CERTIFICATE

This is to certify that the project report entitled "**Telugu Phoneme Based Tribal Language ASR system**", being submitted to University of Hyderabad by Banothu Rambabu (08MCMT30), in partial fulfillment for the award of Master of Technology in Computer Science, is a bonafide work carried out under the guidance of Ms. M. Nagamani, Lecturer (DCIS, UoH).

M. Nagamani,
Project Supervisor,
Department of CIS,
University of Hyderabad.

Prof. Arun Agarwal
Head of the Department,
Department of CIS,
University of Hyderabad.

Prof. T. Amaranath,
Dean,
School of MCIS,
University of Hyderabad.

ACKNOWLEDGEMENTS

I would like to thank my guide **M. Nagamani**, Lecturer, DCIS, University of Hyderabad for her guidance, valuable suggestions, support in every task of my work, and freedom in expressing my opinions during my project work.

I sincerely thank Prof. **T. Amarnath** Dean, MCIS and Prof**. Arun Agarwal**, Head, DCIS, University of Hyderabad for providing the facilities required to proceed with my work. I thank Prof. **S. Bapiraju**, M.Tech project coordinator, DCIS, University of Hyderabad for his schedule to complete the project.

I would like to thank Prof. Peri Bhaskarrao, Dr. Vijaya Bharadwaja Kumar, Ramya, Mounika, B.S.R Krishna, N. Rambabu, Ramesh, MCA friends and also rest of my friends who are recorded the data for their support which they given me in completion of my course.

I thank all the faculty members and AI lab staffs of DCIS, ES lab Instrument in charge P. Prasad Rao UOH for their support. I wish to thank them all for being very patient, understanding and helpful.

I would like to extend special thank to my **parents** and family members who gave me this position and providing me inspiration and moral support throughout my studies.

Banothu Rambabu

To

My Parents

And

Banjara Language Community

ABSRTACT

The Government of India presently focusing in the mother tongue based multilingual education to improve the literacy rate in tribal community. Growing IT field in the area of speech based applications give a scope to build speech interactive system in tribal language, as speech is the most common means of communication among the humans.

As almost all Tribal languages don't have a script for a spoken word, in our work the regional language (Telugu) script is adopted for representing those languages. Hence the proposed work "Telugu Phoneme based Tribal language Automatic Speech Recognition system" used for learning Banjara language, which is the native tribal language in the Andhra Pradesh State region.

A speaker independent Isolated Word Speech Recognition system has been developed for Banjara language. Hidden Markov Model (HMM) is used to model speech parameter. The required speech corpus has also been developed for the language. The GUI for Banjara to Telugu translation dictionary is developed using database and web technology concepts. Then developed IWR performance is by compared with lexicon of CMU i.e, English phonemes and UOH which uses Telugu Language Phonemes and found using UOH phonemes around 25 to 35% improvement in Recognition Accuracy for both native and non-native speakers.

Table of Contents

List of Figures

List of Tables

Abbreviation and Symbols

ASR	Automatic Speech Recognition
BFCC	Bark Frequency Cepstral Coefficients
BM	Basilar Membrane
DARPA	Defense Advanced Research Projects Agency
ERB	Equivalent Rectangular Bandwidth
FFT	Fast Fourier Transform
HMM	Hidden Markov Model
HTK	Hidden Markov Model Toolkit
IDFT	Inverse Discrete time Fourier Transform
IPA	International Phonetic Alphabet
ADC	Analog to Digital Conversion
MFCC	Mel Frequency Cepstral Coefficients
MLE	Maximum Likelihood Estimation
MLF	Multiple Label Files
MLLR	Maximum Likelihood Linear Regression
MMIE	Maximum Mutual Information Estimation
PLP	Perceptual Linear Prediction
SLF	Standard Lattice Format
STT	Speech-to-text
TTS	Text-to-speech
VTLN	Vocal Tract Length Normalization
WLP	Warped Linear Prediction
/a/	Phoneme “a”
[a]	Speech sound “a”
Mon	Monophone Model Set
PoA	Triphone models clustered according to the Place of Articulation
ToP	Triphone models clustered according to the Type of Articulation

TLo	Aggressively tree-clustered triphone models
THi	Moderately tree-clustered triphone models

Chapter 1

Introduction

The goal of Automatic Speech Recognition (ASR) system is to symbolise the human generated sounds i.e. to transcribe human speech into text, which can be further processed by machines or displayed for human's readable form in various applications. It can be simply a signal to symbol transformation. Despite the vast amount of efforts put in this research area of ASR, there are still numerous problems to solve.

Emerging applications of ASR system brings the technology tools to teach languages and their pronunciation without human tutors. This kind of work will helps to improve the literacy rate in multi lingual country like India. Few of the examples of this research work is an ongoing research institutions in worldwide, one such example is Project LISTEN (**L**iteracy **I**nnovation that **S**peech **T**echnology **EN**ables) is an inter-disciplinary research project at Carnegie Mellon University to develop a novel tool to improve literacy [7].

Even in India, a lot of research work is going in the area of Speech Technology usage in Language learning process. This technology teaches pronunciation using speech synthesis and learning pronunciation by Speech Recognition system. It helps illiterate people to communicate with computers and internet access by querying the requirements through speech in their mother tongue and any other natural languages. This idea opens another domain for Language Translation research, so that connecting the different language speaking people can have common discussion without knowing other language. Already online Multilingual dictionaries are available in text form. Accessing those information by simple speech query is another new research domain in Speech Technology.

India is a multilingual multi cultural country with more than 200 Languages. The 1961census reports 1652 mother tongues according to one estimate, there are 613 Tribal communities speaking 304 mother tongues which can be reduced to 101 distinct identifiable languages. The Government of India presently focusing in the mother tongue based multilingual education to improve the literacy rate in tribal

community. The purpose of a multilingual education (MLE) [23] program is to develop appropriate cognitive and reasoning skills through a program of structured language learning and cognitive development, enabling children to operate successfully in their native, state and national languages. MLE provides a strong foundation in the first language (mother tongue), adding second (e.g. national) and third languages (e.g. English) enabling the appropriate use of both/all languages for life-long learning (Malone 2005) [24].

Multilingual education is also multicultural, with learning process in the child's known environment and bridging to the wider world. The bridging process allows children to maintain local language and culture while providing state and/or national language acquisition and instruction. This process provides learners with the opportunity to contribute for national society without forcing them to sacrifice their linguistic and cultural heritage.

"UNESCO supports mother tongue instruction as a means of improving educational quality by building upon the knowledge and experience of the learners and teachers." "UNESCO supports bilingual and/or multilingual education at all levels of education as a means of promoting both social and gender equality and as a key element of linguistically diverse societies" [24].

In India, with the growing popularity of computer and increasing number of users in rural places, there is a lack of computer literates. The Speech Recognition technology will helps in this scenario by accessing computers operation. The common man will find it easier to operate the computer simply by using speech as the mode of operation in his/her own language.

1.1 Motivation

The motivation for this work came from my interest in Speech Recognition and love for my language Banjara. The constitution of India Article 350-A says that adequate facilities should be provided of teaching in mother tongue at primary level for linguistic minorities [23].

Article 29 envisages that for educational and cultural development of the linguistic minority's protection should be provided for special language groups and opportunities should be given for preserving their language, script and culture through use in education.

1.2 Problem definition

The main aim for this work is to build a Speech Recognition System for Banjara Language also known as Lambada in Andhra Pradesh. This work focuses on studying the banjara language and collecting data for the Speech Recognition. Sphinx-III has been used to build the Speech Recognition system for the language.

1.3 Overview of the work

The work focuses on building the ASR system for Banjara language. Speech samples are collected from different people for creating a speech database for the experiment.

The data are being collected from different environment viz. lab environment, room environment etc. The works analyses and compare the different result obtained in ASR experiment.

Speech sample of native speaker and non-native speaker is also being analyzed in this work.

1.4 Practical applications

Today the Speech Recognition technology is being used in many computer applications.

It's being used as biometric application to authenticate voice and enabled security system. Dictation is widely used in various fields. It is used for medical transcription, general word processing etc. also finds its application in language courses for improving user's pronunciation. And also come as a relief for medically unfit people in helping them to used computer via voice.

In this work, design an application for Banjara speaking people to learn other languages like Telugu, English, etc…and another task is to translate Banjara to Telugu language.

1.5 Objective &Goal

The main objective of the work is to develop a speaker Independent Automatic Speech Recognition (ASR) system for Banjara language and to build a simple computerized translation from Banjara to Telugu Language.

1.6 Organization of the Thesis

The Thesis is divided into eight chapters. The contents of the chapters are as follows.

Chapter2 describes about "the basics of speech, phonetics, speech production and speech perception".

Chapter3 describes "Introduction to the Banjara language, Language phonemes and phoneme statistics".

Chapter4 describes "the Historical review of the Automatic Speech Recognition, structure of the ASR system" …

Chapter5 describes the Brief idea of "an Hidden Markov model".

Chapter6 discusses an "Introduction to the Sphinx" and versions of sphinx and working procedure.

Chapter7 describes about the Database and "experimental procedure" and "the results and result analysis".

Chapter8 discusses about the conclusion and future work of the work.

Chapter 2

Speech, articulation and phonetics

Speech is the vocalized form of human communication. It is based on the syntactic combination of lexical and names that are drawn from very large vocabularies. Each spoken word is created out of the phonetic combination of a limited set of vowel and consonant speech sound units.

2.1 Speech events, phones and phonemes

A speech event is any kind of sound produced by the speech organs of a person. Two speech events are considered to be different if any difference can be found between them, for example, in the spectral structure or timing. Therefore, any two speech events are most likely different even if produced by the same person. [10,11]

Speech events that are phonetically equivalent between speakers can be referred to as phones. Phonetic equivalence is a complex concept that relies on the idealizing assumption that the differences between a set of speakers' phone organs can be ignored when evaluating the speech event's phonetic quality. [25]

Phones are regarded as realizations of the same abstract phoneme they are its allophones or variants [25]. It should be emphasized that a phoneme is indeed an abstract concept. It is not possible to divide a word into phones with clear, unambiguous borders the speech signal is changing continuously. The phoneme model is simply an idealizing abstraction of the complex phenomenon which is known as speech.

Quite often the word "phone" is used when one actually means a phoneme. For example, one could speak about phone recognition, when the ASR system in question is actually based on phonemes. Even more often, there is confusion between the corresponding adjectives phonetic and phonemic.

2.1.1 Notation

It is customary that speech sounds are marked in brackets (e.g. [a]) and phonemes between slashes (/a/). In both cases quantity, i.e., whether a phoneme or a phone is short or long, is marked with a colon ([X] or /X/).

There exists an alphabet that includes symbols for phonemes in different languages, and is called the International Phonetic Alphabet (IPA). Instead of IPA symbols, this thesis uses Telugu graphemes for describing the corresponding Banjara language phonemes. An exception is the /x/ phoneme that has no corresponding grapheme. In this case, the IPA symbol is used.

For most of the Banjara Language phonemes the Telugu character is identical to the grapheme. Table 2.1 presents the exceptions.

Graphemes Notation	IPA Notation
/a/	/ɑ/
/ä/	/æ/
/ö/	/ø/

Table 2.1: Exceptions in IPA notations of phonemes

2.2 Speech production

According to the source-filter model of phonation there are three main aspects needed for speech production: a source of energy, a source of sound, and a filter. None of these can be directly mapped to specific organs for example, the location of the sound source may vary for different speech sounds [6].

The main part of the energy source is provided by the respiratory muscles. When the volume of the lungs is decreased by relaxing the diaphragm and intercostals muscles and thus increasing the pressure inside the lungs, the pressure difference between the lungs and the parts of the vocal tract above the glottis (supraglottal vocal tract) begins to equalize, and air starts flowing through the glottis and the vocal tract

unless there is something blocking the tract. This kind of air flow from the lungs to the environment called regressive air flow is the most important mechanism for human speech production. In some rare cases, ingressive air flow to the lungs is utilized as well.

As such, air flowing through the vocal tract does not produce much sound. The glottis, a gap between the vocal chords in the larynx, works as a valve between the lungs and supraglottal vocal tract. For in and exhaling, the valve should be open, allowing air flowing to and from the lungs as freely as possible.

