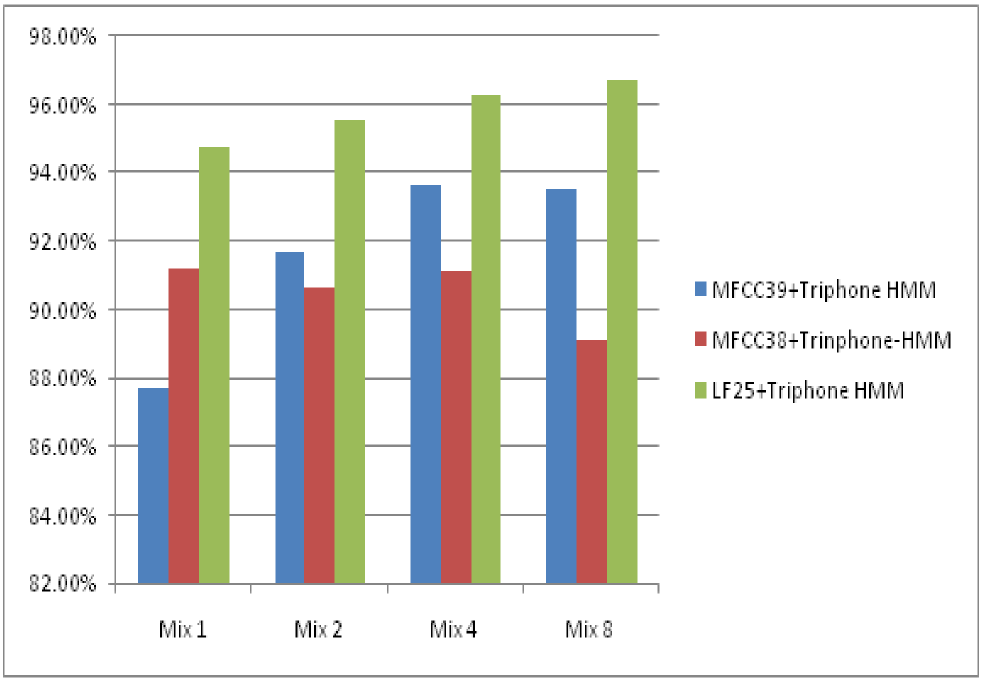


Figure 8.4: Word Correct Rate for female test data set.

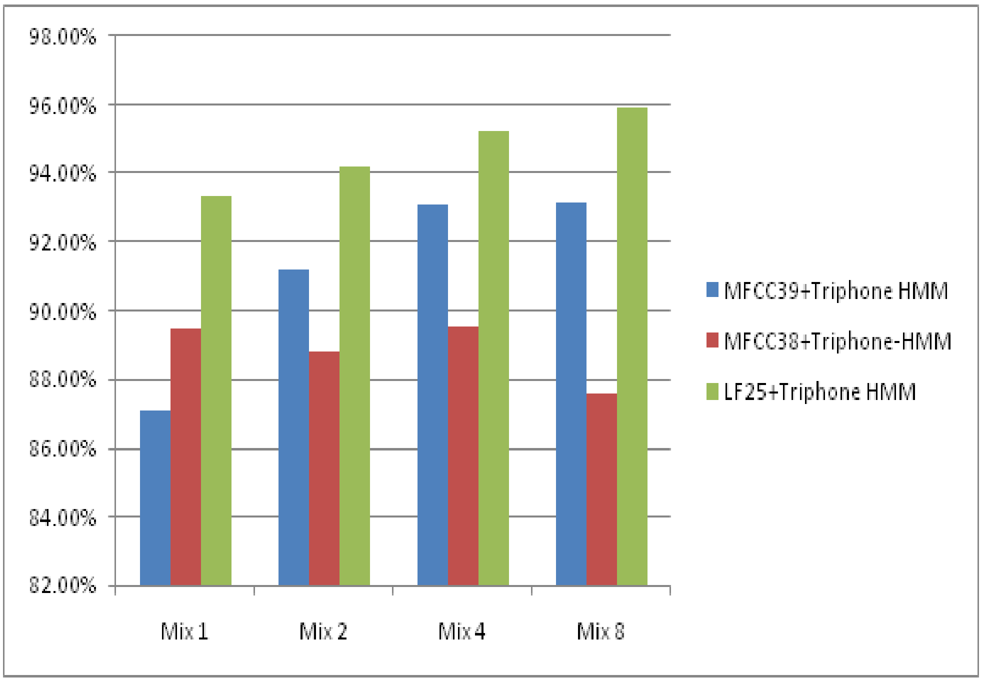


Figure 8.5: Word Accuracy for female test data set.