In phonation of voiced speech sounds the vocal folds are moved closer to each other and the glottal orifice is narrowed. The air flow forced through the glottal slit makes the vocal folds vibrate. The glottis opens and closes quickly at a frequency around 100 Hz for males, 200 Hz for females and 300 Hz for children, causing constant changes in the air pressure above the glottis. This opening and closing acts as the source of sound. This frequency can be heard in voiced sounds and is referred to as pitch. An example of the speech sound [a] is presented in Figure 2.1. The speaker can, to some extent, regulate the frequency of these glottal pulses. The pulses generated by the glottis could be heard as such but in reality the sound wave passes through the vocal tract filter, which causes significant changes to the produced sound. For unvoiced speech sounds the source of sound is a constriction somewhere along the vocal tract that is so narrow that air flow through it produces turbulence and Noise like sound.

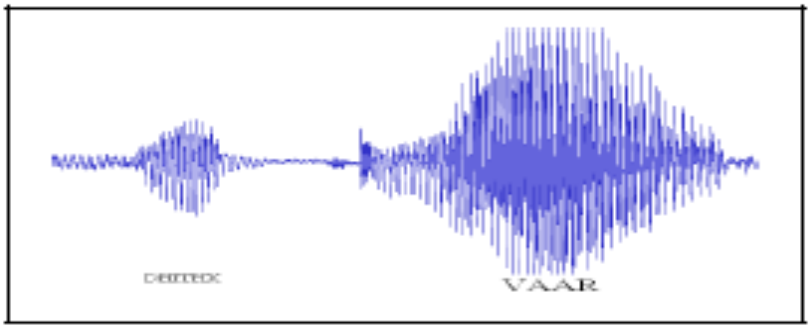


Figure 2.1: Shows the waveform of the word [DHITHXVAAR]

For the speech sound [s], for example, the constriction is between the tip of the tongue and the alveolar ridge.

The part of the vocal tract that is located in front of the sound source acts as the main acoustic filter for voiced sounds it consists of the part from the glottis all the way up to the lips and for labial sound sources, e.g., [f], only of the small gap between the bottom lip and the upper teeth. As the word filter suggests, this part of the speech system amplifies some and attenuates other frequencies of the signal produced by the sound source. [12]

2.3 Production of the different speech sounds

The classification of phonemes to vowels and consonants is a universal feature of all languages. For the Banjara language, vowels are sometimes defined as the set of phonemes that can alone form a syllable. Unfortunately, this definition does not hold for all other languages. Therefore, a more general definition is required.

Vowels are voiced sounds during which the air flow through the vocal tract is relatively free. The state of the vocal tract organs remains relatively stable during their phonation. Additionally, in order to exclude nasals ([m, n, x]) and laterals ([l]) from the vowel class we require that air has to flow freely out of the middle of the mouth.

All the speech sounds that do not match the definition of a vowel, are called consonants. The different types of consonants in the Banjara language are stops, fricatives, nasals, tremulants, laterals, and semi-vowels. [20,21]

2.3.1 Vowels

Vowels can be classified by the positions of the articulators (jaw, tongue, lips). The position of the tongue during the phonation of six Banjara language vowels, [i, e, a, u, o], and

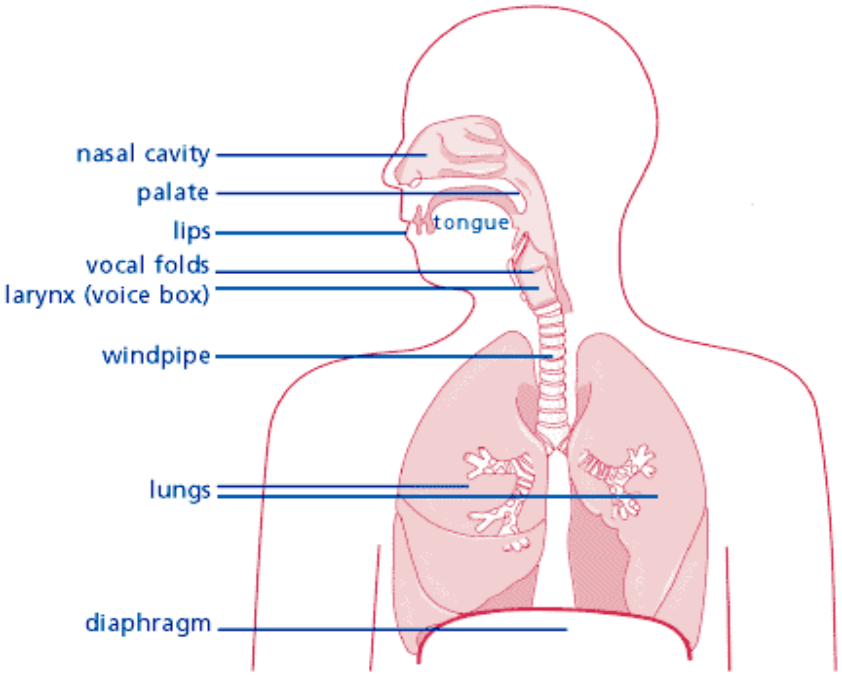


Figure 2.2 :Shows the different sounds production in human body

[a] is shown in figure 2.2. [y] and [ö] are not shown, since the tongue is approximately in the same position as for the vowels [i] and [e], respectively. [20]

The difference in the constrictions of the vocal tract is to a good part defined by the location of the highest point of the tongue.

As depicted in the figure, vowels can be classified in different ways. Whether they are front, central, or back vowels is decided by the location of the constriction caused by the tongue. Two other ways to classify them is by the narrowness of the strait and the roundness of the vowel. For example, [i] and [y] are both closed front vowels, but [i] is broad while [y] is round.

2.3.2 Consonants

The production mechanism of different vowels is quite similar, but the differences between the consonant classes are larger. While producing [l] or [j], for example, the air flow in a fashion that resembles greatly the air flow while producing vowels. On the other hand, for the stops [k, p, t], the air flow stops completely for a moment and the actual sound of these phonemes is produced by the burst like

explosion noise when the vocal tract finally opens. In this subsection, the production of the native Banjara language consonants [d, h, j, k, l, m, n, x, p, s, t, v] as well as [b, f, g] included in Telugu language due to other languages are considered.

Stop consonants [k, p, t, g, b, d]

When producing stops the vocal tract is at first completely, or almost completely, closed at some point causing a high difference in pressure between the parts of the vocal tract above and below the closure. At this point, there is little or no sound present. The actual stop consonant is produced when the vocal tract suddenly opens and a pulse-like sound is released. [k, p] and [t] are voiceless stops while [g, b], and [d] are voiced. For [k] and [g] the closure during the closed phase of the stops is located in the velum, [t] and [d] in the dental area, and [p] and [b] in the labial part of the vocal tract.

Fricatives [f, h, s]

In the production of fricatives there is a constriction somewhere along the vocal tract that is narrow enough to produce noisy turbulent air flow. This noisy sound is then altered by the vocal filter. Banjara language fricatives are always classified as voiceless. However, for example, the [h]-sound between two vowels may be voiced.

Nasals [n, m, x]

Nasals are voiced consonants in which, air flows through the nasal cavity while the primary vocal tract is closed. The location of the closure creates the main distinction between the three nasals. The sound is acoustically filtered by both the part behind the closure of the vocal tract and the nasal tract.

Tremulants [r]

The Banjara language [r] is produced by letting the tip of the tongue vibrate against the alveolar ridge. The frequency of this vibration is typically 20 – 25 Hz. In Indian languages such as Telugu or Hindi, the place of the articulation is farther back in the vocal tract and no strong tip vibration is produced.

Laterals [l]

The tip of the tongue blocks free air flow by pressing against the alveolar ridge. However, on both sides of the tongue tip there is a passage for sound waves and air flow.

Semi-vowels [j, v]

The consonants [j] and [v] resemble vowels in general, but the constriction in the vocal tract is more powerful and the state of the tract is more context-dependent.

2.4 Speech perception

2.4.1 Anatomy of the ear

The ear is composed of three sections: the outer, middle, and inner ear, as depicted in figure 2.4. The outer ear is responsible for directing the sound waves to the eardrum, where the middle ear transforms air pressure variations into mechanical motion. In turn, the inner ear transforms the movements into electrical signals in the auditory nerve. [9]

The outer ear consists of the pinna and the ear canal. The pinna is essential for sound localization as it attenuates sound coming from behind the head. The ear canal, or the meatus, can be regarded as a hard walled acoustical tube which boosts frequencies between 2 and 8 kHz. The first and second formants of speech are located within this frequency range.

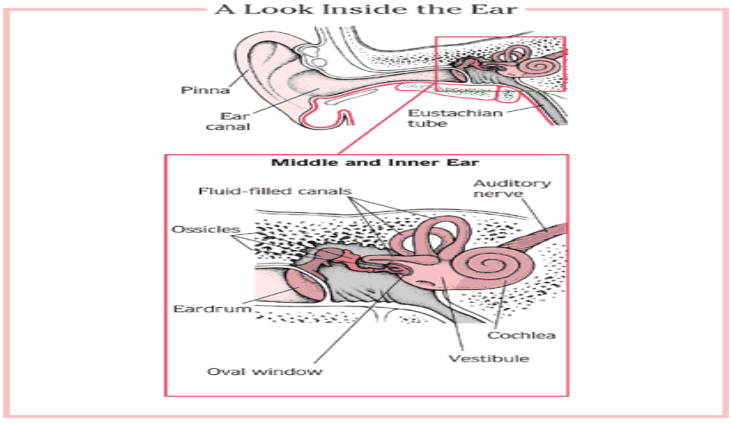


Figure: 2.3 Different parts of the Human ear

The middle ear begins at the eardrum. Together with the three ossicular bones (hammer or malleus, anvil or incus, and stirrup or stapes) it linearly converts air pressure variations in the ear canal into the oval window membrane at the start of the inner ear. The acoustic impedance of the inner ear fluid is about 4000 times that of air. Therefore, most of the pressure waves hitting the inner ear would be reflected back if there was no impedance transformation mechanism. The middle ear also includes a mechanism for protecting the delicate inner ear against loud and low frequency sounds.

The part of the inner ear which performs in hearing is the cochlea, a snail-shaped fluid-filled tube. It is connected to the middle ear through the oval and round windows.

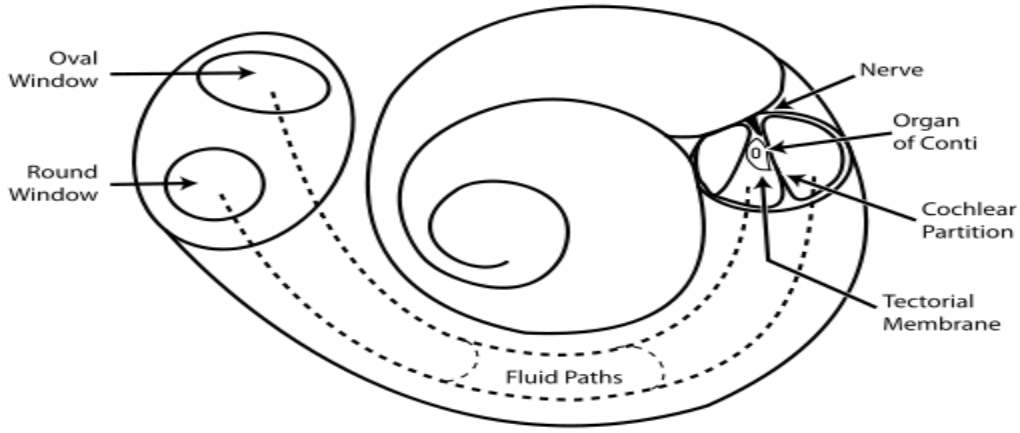


Figure: 2.4 Cross section of the Cochea

The cochlea is a sensitive and complex organ. It is divided into several compartments by membranes (see cross-section in figure 2.4), and each of these compartments contains fluid with different chemical makeup, causing potential differences between the compartments.

On the basilar membrane (BM) lies the organ of Corti, which contains hair cells organized in several rows along the length of the cochlea. The hair cells' hairs bend when the basilar and tectorial membranes vibrate, causing neural firings in the auditory nerve.