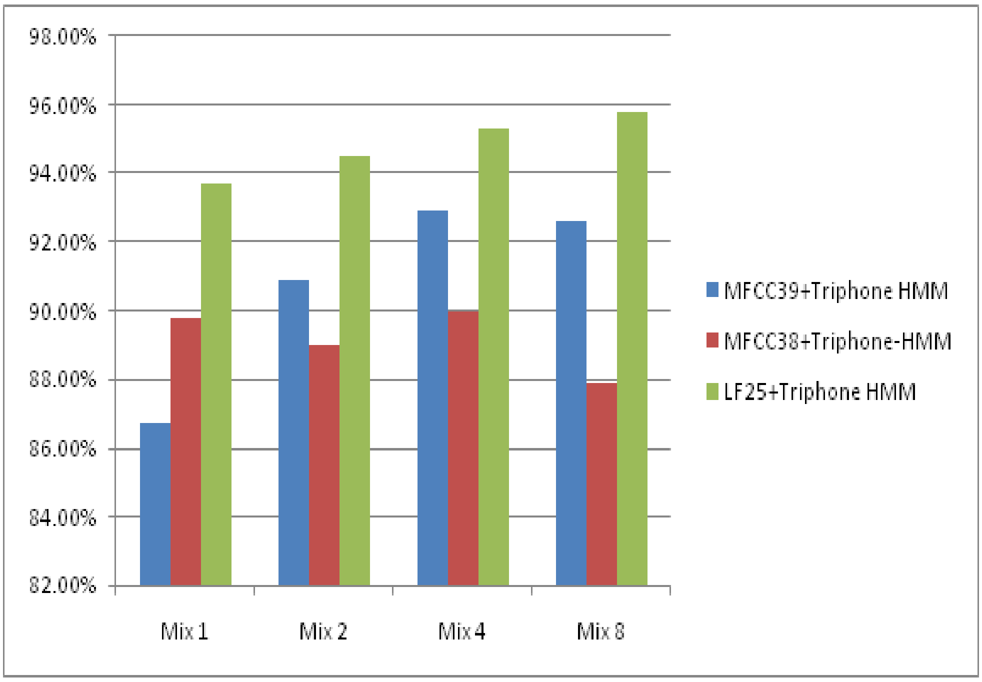


Figure 8.6: Sentence Correct Rate for female test data set.

11, 30 and 48 at mixture components four, one and four, respectively. On the other hand, the lowest substitutions, 95, 161 and 255 at mixture components eight, eight and one are produced by the methods (c), (a) and (b) respectively. Finally, the methods (a), (c) and (b) provide the lowest number of insertions, 11, 27, and 50 at mixture components eight, eight and eight respectively.

Sentence correct rates comparison among the methods, (a), (b) and (c) using the female test data set depict on table 8.10, 8.11 and 8.12. The highest number of correctly recognized sentences by the methods, (c), (a) and (b) are 958, 929 and 900 at mixture components eight, four and four out of 1000 test sentences, respectively.

WCRs using the male+female test data set for the methods, MFCC39 + Triphone - HMM, MFCC38 + Triphone - HMM, LF25 + Triphone - HMM systems are depicted in Fig. 8.7. Beside the WCR, Fig. 8.8 illustrates WA for the corresponding methods investigated. Sentence correct rates (SCRs) for the investigated methods using the female test data set are shown in Fig. 8.9.