The BM varies gradually in tautness and shape. It is stiff and thin in the beginning (close to the round and oval windows), but flexible and thick at its end. Thus, its frequency response also varies: each location on the cochlea has a characteristic frequency at which it vibrates maximally. The characteristic frequency is high in the beginning of the BM and low at the end.

Sound entering the cochlea causes vibrations in the fluid, which creates traveling wave on the BM, which progresses from the beginning of the cochlea towards the end. This wave reaches maximum amplitude at the point on the BM whose characteristic frequency matches the frequency of the input sound.

2.4.2 Psychoacoustics and the ear

There are two main ways to study the properties of the ear: 1) direct physiological measurements and experiments, and 2) psychophysical experiments. The former is very difficult to perform due to the delicate and complex structure of the ear. In the latter, the response to sounds is studied indirectly, and relies on psychic response. This approach is called psychoacoustics and studies sensations, subjective responses to different stimulus.

There are several results of psychoacoustical research that are essential in speech processing. The most important aspects are the different psychoacoustical pitch scales which are used to imitate the human hearing system. Before describing the different scales, a few psychoacoustic concepts need to be introduced.

Critical band

Critical band is an essential concept when studying the frequency resolution of hearing. A critical band defines a frequency range in psychoacoustic experiments for which perception abruptly changes as a narrowband sound stimulus is modified to have frequency components beyond the band. When two competing sound signals pass energy through such a critical-band filter, the sound with the higher energy within the critical band dominates the perception and masks the other sound [9]. A critical band is approximately 100 Hz wide for center frequencies below 500 Hz, and 20 % of the center frequency for higher frequencies.

ERB band

Equivalent Rectangular Bandwidth (ERB) bands are another way to measure the bandwidth of analysis of the hearing system. The ERB of the auditory filter is assumed to be closely related to the critical bandwidth, but it is measured using a more sophisticated method rather than on classical masking experiments involving a narrowband masker and probe tone. As a result, the ERB is thought to be unaffected by activity in frequencies outside the studied band. [10]

Pitch

The pitch describes the subjective sensation of sound on the frequency scale ranging from “low” to “high”. The closest match to pitch in physically measurable quantities is frequency, but pitch is also dependent on other quantities. There are two concurrent theories that try to explain the sensation of pitch: the place theory and the timing theory. The former is based on the fact that that the functioning of the basilar membrane (see section 2.4.1) is responsible for the sensations, while the latter is based on observations of the ear performing some sort of time domain periodicity analysis. Currently it is known that neither of the theories alone explains pitch sensations and that the hearing system includes more than one method. Several pitch scales have been developed based on slightly different aspects of psychoacoustics. The mel, Bark, and ERB scales are presented in the following subsections. [11,12]

Mel scale Filter bank

The Mel scale is formed as follows: a starting frequency is selected and played to a subject. Then, a sound is searched for that in the subject’s opinion is half (or double) in pitch. This procedure is repeated as necessary with the frequencies of new sounds as starting frequencies, and the results are averaged over many subjects. Finally, an anchor point is selected where the frequency and pitch scales are set to be equal. The dependency between frequency and pitch is approximately linear up to 500–1000 Hz but above that it is close to logarithmic. This kind of scale is called the Mel scale1, and the unit of the scale is Mel. An approximation usable on low frequencies is given by the equation:

$$m = 2595\, log10\, (1 + f/700)$$

Where m denotes the number of mels and f is the frequency in hertz. The anchor point for this scale is 1000 Hz which corresponds to 1000 Mels. Other approximations exist, as well, and they are usable on different frequency bands. [9]

Bark scale

The Bark scale (named after the German acoustician Barkhausen) is based on critical bands. On this scale one critical band corresponds to one Bark. The Bark and Mel scales are closely related and 1 Bark approximately equals 100 Mels. [11]

ERB-rate scale

At different times the different pitch scales have been thought of mapping sounds linearly to the basilar membrane so that a constant difference on the pitch scale corresponds to a constant shift in the resonance point on the BM. According to latest research, the ERB-rate scale is the best match for this relation. As the name suggests, the ERB-rate scale is based on the ERB bands in a similar way that the Bark scale is based on critical bands. [12]

Stevens' power law

According to the Power law by Stevens [11], introduced in 1957, the relationship between stimulus intensity and sensation magnitude is described by the equation:

$$Y = kIn$$

Where Y is the sensation magnitude, k is a constant factor, I is the magnitude of the stimulus, and n is the exponent that has a different value for different sensations. If the exponent is less than one, the function is compressive, if n is equal to one, the function is linear, and otherwise it is expansive. Stevens measured values for n for different sensations, for example for brightness ($n \approx 0.33$) and for electrical Shocks to teeth ($n \approx 7.00$). For the loudness of a 3000 Hz tone a value of $n \approx 0.67$ was obtained. However, on moderate sound levels and lower frequencies the value is significantly lower [10, 11].

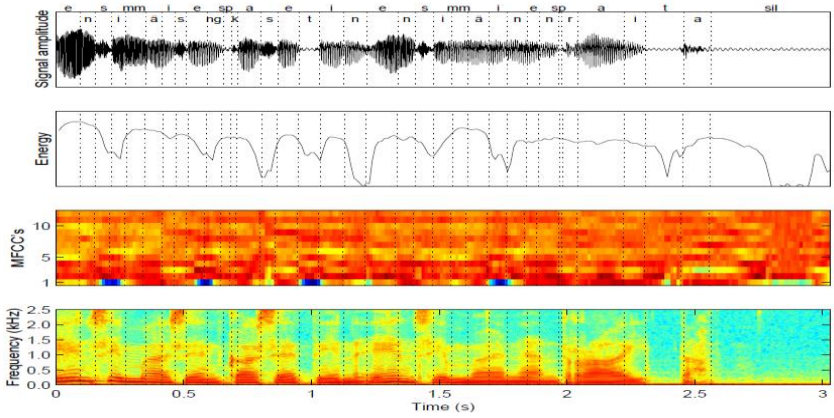


Figure: 2.5 The waveform, spectrogram, energy, Mel cepstral representation of same speech signal along with phoneme annotations.

The power law is in contradiction with previous results by Fechner who argued that the relation between stimulus and sensation intensities would be logarithmic (Y = k log I). Even today, many algorithms use the logarithmic function when approximating human sensation intensities.

2.5 Speech representation in computer

Humans use both audible and visual cues when recognizing speech with the audible information being more important. In the early days of automatic speech recognition only audible information was utilized since computers were not powerful enough for image processing. Currently, there are projects aiming at combining audible and visual information in ASR [1], but for many common applications the acoustical Stimulus is the only one that can be used. An example of this is telephone application.

The analog speech signal is converted into an analog electrical signal within a microphone and into discrete, digital samples by an analog-to-digital converter (ADC). The amount of digital data depends on the sampling frequency as well as the amplitude resolution of the samples. For telephone speech, the frequency is usually 8 kHz and for CD-quality 44 kHz. A reasonable compromise between sound quality and

storage space required dictates a 16 – 22 kHz sampling frequency. In the experiments of this thesis a frequency of 16 kHz was used.

In Figure 2.5 four different views of the same speech signal are shown. The topmost is the speech waveform, the second is the overall energy of the signal, the third the standard 12th order mel-cepstral representation, and finally the short-time FFT representation, shown as a spectrogram. The utterance presented is part of the training data used for the experiments of this thesis and it is uttered by a male speaker.

The individual sounds are visible in the spectrogram as well as the different formants which are the resonances produced by the vocal tract. Different phonemes produce different patterns of formants, and a human being can, for example, identify clearly pronounced vowels from each other quite simply by looking at their spectrograms. For all the /s/'s, there is clearly more energy in the higher frequencies than for other phones. This shows the difference between noise-like fricatives and voiced sounds.

This chapter concludes describing speech basics related to human and machine understanding concepts which are necessary for building the ASR system.

Chapter 3

Introduction to Banjara Language

Author Samuel Johnson put as “Language is the dress of thought”. Language should be developed to preserve and propagate culture and more importantly virtues among a given group of people. Bahadur Singh Rathod, an 8th class pass police constable in Adilabad has designed a script for his mother tongue, the Banjara dialect, who says that “I wanted my mother tongue to be more useful to my tribesmen which are known as Banjaras or Lambada’s as they are known in Andhra Pradesh”[8]. This shows the tribe people love towards their mother tongue.

3.1 Special features of the Banjara language

The Banjara language, along with other languages such as Hindi and Telugu, is fundamentally different from all other major Indian languages. In this section, the most important features of the Banjara languages from the speech recognition point of view are discussed.

3.1.1 Banjara is close to a “phonetic” language

As a rule of thumb, the Banjara language writing system follows the phonemic principle. There exists a different grapheme corresponding to each phoneme that is not used for any other purpose. This makes many aspects easier in speech and language research. For example, the conversion of written text to phoneme similar to the Telugu language.

3.1.2 Strings in speech synthesis is straightforward

Still, there are exceptions, with the most obvious being the phonemes GxG and GxXG that are transcribed as graphemes “n” (usually, in front of a “k”) and “ng”. Additionally, the difference between spoken and written language causes irregularities between the graphemes and phonemes, phonemes are often dropped and added at certain locations of a word. In loanwords, on the other hand, a less educated speaker might replace the foreign consonants [b, f] and [g] with the unvoiced Banjara Language counterparts [p, v] and [k], respectively. Coarticulatory facts do cause phonemes to appear as different allophones in different contexts, but the number of

different phonemes perceived by a native Banjara language speaker is close to the number of different graphemes in the written language. As a result of Banjara language being a phonetic language, native Banjara speakers can easily interpret a string of phonemes produced by a phoneme recognizer.

3.2 Number of words

The morphology of the Banjara language is complex, which makes the number of different words immense. This is due to several factors [13] explains as follows:

3.2.1 (A) Banjara **is a synthetic/agglutinative language**

As opposed to analytic languages, the meaning of a word is typically changed by adding a suffix to the word and not by using additional words. There are sixteen different cases and additionally several other grammatical meanings which are expressed through suffixes, e.g., possessive relation, questioning and hesitation. These kind of suffixes are used with nouns, pronouns, adjectives, numerals, and even verbs. Suffixes of ten replace prepositions used in other languages.

As an example, the word “AANGAPACHADIAK”, “indeed in his/her/their trees” can be split into morphemes (/AA/ /NG/ /P/ /AA/ /D/ /IA/ /K/) in the following way:

- AA is the root of the word. The basic form of the word tree is A, here the second /a/ has disappeared. This shows that Banjara Language is not cleanly agglutinative.
- AANGA denotes a plural form.
- [GA] is the suffix for the inessive case, which is roughly equivalent to the Telugu preposition in.
- [AA] represents the possessive form of the third person, singular or plural.
- [K] makes the word have an insisting tone.
- Verbs are conjugated in person, tense, and mood.
- The word ’AANGAPAACHADHIAKANCHAAL’ is as such a proper sentence in Banjara Language. When translated accurately into Telugu, one needs seven words: Would I really run a little around. Again, this word can be split into different morphemes in the following way:

- AANG is the root of the word, derived from the verb to Front
- PAACHA means that the back is done in an aimless way
- DIAK indicates the conditional mood.
- N indicates the first person, singular
- LAA makes a question of the word
- CHAAL the same suffix as in the previous example, but, in this case, it makes the tone hesitating, rather than insisting.

3.2.1 (B) New words are formed by composing other words

This is true for some Tribal languages, e.g. Banjara and Koya, as well. To some degree, it also applies to Telugu.

The great number of different words makes it inefficient to make a simple list of Banjara language words that would be usable for a large vocabulary speech recognize. One would need to add knowledge about morphology to the language model. This is totally different for Indo-European languages and the large vocabulary recognizers for those languages usually rely highly on word lists, i.e., dictionaries. For Banjara language some other approach is required. In this work, as well as previous works, phoneme recognition has been chosen as the solution. [14, 15]

3.3 Quantity of phonemes

Phoneme duration, or quantity, is a distinctive feature in Banjara language the meaning of a word can be changed by varying the temporal durations of the phonemes. For example, the words KA, KAD, KAAD, KHAAD, and KHAD all have different meanings, and even the non-existent words KAA and KKAD sound like proper Banjara language words, but do not happen to have a meaning.

3.4 Vowel harmony

In a Banjara language non-compound and non-loan word, front vowels [y, a], and [ö] and vowels [a, o], and [u] are never present at the same time. The front vowels [i] and [e] are neutral in this respect. They can be combined with both front and back vowels. This fact can be utilized in speech recognition applications especially at the language model level.

3.5. Few consonant sequences

Consonant sequences are rare and never appear in the beginning or end of a native Banjara language word. In contrast, it is possible that a word consists solely of vowels (e.g., aie).

3.6 Phonemes present in the Banjara language

The Banjara language phoneme set is smaller than the Telugu one. In native words, there are six vowels and thirteen consonants, and three additional consonants appear in loanwords. Both long and short versions exist for most phonemes.

Vowels

The Banjara language set of vowels consists of four front vowels and two back vowels. The front vowels are /e/, /i/, /y/, /ä/, and /ö/ corresponding to the graphemes /e/, /i/, /y/, /ä/, and /ö/, respectively. The back vowels are /a/, /o/, and /u/ with respective graphemes /a/, /o/, and /u/.

Consonants

There are thirteen different consonants in native Banjara Language words: /d/, /h/, /j/, /k/, /l/, /m/, /n/, /p/, /r/, /s/,/ t/, /v/, and /x/. Not all of them appear at all word positions for example, the consonants /d/, /h/, /j/, /k/, /m/, /p/, /v/ and /x/ do not appear at the end of a word. Additionally, /d/ and /x/, which are usually present only due to word infection, have more restrictive appearance constraints. The appearance of the phoneme /h/ is fundamentally different, depending on its location in the syllable. In the beginning of a syllable it is unvoiced and could be classified as a semi-vowel. In the end of a syllable it is often voiced and clearly a fricative. Both of these variants undergo a heavy allophonic variation. [16]

3.7 Phoneme statistics

Table 3.1 and 3.2 shows the statistics of Banjara language phonemes from four different studies. Also, the statistics of the book used for the experiments of this thesis are shown (1996). The statistics made by the linguists as well as the Dr GonaNaik a person who is belong to the Banjara community and Aacharya H.S. Brahmananda are

provided letter statistics than phoneme statistics, and the phoneme x is not taken into account. Phoneme based his study on the Banjara Language translation of the New Testament from Banjara to Telugu Language[ ]. The table is ordered according to the statistics gained from Bharata Desham lo Banjaraalu book.

వర్ణాలు : లంబాడీ భాషలో వర్ణాలు			
అజ్వర్ణాలు:-			
	అగ్ర	కేంద్ర	పశ్చాత్
ఊర్ధ్వ:-	ఇ ఈ		ఉ ఊ
మధ్య:-	ఎ ఏ		ఒ ఓ
నిమ్న:-		అ ఆ	

Table: 3.1 Shows the Vowels

హల్వర్ణాలు:- →					→		→	
స్థానములు	ప్రయత్నాలు							
ప్రయత్నములు	ఓష్ఠ్యములు		దంత్యములు	దంత్యమూలీయం	మూర్ధన్యములు	తాలవ్యములు		కంఠ్యములు
స్పర్శములు	శ్వాస	ప	త		ట	చ		క
	నాద	బ	ద		డ	జ		గ
ఊష్మములు				స	ష			
అనునాసికం		మ		న	ణ			
పార్శ్వికము				ల	ళ			
కంపితం				ర				
అంతస్థం		వ				య		
స్పష్టోష్మం	శ్వాస							
	నాద							
తాడితం				ఢ				

Table: 3.2 Shows the Consonants of Banjara Language

The table No. 3.1 shows how well important vowels are in Banjara Language almost fifty percent of the phonemes are vowels. It also clearly shows that phonemes /a/, /i/, /e/, and /n/ are most common while /ö/, /x/,/ g/, /b/, and /f/ the most infrequent.

This chapter concludes the basic language structures of Banjara languages and their phoneme information comparing with the Telugu language.

Chapter 4

Automatic speech recognition systems

4.1 Historical review

This section is based on [1], unless otherwise stated.

The First pioneer in speech technology design was Wolfgang von Kempelen. He designed a mechanical device mimicking the human vocal tract in 1791 [5]. Forty years later, based on the design by Kempelen, Charles Wheatstone built a machine capable of producing a good selection of both voiced and unvoiced sounds. This machine had a leather tube that replicated the vocal tract and a vibrating reed as the vocal cords, and bellows representing the lung.

The trend of imitating the human vocal tract using mechanical means continued into the 20th century. Gradually, when new algorithms, electronics, and finally computers with increasing computational power were developed, the scope of speech research broadened to speech coding and recognition applications. In 1928, Homer Dudley invented the vocoder that made it possible to transfer speech on the phone line by using only a fraction of the bandwidth. [1]

The first application of automatic speech recognition was a toy dog called Radio Rex produced in the 1920's. An electromagnet kept the dog in its house until its name was called. Then the simple electrical recognizer would react to the acoustical energy contained in the /e/, cut the current from the electromagnet, and let the dog jump out of its house. Of course, the recognition was not accurate, the dog would react to a host of other words as well. [11]

An early but more serious application of speech recognition was the single isolated speaker digit recognizer designed by Davis et al. at Bell Laboratories in 1952 [13]. Olson and Belar tried to build a recognizer for ten monosyllable words at RCA Laboratories in 1956. These two recognizers relied heavily on spectral measurements and direct pattern matching [9].

In 1959, Fry and Denes tried to build a phoneme recognizer to recognize four vowels and nine consonants using a spectrum analyzer and a pattern matcher to make the recognition decision. This recognizer contained simple language model that relied in statistical information about allowable phoneme sequences.

During the 1960's and 1970's new methods for speech recognition were introduced silence detection methods, time warping, and dynamic programming. In the 1970's the first large vocabulary and speaker-independent systems were developed at IBM and AT&T. These recognizers mostly concentrated on Isolated Word Recognition.

The most fundamental change in the 1980's in the field of speech recognition was the shift from the spectral pattern matching techniques to statistical modeling methods, for example neural networks and more importantly hidden Markov models or HMM's. HMM's were first introduced in several papers by Baum et al. in the late 1960's and early 1970's [1,15] and shortly after independently extended to automatic speech recognition by Baker and Jelinek [16]. However, they did not become popular in speech recognition until the 1980's.

In 1975, Jelinek et al. presented the idea of language modeling in the form it is used today [26]. The goal of language modeling is to find and represents the relationship among words in sentences, just as the goal of acoustic modeling is to find regularities inside models. The so-called N-gram model is a set of conditional probabilities of vocabulary words followed by other words, estimated from large text material. A well trained model gives lower probabilities to ungrammatical than grammatical sentences.

Few fundamental new innovations have been discovered in speech recognition since the discovery of HMM's. Fine details have been tuned and growing computational capacity has allowed the use of more complex models giving some performance boost. However, at the same time the need for training data has also grown.

4.2 Structure of an ASR system

This section explains in short the basic structure of a current state of the art standard speech recognition system. In the following subsections the different parts of the system are discussed in more detail.

A block diagram of a speech recognition system is shown in figure 4.1. The basic structure of all modern recognizers, independent of recognizer type, is presented in this figure. First discrete, equally spaced feature vectors are formed from a continuous speech signal. These vectors are assumed to carry a compact presentation suitable for classification of parts of speech. The actual recognizer makes use of the acoustic models, the lexicon, and the language model. Using these information sources the most likely transcription of the sentence is produced.

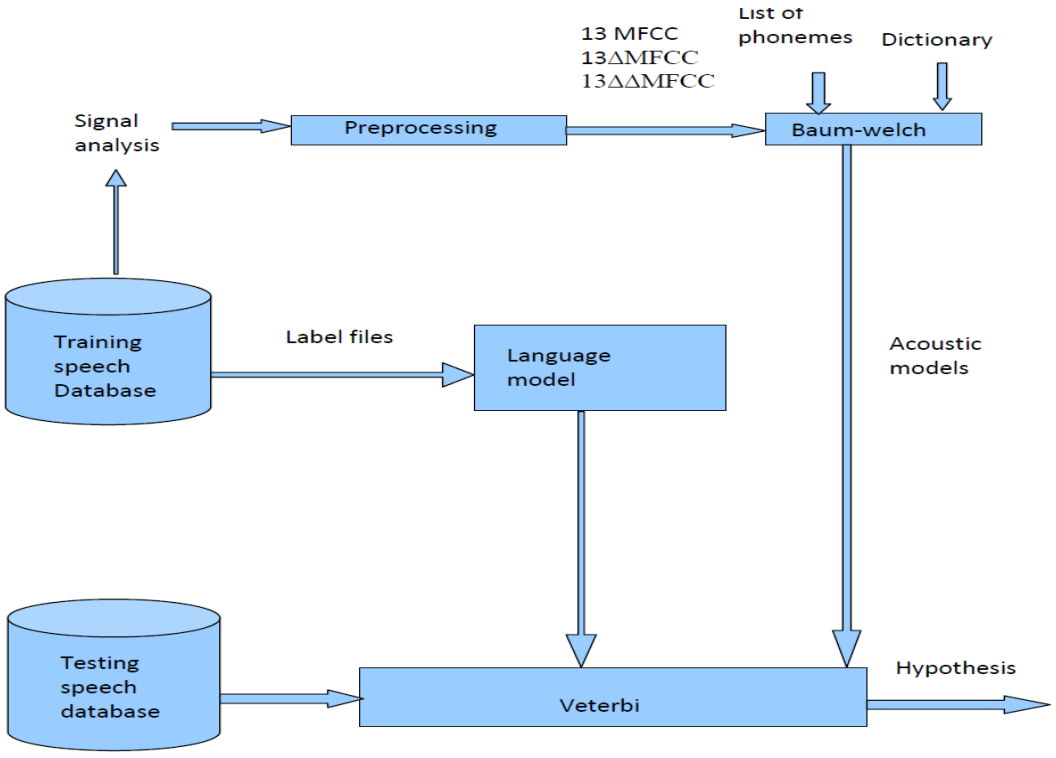


Figure: 4.1 The Automatic Speech Recognition System Architecture.

First of all, the speech signal has to be digitized so that it can be processed by the computer. The signal is then fed to the preprocessor, which is the feature extraction unit. The signal is split into frames consisting of about 200–800 samples,

and the frames are fed to the feature extractor itself which reduces the amount of data by calculating a vector of about 10–60 features, depending on the front end.

The feature vectors of the training data are used for parameter estimation for the acoustic models which characterize the acoustic properties of the basic recognition unit, e.g. a phoneme or a word. The lexicon is used to map the acoustic models to vocabulary words. The language model restricts the number of acceptable word combinations based on statistical information gathered from large text materials.

4.3 Acoustical front ends

Acoustical front ends are used for feature extraction purposes to transform the raw speech test files into more usable information. Features are numerical vectors that are designed to contain as much information relevant to recognizing phonemes as possible. On the other hand, they are a compressed representation of the speech data: much irrelevant information is discarded by the front ends.

Naturally, features should be discriminative they should differ sufficiently for the different phonemes so that building and training models is possible. Additionally, they should be robust features computed from separate utterances under different conditions should be similar. Another factor is computational efficiency that how much processor time is needed to extract the features from a speech utterance.

4.3.1 Common front end operations

Sampling: The acoustic speech signal has to be converted into a form that a computer can handle, a sequence of numerical values. This is done by taking discrete samples, or discrete values of the signal evenly spaced in time. A reasonable sampling rate for speech processing is 16000–22000 times a second, corresponding to sampling rates of 16 kHz–22 kHz

Pre-emphasis: Since there is more energy in lower than in higher frequencies in speech, speech data is high-pass filtered according to equation 4.1. The value of the constant a is typically around 0.97.

$$x'(n) = x(n) - ax(n-1) \qquad (4.1)$$

Windowing: Single sample values of raw digitized speech are hardly of any use for feature extraction. On the other hand, the whole utterance is usually too long and contains many different phonemes so it cannot be used for feature extraction, either. Instead, the utterance is divided into equally sized pieces, called frames.