Word recognition performance for the methods, MFCC39 + Triphone - HMM, MFCC38 + Triphone - HMM and LF25 + Triphone - HMM using male+female test data set are shown in Tables 8.13, 8.14 and 8.15. The methods,(c), (b) and (a) provide the lowest number of

Table 8.7: Word Recognition Performance for MFCC 39 + TRIPHONE - HMM using Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	2885	3017	3080	3076
Deletion, D	105	64	48	53
Substitution, S	300	209	162	161
Insertion, I	19	16	17	11
Total, N	3290	3290	3290	3290

Table 8.8: Word Recognition Performance for MFCC38 + TRIPHONE - HMM Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	3001	2982	2998	2932
Deletion, D	30	33	37	47
Substitution, S	259	275	255	311
Insertion, I	58	60	52	50
Total, N	3290	3290	3290	3290

Table 8.9: Word Recognition Performance for LF25 + TRIPHONE - HMM using Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	3118	3143	3167	3182
Deletion, D	18	14	11	13
Substitution, S	154	133	112	95
Insertion, I	47	43	33	27
Total, N	3290	3290	3290	3290

Table 8.10: Sentence Recognition Performance for MFCC 39 + TRIPHONE - HMM using Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	867	909	929	926
Substitution, S	133	91	71	74
Total, N	1000	1000	1000	1000

Table 8.11: Sentence Recognition Performance for MFCC38 + TRIPHONE - HMM using Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	898	890	900	879
Substitution, S	102	110	100	121
Total, N	1000	1000	1000	1000

Table 8.12: Sentence Recognition Performance for LF 25 + TRIPHONE - HMM using Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	937	945	953	958
Substitution, S	63	55	47	42
Total, N	1000	1000	1000	1000

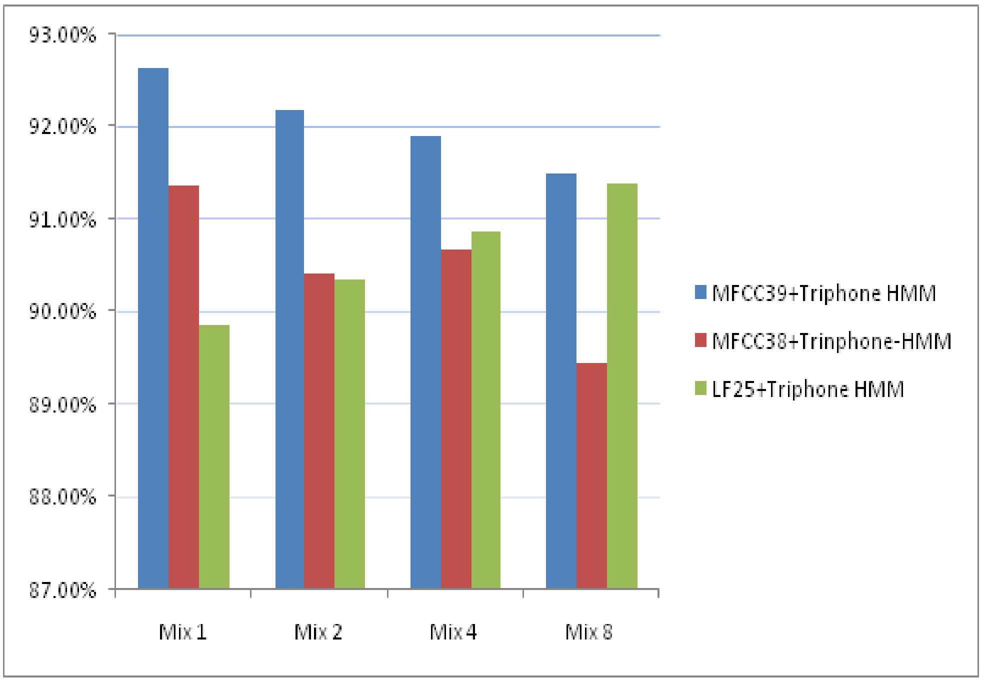


Figure 8.7: Word Correct Rate for male+female test data set.