Some further operations that are done for the framed data consider the signal as a periodic signal that is formed by concatenating the contents of the frame several times after each other, with the Discrete Time Fourier Transform (DTFT) being the most important of these. If the frames are formed simply by cutting pieces of the long utterance discontinuities arise wherever the beginning and the end of the original frame meet, which introduces anomalies to the spectrum produced by the DTFT. To prevent this from happening, a windowing function is used, with the most common being the Hamming window.

Additionally, in order to prevent problems occurring due to the selection of the frame locations, the signal is windowed in such a way that the frames overlap.

Frequency warping: It seems naturally desirable to have the front end simulate the function of the human ear, which is more sensitive at low than high frequencies. The frequency scale of the front end is altered similarly which can be achieved through filter banks or Warped Linear Prediction, for example.

Δ- ***and*** ΔΔ-***coefficients***: The performance of a speech recognition system can be enhanced by adding derivate coefficients to the static feature coefficients. While the static features describe the static state of the speech signal, the first and second order derivative coefficients, called Δ- and ΔΔ-coefficients, describe its dynamic properties, which are essential for modeling the transitions from one phoneme to another. The Δ-coefficients are calculated according to the following equation 4.2:

$$d_t = \frac{\sum_{\theta=1}^{\Theta} \theta(c_{t+\theta} - c_{t-\theta})}{\sum_{\theta=1}^{\Theta} \theta^2} \quad \ldots\ldots\ldots\ldots\ldots$$

(4.2)

where d_t is the Δ-coefficient, Θ is the width of the window used for calculating the Δ-coefficients, normally equal to two. c_t Is the value of the static coefficient at time t.

ΔΔ-coefficients are calculated similarly, except that instead of c_t -coefficients, the d_t -formulas are used in the equation above.

Cepstral liftering: The high order cepstral coefficients are often numerically quite small, and thus the variances of the different order coefficients would be from a wide range. This is not a recognition issue, but when displaying purposes it is convenient to rescale the cepstral coefficients to have similar magnitudes. This is done by cepstral liftering according to the equation 4.3, where c_n and c'_n is the original and the liftered coefficient, 'n' is the cepstral order, and L is a liftering constant.

$$c_n^{'} = \left(1 + \frac{L}{2} sin \frac{\pi n}{L}\right) c_n \qquad (4.3)$$

4.3.2 Mel Frequency Cepstral Coefficients

Mel Frequency Cepstral Coefficients (MFCC's) is the standard front end used in speech recognition. It provides good performance under clean conditions but is not very robust to noise. When calculating MFCC's, the spectrum of the speech signal is divided into channels by applying a set of band-pass filters to the signal. A computationally efficient way to do this is to operate directly on the DTFT of the signal with a triangular filter bank directly in the frequency domain (see figure 4.2). The triangles of the filter bank overlap partially, and they are broader at high than at low frequencies. This provides better resolution at low frequencies where the most important first and

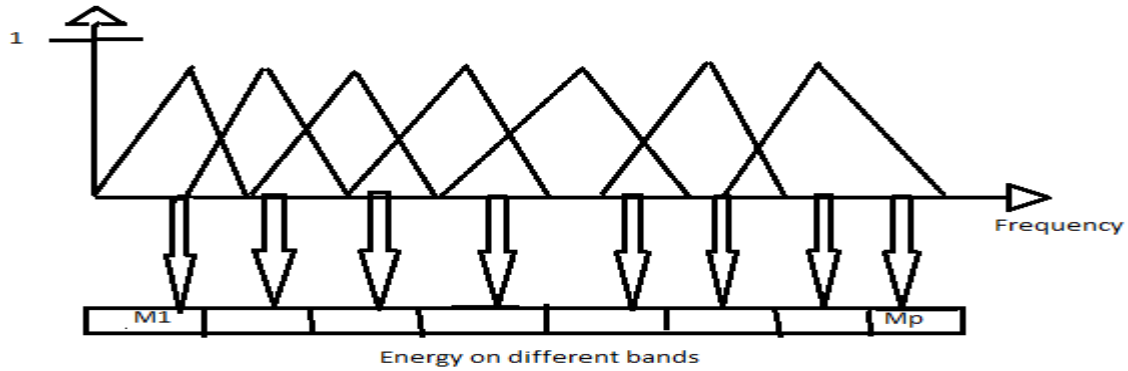


Figure: 4.2 The Structure of the Mel filter bank. The separate m-terms are the components of mel spectrum.

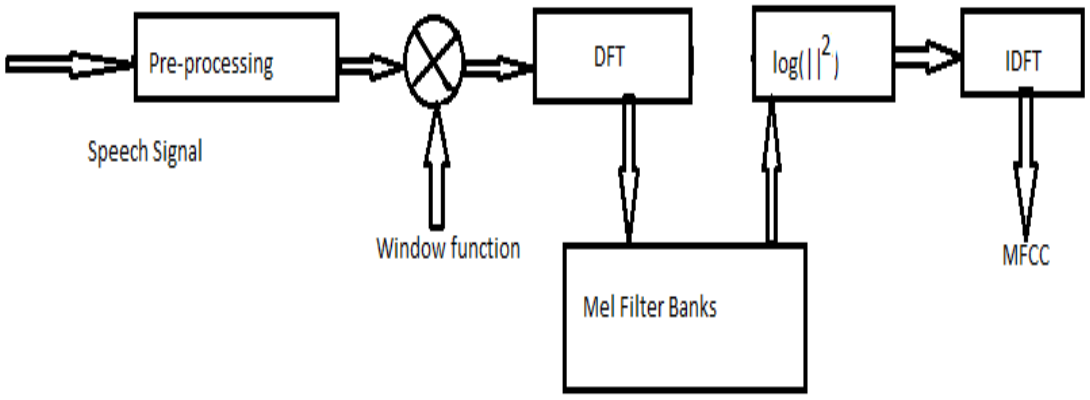


Figure: 4.3 The Computation of the Mel cepstral coefficients.

second formants are located. The triangles need to be tuned carefully to perform mel-warping. If they are located differently, the front end might still produce reasonably good results, even though the resulting scale would be different from the mel scale.

The outputs of the filter bank form the mel-spectrum. The final cepstrum coefficients are obtained by applying the inverse Fourier transform on the logarithm of the absolute values of the mel-spectrum squared. In this case, the inverse Fourier transform is equivalent to the discrete cosine transform (DCT), which is a computationally lightweight operation. Additionally, the DCT produces values that are highly uncorrelated, making the production of speech models easy. A summary of the calculation of the MFCC is depicted in figure 4.3.

4.4 Recognition units and types

When building a speech recognizer, an important decision, one has to initially make to choose the recognition unit. This unit can be of different lengths, and each choice has its own benefits and disadvantages. The longer the unit is the more accurately it will models the effects of context-dependency, but more training data will be required. The unit should be consistent and trainable. Different instances of the same unit should have similarities, and there should be enough training examples to reliably estimate the model parameters. The following discussion is based on.

4.4.1 Words

Words are the actual unit we eventually want to recognize. They are thus a natural choice for the speech recognition unit. They inherently take context-dependency issues into account, since phonemes inside a word are very likely the same in each utterance. The obvious problem with word models is the large number of different words and the problem of too little training data. This makes word models unusable for large vocabulary recognition.

Sometimes word models are used together with other kinds of models. For example, short function words may have their own models implemented within a phoneme recognizer.

4.4.2 Phonemes

To facilitate larger vocabularies than what are supported by word models one has to find a way to share information between different models. In other words, one has to take sub word units into use. The most obvious choice is the phoneme. The number of phonemes is moderate, ranging between 15 to 50 in most languages. Therefore, it is very easy to get enough training examples for each phoneme. However, the obvious problem is that the context of the phonemes is ignored. Thus, while word models lack generality, phoneme models over generalize.

4.4.3 Multiphone units

Larger units of speech can be used as recognition units to model the coarticulatory effects, for example, syllables or demisyllables. They solve most of the

problems involving context-dependency leaving the middle phonemes of the unit out of the effect of the neighboring units. However, the edges of these units undergo some coarticulatory variation. Furthermore, there may not be enough data to train rare units.

4.4.4 Explicit transition modeling

Biphones model pairs of phonemes without the use of stationary phonemes. They are used for explicit transition modeling. Another approach is to use stationary phoneme models and create explicit models for the transitions. These approaches suffer from the problem of the large number of different models: with N phonemes there may be up to N 2 transitions.

4.4.5 Word-dependent phonemes

Word-dependent phonemes are a compromise between word modeling and phone modeling. Phoneme models are trained for each word separately, but if a specific word phoneme is poorly trained, the parameters can be interpolated from the phoneme models in other words. Therefore, not all words have to be present in training, and new words may be added later.

Chapter 5

Basic Structure of Hidden Markov Models

Despite their many weaknesses, Hidden Markov Models (HMM's) are the most widely used technique in modern speech recognition systems. This is due to the fact that a great deal of effort has been devoted in research during the 1980's and 1990's, making it very challenging for alternative methods to get even close to their performance level with moderate investments [16].

Markov models were introduced by Andrei A. Markov and were initially used for a linguistic purpose, namely modeling letter sequences in Russian literature. Later on, they became a general statistical tool. Markov models are finite state automatons with probabilities attached to the transitions. The following state is only dependent on the previous state.

Traditional Markov models can be considered as 'visible', as one always knows the state of the machine. For example, in the case of modeling letter strings, each state would always represent a single letter.

However, in hidden Markov models the exact state sequence that the model passes through is not known, but rather a probabilistic function of it.

Hidden Markov models can be built for parts-of-speech differing in size. Usually, simple, small-vocabulary recognizers use word models and larger-vocabulary recognizers use phoneme models [15].

5.1 Definition of HMM's

The Hidden Markov Model is a finite set of states, each of which is associated with a probability distribution. Transitions among the states are governed by a set of probabilities called transition probabilities. In a particular state an outcome or observation can be generated, according to the associated probability distribution.

For defining HMM's, the following elements are needed:

• The number of states, N.

• The number of elements in the observation alphabet, M . The alphabet can also be infinite (continuous).

• Transition probabilities aij between the states. These are usually presented in a transition matrix.

• A probability distribution associated with each state: B = bj (k). For continuous probabilities, the probability density function can be approximated by a sum of Gaussian mixtures (N), each having their own weighting coefficients cjm, mean vector μjm, and variance vector Σjm. With these defined, the probability of an observation vector Ot can be calculated according to the equation 4.4.

$$b_j(k) = \sum_{m=1}^{M} c\,\mathcal{N}(\mu_{jm}, \Sigma_{jm}, O_t) \qquad (4.4)$$

In this example, each emitting state has transitions to itself and the next state. This simple left-to-right structure is quite common in speech recognition. Sometimes transitions for skipping a state are used and sometimes even backward transitions may exist.

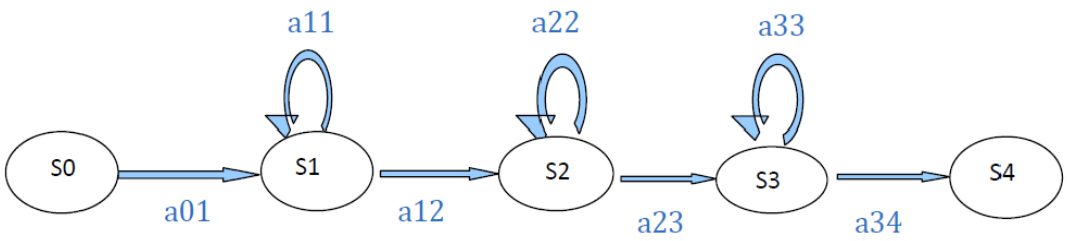


Figure: 5.1 An example of a hidden markov model.

The mathematical problems of HMMare quite complicated and have been covered in several good papers before, e.g. [18], so they will not be covered in detail here. In short, the three main problems that have to be solved for utilizing the models in speech recognition are:

1. The evaluation problem

Given a model and a sequence of observations, one needs to compute the probability that the sequence was produced by the model. This can be also seen as the

problem to score the match between the observations and the model. For this problem an exact solution exists and can be efficiently calculated by using the forward-backward algorithm [16].

2. The estimation problem

Given an observation sequence or a set of sequences, this problem involves finding the parameter values that specify a model most likely to produce the given sequence. This problem is involved in speech recognition in the training phase, and is solved iteratively using the Baum-Welch algorithm [17].

Two approaches can be used in training, maximum likelihood estimation (MLE) and maximum mutual information estimation (MMIE). The goal in MLE is to maximize the probability of the model of generating the training sequences, while MMIE-training tries to maximize the ability of the model to discriminate between models of different speech classes (e.g. Phonemes).

3. The decoding problem

The third problem involves finding the most likely state sequence for a given observation sequence. There are different search algorithms for this, for example, the beam search algorithm.

The HMM concept has several weaknesses. As its name suggests, the Markov assumption, i.e., the probability of being in a given state at time t depends only on the state at time $t - 1$, is used. This is definitely not true for speech signals. Another assumption not true for speech signals is that successive observations should be independent. [16]

5.2 Language models and lexicons

An ASR system that only utilizes acoustic features ignores a lot of linguistical information involved in the spoken text. Even though perfect modeling of linguistical information is not possible (e.g. speakers do not strictly follow the grammatics of a language), using statistical language models often radically improves the performance of an ASR system.

Lexicon is a list of allowed words in a task. The larger the lexicon is and the more there are words that sound similar the harder the recognition task is.

Language models are used to restrict the possible word combinations based on statistical information collected from large text material. The most common approach is using stochastic descriptions of text usually involving the likelihoods of the local sequences of one to three consecutive words in training texts. Given a history of prior words in a sentence and based on the statistical model the number of words that need to be considered is lower than the size of the vocabulary. For successful language modeling the text on which the language model is based on should be about similar topic as the data that is going to be recognized. Language features, such as word inflection, also affects the success of language modeling.

Chapter 6

Introduction to Sphinx

We used the SPHINX [2] Speech Recognition System provided by the Carnegie Mellon University to implement the HMM based speech recognition. The SPHINX system is an open source tool which includes all the trainers, decoder's acoustic models and language models required to build and test a complete recognition system. The tool provided by SPHINX has a wide variety of applications which cater to different needs of the user.

VERSION	FEATURES
SPHINX-2	Semi-continuous and continuous output probability density functions. Tree Lexicon
SPHINX-3	Continuous output probability density functions. Batch processing Flat Lexicon LVCSR System
SPHINX-3.X (S3-fast)	Continuous output probability density functions. Batch and Live mode processing Tree Lexicon
SPHINX-4	Continuous output probability density functions. Batch and Live mode processing JAVA platform

Table: 6.1 Different versions of SPHINX trainers and decoders

The main advantage of SPHINX [3] package is that it is fully customizable for a range of different applications. The user can carefully choose the version depending on the types of models his or her working with and the level of accuracy required.

We used sphinx-III, which is a speaker independent large vocabulary speech recognition system to perform the recognition task. The procedure was realized on a

LINUX/UNIX plat form which has inbuilt C-compiler, Perl and audio converter which made the task easier. The recognition was performed in a batch mode which required the speech input to preprocessed in a cepstral format to be compatible with the SPHINX acoustic trainer and decoders. For a batch mode processing the entire input that has to be recognised must be available beforehand. The pre-record speech has to be processed from its raw format into cepstrum files which is compatible with the SPHINX trainer and decoder. The HMM based system uses the trainer to learn the characteristics of the speech input and the decoder to deduce the most probable sequence of sound units for a given speech input.

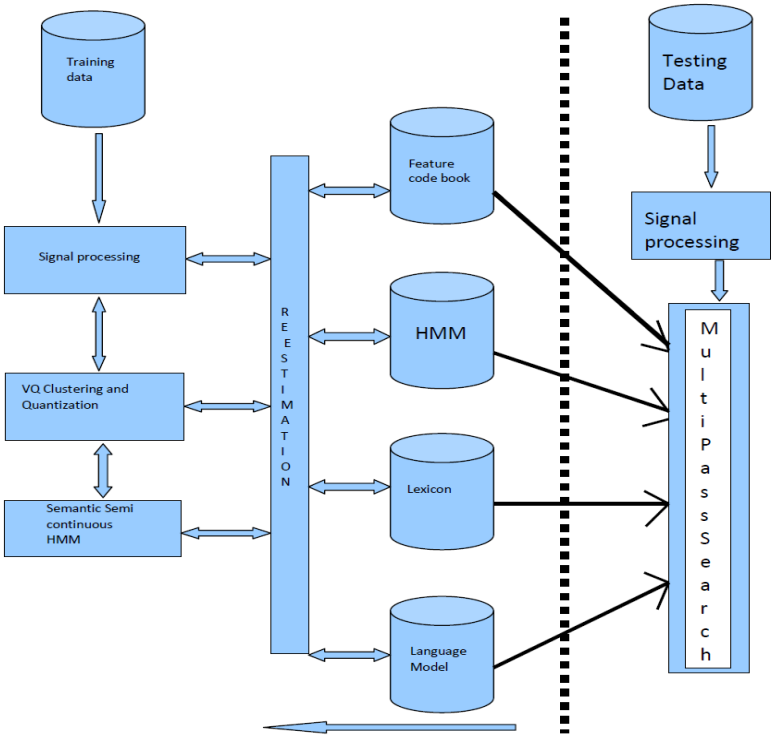


Figure: 6.1 A Block Diagram of SPHINX-III Architecture.

6.1 Sphinx Trainer

SPHINX trainer learns the characteristics of the sound modules using a set of programs. The training database which comprise of test data, pronunciation dictionary, filer dictionary and the transcripts files are provided to the trainer. The trainer then functions by mapping each word to a sequence of sound units in the dictionary and derives the sequence of sound units associated with each signal. The generates model index files which contains reference to states in the HMM modules

which helps both the trainer and decoder to access the right parameters. Using the open source trainer and the relevant input files acoustic models can be built for the speech recogniser.

The training procedure involves optimizing HMM parameters of a given training data. An iterative procedure, the Baum-Welch or forward-backward algorithm is employed to estimate transition probabilities, output distribution, and codebook means and variance under the probabilistic framework.

6.1.1 Pronunciation Lexicon

A pronunciation lexicon (or dictionary) files specifies word pronunciation. In Sphinx, pronunciation is specified as a linear sequence of Phonemes. Each line in the file contains one pronunciation speciation, except that any line that begins with a "#" character in the first column is treated as a comment and is ignored. The lexicon is completely case-insensitive (unfortunately).For example, it's not possible to have two different entries Brown and brown in the dictionary. Example of pronunciation Lexicon:

RAM R AX M

THAM TH AA M

For multiple pronunciations the words are distinguished by a unique parenthesized suffix for the word string like,

LA L AA

LA (2) L AX

Compound words are represented as

WANT_TO W AA N AX

6.1.2 Main and Filler Lexicons

The sphinx decoders actually need two separate lexicons: a "regular" one containing the words in the language of interest, and also a filler or noise lexicon .The latter defines "words" not in the language. More specifically, it defines legal "words" that do not appear in the language model used by the decoder, but are nevertheless encountered in normal speech. This lexicon must include the silence word <sil>, as

well as the special beginning-of-sentence and end-of-sentence tokens <s>, and </s>, respectively. All of them usually have the silence-phone SIL as their pronunciation. In addition, this lexicon may also contain "pronunciation" for other noise event words such as breath noise, "UM" and "UH"sounds made during spontaneous speech, etc.

6.1.3 Acoustic Model

Sphinx-3 is based on sub phonetic acoustic models [14]. First, the basic sounds in the language are classified into phonemes or phones. There are roughly 50 phones in the English language. For example, here is a pronunciation for the word.

DHITHXVAAR DH IX THX V AA R

For most applications, one builds acoustic models for triphones, qualified by the four position attributes. Each triphone is modeled by a hidden Markov model or HMM. Typically, 3 or 5 state HMM is used, where each state has a statistical model for its underlying acoustics. But if we have 50 base phones, with 4 position qualifiers and 3-state HMM, we end up with a total of 50^3*4*3 distinct HMM states! Such a model set would be too large and impractical to train. To keep things manageable, HMM states are clustered in to a much smaller number of groups. Each such group is called a senone (in sphinx terminology),and all the states mapped into one senon share the same underlying statistical model.

Each triphone also has a state transition probability matrix that defines the topology of it's HMM. Once again to conserve resource, there is a considerable amount of sharing. Typically there is one such matrix per base phone derived from the same parent base phone share its state transition matrix

6.1.3.1 Acoustic Model Training

The acoustic models are not built directly on raw audio file. Instead, the audio is processed to extract of relevant features. All acoustic modeling is carried out in terms of such feature vector.

Input: Speech wave form 16-bits (sampling rate 16 kHz)

Input: Front and processing parameters

Pre-emphasis module (pre-emphasis alpha =0.97)

Framing (100 frames/sec)

Power spectrum (using DFT size 512)

Filtering (lower = 133.334 Hz, upper = 6855.4976 Hz)

Mel Spectrum (multiplying the power Spectrum with the Mel

weighting filters(number of mel Filters 40)

Mel Cepstrum (number of cepstra 13)

Mel Frequency cepstral Coefficients (39 32-bit floats)

6.1.3.2 Acoustic Modeling Training

This refers to the computation of a (statistical) model for each senone in the mode as a rough approximation, this process can be described by the following conceptual steps:

Obtain a corpus of training data. This may includes thousands of sentences (or utterance, in Sphinx jargon), consisting of the spoken text and corresponding audio sample stream.

For each utterance, convert the audio data to a stream of feature vectors as described above.

For each utterance, convert the text into a linear sequence of tri phone HMM using the pronunciation lexicon.

Find the best state sequence or state alignment through the sentence HMM, for the corresponding feature vector sequences.

The best state sequence is one with the smallest mismatch between the input feature vector and the labeled saneness underlying statistical models.

For each senopne, gather all the frames in the training corpus that mapped to that senone in the above step, and build a statistical model for the corresponding collection of feature vectors.

The circularity obtained is resolved by using the iterative baum-welch or forward-backward training algorithm. The algorithm begins with some initial set of models, which could be completely flat, for the senones. It then repeats the last two

steps several times. Each iteration use the model computed at the end of the previous iteration.

6.1.4 Language Model

The main language model (LM) used by the Sphinx decoder is a conventional bigram or trigram back off language model. The CMU-Cambridge SLM toolkit [15] is capable of generating such a model from LM training data. Its output is an ASCII text file. But a large text LM file can be very slow to into memory. To spend up this process, the LM must be compiled into a binary form.

A trigram model may be trained simply by using the equation:

$$P(\omega_1|\omega_2, \omega_3) = \frac{f(\omega_1|\omega_2,\omega_3)}{f(\omega_1,\omega_2)} \quad (6.1)$$

Here $f(\omega_1, \omega_2, \omega_3)$ refers to the frequency of occurrence of the trigram$(\omega_1,\omega_2, \omega_3)$ in the training text and $f(\omega_1, \omega_2)$ refers to the frequency of occurrence of the bigram(ω_1, ω_2).

6.1.4.1 Unigrams, Bigrams, Trigrams, LM Vocabulary

A trigram LM primarily consists of the following:

Unigrams: The entire set of words in this LM, and their individual probabilities of occurrence in the language. The unigram must include the special beginning-of-sentence and end-of sentence tokens :<s>,and </s> respectively.

Bigrams: A bigram Is mathematically P(word2|word1).That is the conditional probability that word2 immediately follows word1 in the language. An LM typically contains this information for some subset of the possible word pairs. That is, not all possible word1 word2 pairs need be covered by the bigrams.

Trigrams: Similar to bigram is P(word3|word1,word2),or the conditional probability that word3 immediately follows a word1, word2 sequence in the language. Not all possible 3-word combinations need be covered by the trigrams.

The vocabulary of the LM is set of words covered by the unigrams.

The LM probability of an entire sequence is the product of the individual word probabilities.

For example, the LM probability of the sentence “HOW ARE YOU” is

P (HOW|<s>)*

P (ARE|<s>, HOW)*

P (YOU|HOW, ARE)*

P (</s>|ARE, YOU)

6.2 Sphinx Decoder

The SPHINX decoder has a set of programs in order to perform the recognition task. The input files required for the decoding procedure includes trained acoustic models, pronunciation dictionary, filler dictionary, language model and the test data in the right format as explain above. With any given set of acoustic models, the corresponding modules index file must be used for decoding. Both the trainer and decoder process test data in the form of feature vectors. These features vectors are generated from converting the acoustic signals to a cepstral format using the front-end executable provided with the SPHINX training package.