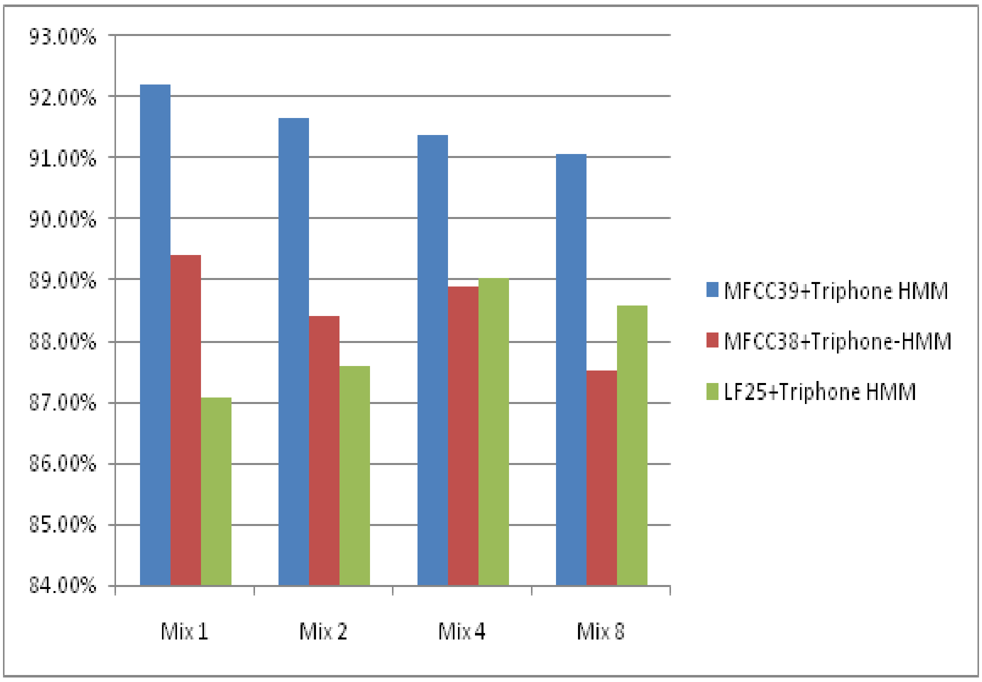


Figure 8.8: Word Accuracy for male+female test data set.

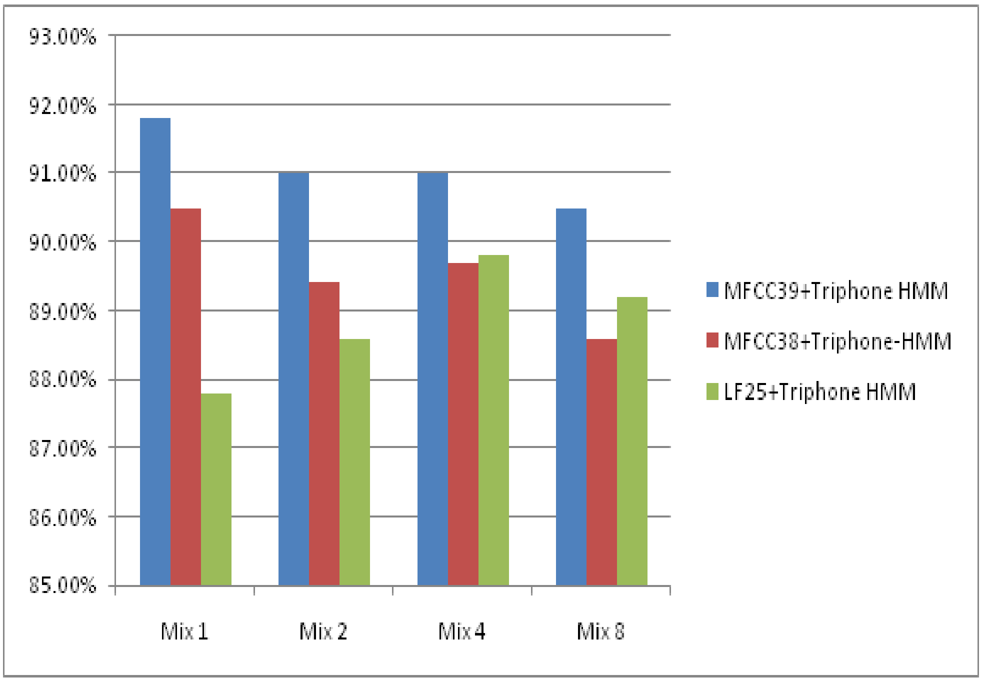


Figure 8.9: Sentence Correct Rate for male+female test data set.

deletions, 46, 62 and 106 at mixture components eight, one and two, respectively. On the other hand, the lowest substitutions, 364, 500 and 506 at mixture components one, four and one are produced by the methods (a), (c) and (b), respectively. Finally, the methods (b), (c) and (a) provide the lowest number of insertions, 116, 160 and 28 at mixture components four, eight and one respectively. All these are out of 6580 words.