6.3 Banjara Language Speech Recognition

In this chapter we will discuss about the Banjara Language and the main work is done in the project. A complete experimental procedure is given. The working of the machine translation is explained.

The below fig represent the proposed speech to speech system for developing Banjara to Telugu language online dictionary. The spoken word is a Banjara language and the corresponding meaning is uttered by the system in Telugu Language.

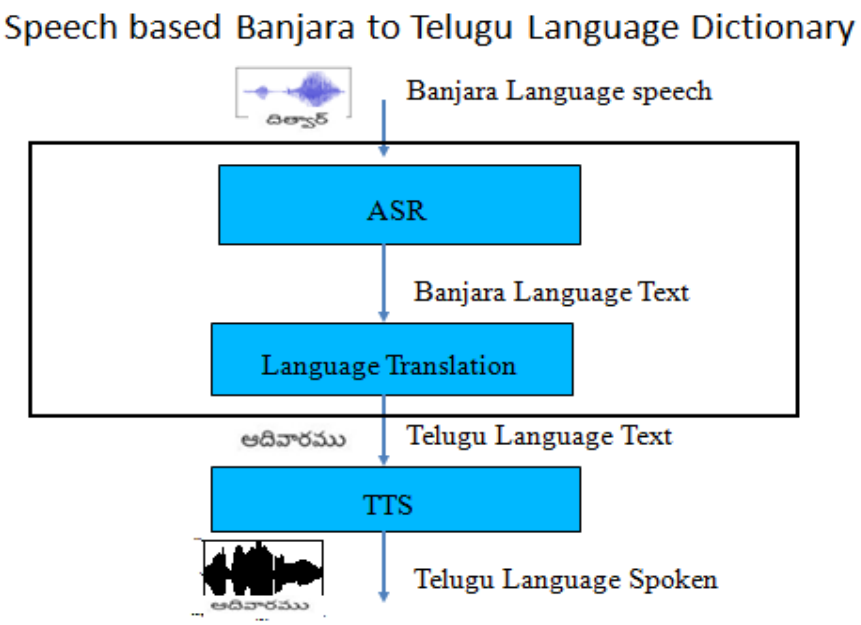


Figure6.2: Shows the proposed system Architecture for building Dictionary

6.3.1 Speech Database

The first step we followed in creating a speech database for building an Automatic Speech Recogniser (ASR) is the generation of an optimal set of textual isolated words to be recorded from the native speakers of the language. Since this is the first time any work is being done for Banjara Language. We had tough time in collecting the speech sample. Speech sample were collected from different people through normal microphone recording.

The database consists of 1000 different commonly used words, numbers from one to hundred and 500 names of Banjara Language recorded several time by a single speaker. So the database consists of around 1500 files. Care has been taken to record the speech with minimal background noise and error in communication, but due to insufficient knowledge of the people while recording the voice for speech recognition some errors have crept in while recording. These errors had to be identified manually by listening to the speech and unwelcome speech wave discarded.

6.3.1.1 Data Preparation

As Sphinx supports the speech signal should be PCM bit Mono, the speech signal in the data will be 16 bit Mono. The data wav files recorded through mobile were converted to the required format in 16 kHz, 16 bit mono format using the PRAAT software speech tool and using arecord or rec command in Linux. And the normal recordings were recorded using the close in microphone in the required format.

The recorded wave files were then converted to raw format to remove the unwanted information, the header file, from the wave file in speech recognition. The experimental procedures are explained step by step-in the following ways.

6.4 Experimental procedure

First, the data is obtained, for the experiment in this work, are 1000 Banjara Language commonly used isolated words and numbers from 1 to 100 were recorded. As explained earlier the data are then prepared to the desired format as expected to be in Sphinx. Since these experiments are done using CMU.s Sphinx the experimental procedure is thus the procedure for Sphinx.

A complete guide to working in Sphinx is discussing step by step.

1. SPHINX can be download from the website
   http://www.cs.cmu.edu/~robust/Tutorial And download singlemachine.tar.gz
2. Unzip the Downloaded file using the command
   Tar –zxvf singlemachine.tar.gz.

This will give the Dictionary name called TUTORIAL in this base dictionary it will install all the files necessary to train and test SPHINX.

3. In TUTORIAL create new experiment to do by the command
   ./create_newexpt.csh <expt_name>
4. In this **expt_name** the entire file required to train and decode the system will be loaded. Copy the **util** dictionary from the Sphinx-III to **expt_name**

Now by giving plain text to the site www.speech.cs.cmu/tool/lmtool.htm get Dictionary File (Make necessary changes), LM File.

Create the Transcription file, Phoneme list and mfc.ctl file i.e, the file contains the complete path of all the MFCC features of the wav files.

6.5 Training

The important components for training in Sphinx are

- Transcription file
- Lexicon Model
- Acoustic Model
- Language Model
- Phoneme list

Transcription File

A transcription file is where the corresponding textual representation of the wav files are listed followed by the file name of the wav file. A general transcription file shown below.

<s> DHITHXVAAR </s> (ram001.wav)

<s> SOAMAVAAR </s > (ram002.wav)

<s> MAMGHALHAVAAR </S> (RAM003.WAV)

Here <s> and </s> are the beginning and ending of file.

DHITHXVAAR is the word being recorded a speaker, the file name is ram001.wav

<SIL> is added in the word wherever required when there is a presence of SILENCE in the recording of the word.

Transcription file is very important as it map the wav file with its corresponding text. Misrepresentation of this file may cause the system not work or give poor recognition accuracy.

<s> DHITHXVAAR </s> (ram001.wav)
<s> SOAMAVAAR </s> (ram002.wav)
<s> MAMGALHAVAAR </s> (ram003.wav)
<s> BADHVAAR </s> (ram004.wav)

<s> VARASPATHX </s> (ram005.wav)

Lexicon Model:

The lexical or pronunciation model contains pronunciations for all the words of interest to the decoder. Sphinx-III uses phonetic units to build word pronunciations. Currently, the pronunciation lexicon is almost entirely hand-crafted. One can build the Lexical model using the the CMU's SLM Toolkit [15].However the tool is build to build English pronunciation dictionary. Hence we have to handcraft the dictionary based on the pronunciation of Banjara Language.

UOH	
DHITHXVAAR	DH IX THX V AA R
SOAMAVAAR	S OA M AX V AA R
MAMGALHAVAAR	M AX M G AX LH AX V AA R
BADHVAAR	B AX DH V AA R
VARASPATHX	V AX R AX S P AX THX
SAKRAVAAR	S AX K R AX V AA R
TTHAAVAR	TTH AA V AX R
DHAADHIX	DH AA DH IX
DHAADHAA	DH AA DH AA
NAANAA	N AA N AA

CMU	
DHITHXVAAR	D HH IH TH S V EY AH R
SOAMAVAAR	S OW M AE V EY AH R
MAMGALHAVAAR	M AE M G AE L HH AE V EY AH R
BADHVAAR	B AE D HH V EY AH R
VARASPATHX	V AH R AE S AH TH S
SAKRAVAAR	S AE K R AE V EY AH R
TTHAAVAR	T TH AA V AH R
DHAADHIX	D HH AA D HH AH K S
DHAADHAA	D HH AA D HH AH
NAANAA	N AA N AH

Table: 6.2 Shows the different phonemes in two different Dictionary

Acoustic model

Sphinx uses acoustic models based on statistical hidden markov models (HMM). The acoustic model is trained from acoustic training data using the sphinx-III trainer. The trainer is capable of building acoustic models with a wide range of structures, such as discrete, semi-continuous, or continuous. The acoustic model is built using the CMU SLM Toolkit [15] provided by CMU.

Language Model

Sphinx-III uses a conventional back off bigram or trigram language model. CMU provides a toolkit to build a Language Model [15].

```
[rambabu@localhost project1]$ ./lm3g2dmp /home/rambabu/TUTOR
IAL/project1/hcu.lm /home/rambabu/TUTORIAL/project1/
INFO: lm3g2dmp.c(708): Reading LM file /home/rambabu/TUTORIA
L/project1/hcu.lm (name "")
INFO: lm3g2dmp.c(730): ngrams 1=530, 2=1051, 3=528
INFO: lm3g2dmp.c(428): Reading unigrams
INFO: lm3g2dmp.c(754):      530 = #unigrams created
INFO: lm3g2dmp.c(473): Reading bigrams
INFO: lm3g2dmp.c(766):     1051 = #bigrams created
INFO: lm3g2dmp.c(767):        4 = #prob2 entries
INFO: lm3g2dmp.c(774):        3 = #bo_wt2 entries
INFO: lm3g2dmp.c(554): Reading trigrams
INFO: lm3g2dmp.c(783):      528 = #trigrams created
INFO: lm3g2dmp.c(784):        2 = #prob3 entries
INFO: lm3g2dmp.c(910): Dumping LM to /home/rambabu/TUTORIAL/
project1//hcu.lm.DMP
```

Training procedure

1. To compute MFCC run the script in c_script by the command

c_scripts#] ./compute_mfcc.csh <Path of raw file or MFCC file>

2. Change the paths of file in the file called ***variable.def***

Contains in the c_scripts

#Input to the Trainer.

Set listoffiles =$base_dir/train/mfc.ctl

Set transcription file =$base_dir/train/BanjaraLang.trans

Set dictionary =$base_dir/train/UOH.dic

Set fillerdict =$base_dir/lists/filler.dict

Set phonefile =$base_dir/train/BanjaraLang.phonelist

Here base_dir is my experiment dictionary and I have all my files required for training in a TRAIN folder.

3. Go to c_script file, change the command for all the scripts contained in the directories 01* through 05*, by kept the Unlimited as comment (#) line

4. Then run the slave *.csh script within the directory starting from 01* through 05*. If the input if error free then we get number of files by running the each script. The errors will be checked in the dictionary

5. Training will be completed if there is no error in the set up.

6.6 Decoding

For decoding pronunciation dictionary, LM, and sentence file for computing the accuracy of the recognition engine are required.

Sphinx-III actually doesn't support the text ARPA file format. So we need to convert the LM file to .DMP file using a tool called *lm3g2dmp [15]* to convert the text file to .DMP file format. it can be downloaded it from CMU website.

Then conversion of the LM could be run as

Lm3g2dmp<.arpa file> <destination>

A file named something.arpa.DMP will be created.

Now the system is ready for Decoding and obtaining the result.

Decoding procedure:

1 create a dictionary called test (for self understanding)

2 convert the yourfile.lm to .DMP [20] file by running the command

Lm3g2dmp <yourfile.lm> <destination path>

3 copy the sentence file into this directory say mywavdir

Change the file path for decoding in variable.def

#Input to the decoder

Set ctlfn =$base_dir/test/mfc.ctl

Set testref =$base_dir/test/BanjaraLang.sent

Set cepdir =$base_dir/feature_files

Set lmfile =$ base_dir/train/BanjaraLang.lm.DMP

Set dictfn =$ base_dir/train/UOH.dic

Set fdictfn =$base_dir/lists/filler.dict

5 now decode by going to decode folder and running the command

./launch_decode.ci.1guamodels

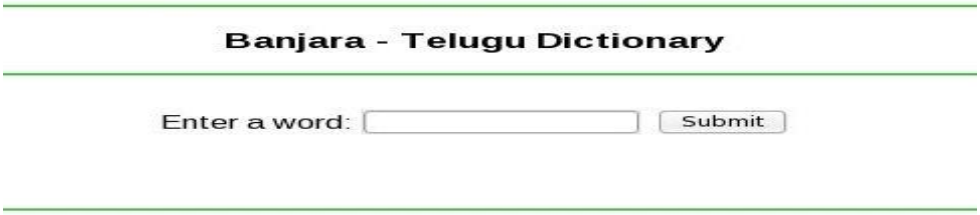
To view the recognised word go to Decoding/Result

6 To compute the word accuracy

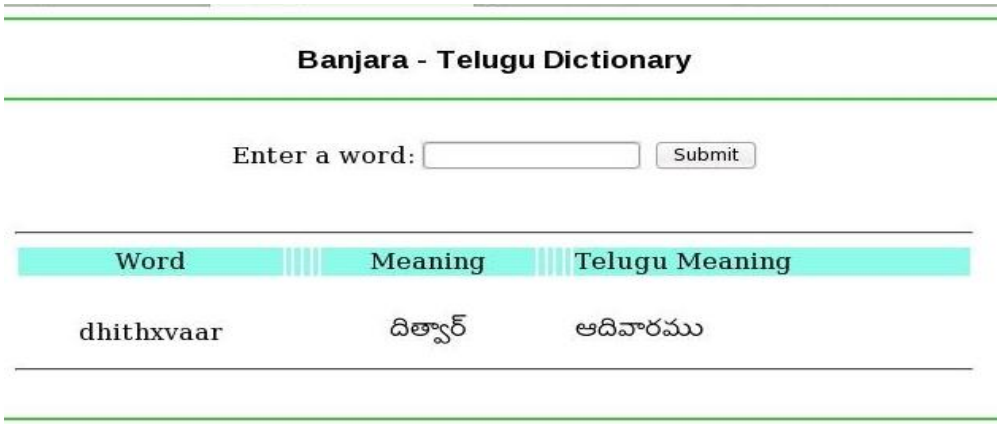
./compute_acc.ci.csh

6.7 Banjara Language Dictionary

In this section we briefly discuss about the dictionary creation and connectivity of the speech part.