Sentence correct rates comparison among the methods, (a), (b) and (c) using the male + female test data set depicted on Table 8.16, 8.17 and 8.18. The highest number of correctly recognized sentences by the methods, (c), (b) and (a) are 1796, 1810 and 1836 at mixture components four, one and one out of 2000 test sentences, respectively.

According to the experimental results, it is seen that the performance of *LF25* for male and female voice is best among the three procedure of feature extraction. For male+female voice the performance of *LF25* is less than or close to *MFCC-39*.But, these results may vary according to the type of mixtures. So, using *LF25* more accurate results can be achieved.

Table 8.13: Word Recognition Performance for MFCC39 + TRIPHONE - HMM using Male + Female test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	6096	6066	6048	6020
Deletion, D	120	106	110	140
Substitution, S	364	408	422	420
Insertion, I	28	36	36	28
Total, N	6580	6580	6580	6580

Table 8.14: Word Recognition Performance for MFCC38 + TRIPHONE - HMM using Male + Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	6012	5950	5966	5886
Deletion, D	62	74	78	78
Substitution, S	506	556	536	616
Insertion, I	128	132	116	128
Total, N	6580	6580	6580	6580

Table 8.15: Word Recognition Performance for LF25+TRIPHONE-HMM using Male + Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	5912	5946	6030	5990
Deletion, D	62	66	50	46
Substitution, S	606	568	500	544
Insertion, I	182	184	172	160
Total, N	6580	6580	6580	6580

Table 8.16: Sentence Recognition Performance for MFCC 39 + TRIPHONE - HMM using Male + Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	1836	1820	1820	1810
Substitution, S	164	180	180	190
Total, N	2000	2000	2000	2000

Table 8.17: Sentence Recognition Performance for MFCC38 + TRIPHONE - HMM using Male + Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	1810	1788	1794	1772
Substitution, S	190	212	206	228
Total, N	2000	2000	2000	2000

Table 8.18: Sentence Recognition Performance for LF 25 + TRIPHONE - HMM using Male + Female Test Data Set.

	Mix1	Mix2	Mix4	Mix8
Correctly Recognized, H	1756	1772	1796	1784
Substitution, S	244	228	204	216
Total, N	2000	2000	2000	2000

CHAPTER 9
CONCLUSION

The focus of this research was to show a preparation of Bangla speech corpus and provided some experiments to obtain better word recognition performance. This covered the basics of speech recognition up to the principles of isolated word recognition. The following conclusions are drawn from the experiments:

i) The *LF-25* based system provides tremendous improvement of Bangla word recognition accuracy for both training and test data.

ii) A higher Bangla word correct rate for training and test data is also obtained by the *LF*-based system.

iii) We have learned how the basic algorithm for isolated word recognition with *HMMs* works.

iv) We have learned how we can integrate additional knowledge sources into the recognition process.

v) K-mean clustering instead of model clustering. This would lead the clustering process even more towards the data-driven direction.

vi) We have learned how we can reduce the computational complexity of the search algorithm so that we are able to meet real time constraints.

Speaker independency is a major fact in many experimental applications. The methods presented in this thesis could also be applied to such a recognizer, but speaker independency poses many new problems, as well. The need for training data increases dramatically. As there exists very few research works for Bangla *ASR* system, still there prevails the need for conducting lot more research works to enrich this system. So, our research based on *LF-25* in the field of Bangla *ASR* system is a positive approach to achieve better word accuracy as well as word correct rate which will be highly beneficial to the future research analysis.

REFERENCES

[1] M. J. Alam, N. Uzzaman, and M. Khan. N-gram based statistical grammar checker for bangla and english. *Proc. of 9th International Conference on Computer and Information Technology (ICCIT 2006)*, 2006.

[2] L. R. Bahl, P. F. Brown, P. V. De Souza, and R. L. Mercer. Acoustic markov models used in the tangora speech recognition system. *In Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, 1988.

[3] K. H. Davies, R. Biddulph, and S. Balashek. Automatic speech recognition of spoken digits. *J. Acoust. Soc. Am.*, 1952.

[4] L. Deng, M. Lennig, V. N. Gupta, and P. Mermelstein. Modeling acousticphonetic detail in an hmm-based large vocabulary speech recognizer. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1988.

[5] A. M. Derouault. Context-dependent phonetic markov models for large vocabulary speech recognition. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1987.

[6] N. R. Dixon and H. F. Silverman. The 1976 modular acoustic processor (map). *IEEE Trans. Acoustics, Speech and Signal Processing*, 1977.

[7] S. Dupont, C. Ris, L. Couvreur, and J. Boite. A study of implicit and explicit modeling of coarticulation and pronunciation variation. *In Proc. of InterSpeech, Lisbon*, 2005.

[8] J. Ming et. al. Improved phone recognition using bayesian triphone models. *In Proc ICASSP*, 1998.

[9] X. D. Huang et. al. Hidden markov models for speech recognition. *Edinburgh: Edinburgh University Press*, 1990.

[10] O. Fujimura. Syllable as a unit of speech recognition. *IEEE Trans. Acoust, Speech Signal Processing*, 1975.

[11] M. A. Hasnat, J. Mowla, and M. Khan. Isolated and continuous bangla speech recognition: Implementation performance and application perspective. *In Proc. International Symposium on Natural Language Processing (SNLP)*, 2007.

[12] S. A. Hossain, M. L. Rahman, and F. Ahmed. Bangla vowel characterization based on analysis by synthesis. *Proc. WASET*, 2007.

[13] S. A. Hossain, M. L. Rahman, F. Ahmed, and M. Dewan. Bangla speech synthesis, analysis, and recognition: an overview. *In Proc. NCCPB, Dhaka*, 2004.

[14] A. K. M. M. Houque. Bengali segmented speech recognition system. *Undergraduate thesis, BRAC University, Bangladesh*, 2006.

[15] D. Jurafsky, W. Ward, Z. Jianping, K. Herold, Y. Xiuyang, and Z. Sen. What kind of pronunciation variation is hard for triphones to model? *In Proc. of ICASSP*, 2001.

[16] R. Karim, M. S. Rahman, and M. Z. Iqbal. Recognition of spoken letters in bangla. *In Proc. 5th International Conference on Computer and Information Technology (ICCIT02)*, 2002.

[17] S. P. Kishore, A. W. Black, R. Kumar, and R. Sangal. Experiments with unit selection speech databases for indian languages. *Carnegie Mellon University*.

[18] K.F. Lee. Context-dependent phonetic hidden markov models for speaker-independent continuous speech recognition. *IEEE Trans. Acoust. Speech Signal Processing*, 1990.

[19] J. Matousek, Z. Hanzlcek, and D. Tihelka. Hybrid syllable/triphone speech synthesis. *In Proc. of InterSpeech, Lisbon*, 2005.

[20] T. Nitta. Feature extraction for speech recognition based on orthogonal acoustic-feature planes and lda. *In Proc ICASSP*, 1999.

[21] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 1989.

[22] K. J. Rahman, M. A. Hossain, D. Das, T. Islam, and M. G. Ali. Continuous bangle speech recognition system. *In Proc. 6th International Conference on Computer and Information Technology (ICCIT03)*, 2003.

[23] R. M. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul. Context-dependent modeling for acoustic-phonetic recognition of continuous speech. *In Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1985.

[24] R. M. Schwartz, J. Klovstad, J. Makhoul, and J. Sorensen. A preliminary design of a phonetic vocoder based on a diphone model. *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1980.

[25] R. Thangarajan, A. M. Natarajan, and M. Selvam. Word and triphone based approaches in continuous speech recognition for tamil language. *WSEAS Transactions on Signal Processing*, 2008.

[26] S. J. Young, J. J. Odell, and P. C. Woodland. Tree-based state tying for high accuracy acoustic modelling. *In Proc. of ARPA Workshop on Human Language Technology*, 1994.