In the above GUI you will connect with the system and with the help of the microphone a person can pronounce a word after that the ASR system takes an analog signal and the system generates the text corresponding to that uttered word, after that the Java API is takes the system generated text as search field to the connected database which is MySql and the system should search for the given word, once the word found then the mapping could be takes place i.e. the Banjara Language word to be converted into Telugu Language and the corresponding meaning should display on screen with the Telugu letters (The Google transliteration) will taking care of the conversion of the letters. The below GUI shows the corresponding meaning and the letters



The above shows the output of the uttered word and the corresponding Telugu meaning is displayed on the screen. Here the translation has been done.

Chapter 7

Result and Analysis

This chapter includes the various experiments and its results being performed in the project work. In the first section we will explain the different types of experiments carried out. In the next section it will discuss the analysis of the experiments done.

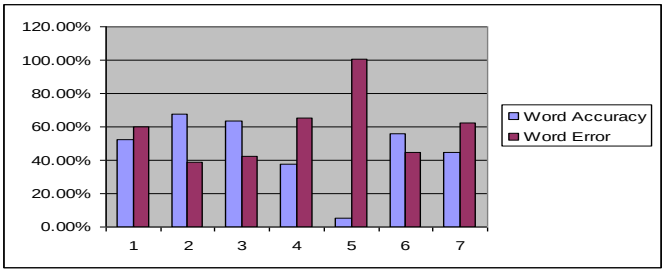
7.1 Experiment and result

7.1.1 Speaker Dependent and Independent Recognition system performance

In this Experiment 1: There are 1000 isolated words from banjara language collected from different speakers both native and non native category. Compared the Lexicon of CMU which uses 39 phonemes and American pronunciation, collected from CMU dict 4.0 online Knowledge base systems. The HCU lexicons are using 51 phonemes representing the Telugu alphabets and pronunciation dictionary is manually handcrafted. The given Table NO.1 shows the comparison of ASR system performance by using these two pronunciation dictionaries in terms of Word Accuracy and Word Error Rate of ASR system.

	HCU		CMU	
	%WA	%WER	%WA	%WER
Rambabu	81.70%	21.40%	52.48%	60.12%
Kishore	88.20%	11.82%	67.49%	38.92%
Ravindar	82.67%	29.67%	63.56%	42.45%
Veeranna	85.12%	24.56%	37.62%	65.35%
Sreenu	81.50%	25.57%	5.50%	100.50%
Srinu	93.00%	7.00%	56.00%	45.00%
Naik	83.33%	16.67%	45.00%	62.50%

Table1: Shows the overall performance of both Dictionary (CMU and UOH)

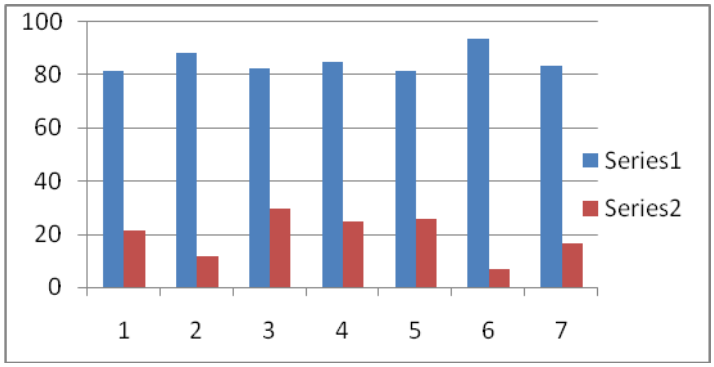


Graph: 7.1 Shows the individual speech recognition performance using CMU dictionary.

Now we use our modified dictionary for training and testing purpose for the same data as the above experiment. We find the word recognition accuracy slightly higher for this experiment. The graph of the result is shown below.

1	81.7	21.4
2	88.2	11.82
3	82.67	29.67
4	85.1	24.56
5	81.5	25.57
6	93.5	7
7	83.33	16.67

Table2: Shows the UOH dictionary performance



Graph: 7.2 Shows the results using the modified (UOH) dictionary.

Thus the accuracy improves on the Speaker dependent System when the dictionary is hand crafted for Banjara Language.

In speaker independent speech recognition is where the system should act as independent for any speaker. The testing data of the system may not be present in the training process. Here again, the same number of speaker is taken as the experiment done above. Again we used the same format for the experiment. Different speakers are used for training the system and testing the system. The result obtain for the experiment are.

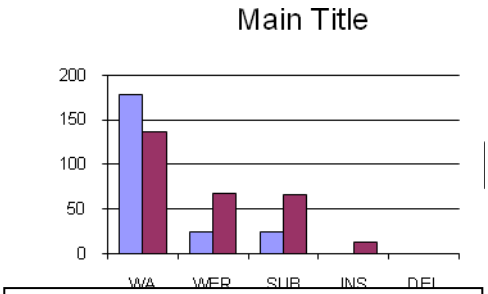
7.2 Result and analysis

Performing various experiments, the result accuracy increases when we used our modified (UOH) dictionary based on the Banjara Language pronunciation. The accuracy also increases when we increase the number of speakers. The accuracy improves when the training data is more. The correct format of recording is also desirable for better recognition accuracy.

The table below gives the summary of the experiment performed with the word recognition accuracy.

	UOH	*CMU*
WA	*179*	*136*
WER	*24*	*67*
SUB	*24*	*66*
INS	*0*	*13*
DEL	*0*	*0*

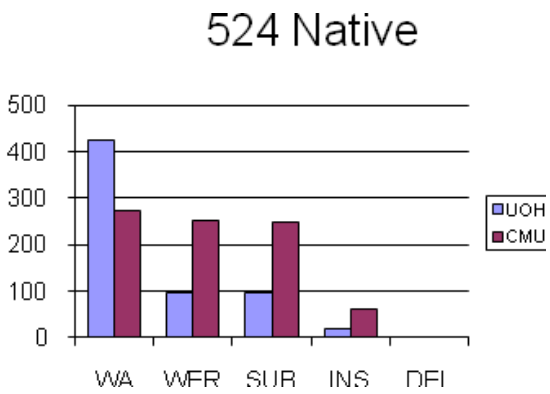
Table 3: recognized words for 200 words



Graph 7.3: comparision of 200 words with CMU and UOH lexicon

Column1	**524 Native**	**Column2**
	UOH	CMU
WA	427	274
WER	97	250
SUB	96	249
INS	16	60
DEL	0	0

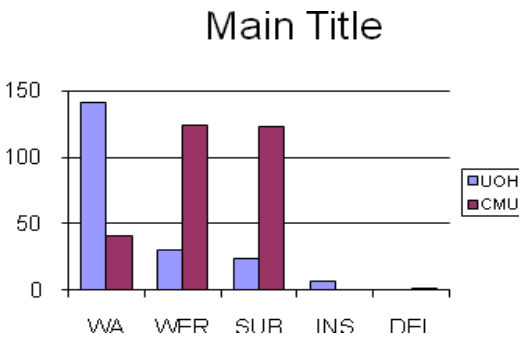
Table4:: Native speaker with 524 words



Graph 7.4: comarision of UOH and CMU lexicon of 524 words of native spkr

Column1	**165 Non Native**	**Column2**
	UOH	CMU
WA	141	41
WER	30	124
SUB	24	123
INS	6	0
DEL	0	1

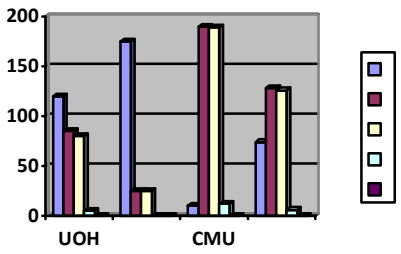
Table5: Non native speaker data



Graph 7.5: native speaker performace comparision

Column1	Column2	Column3	Column4	Column5
	200 Native			
	UOH		CMU	
WA	120	175	10	74
WER	85	25	190	128
SUB	80	25	189	126
INS	5	0	12	6
DEL	0	0	0	0

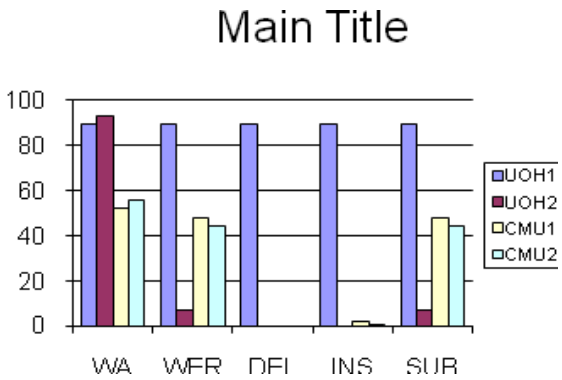
Table6:Both native and non native spkr comparison



Graph 7.6: recognition comparision graph with error analysis

Column1	F:UOH	M:UOH	F:CMU	M:CMU
WA	89	93	52	56
WER	11	7	48	44
DEL	0	0	0	0
S	0	0	2	1
SUB	11	7	48	44

Table7: Gender variant comparision



Graph 7.7: Gender variant comparison for two lexicons

To check the performance of the developed ASR system the speakers are considered in different category like native and non native, then male and female. Also taking the CMU lexicon where only 39 phonemes are used to represent the pronunciations of the build ASR system. This is an online knowledge base tool which gives automatically pronunciation dictionary for given set of words. As CMU dictionary is built by considering American accent based English, the performance of Target language ASR system performance will degrade by insertion, deletions i.e. substitution errors even for clean speech. Hence an attempt is made to use the Indian language based pronunciation dictionaries to reduce the substitution errors. One such attempt is using UOH [19]

Pronunciation dictionary where the phonemes are represent the Telugu language phonemes i.e total 51 in number. The different data conditions experiments shown from Table 1 to Table 7 and Graph no. 1 to graph no. 7, the performance of ASR system in terms of Word Accuracy and Word Error rate with comparison of CMU and UOH based lexicon. From the above result it is observed that in all aspects the UOH lexicon is giving more Word accuracy than the CMU. Here UOH lexicon are handcraft pronunciation dictionary.

Chapter 8

Conclusion and Future work

8.1 Conclusion

Today the technology of speech recognition is gaining with the popularity of ubiquitous computer and speech as a communication modality. The multilinguality in speech recognition is another challenging area for speech recognition in India. In this work Banjara language is used for recognition. An application of translation of Banjara to Telugu language is build in this project work. The lexicon models, creation of pronunciation dictionary and phoneme list are areas where we focused most. One can have an easy using this application of translation.

With variant data the performance of the Developed BASR system is giving better word accuracy for the Telugu phoneme based UOH lexicon than Enlish phoneme based CMU dictionary. It is observed around 93% word accuracy for native Banjara speaker for UOH dictionary and with same data but CMU dictionary it is only giving 56 % WA. Non-native female speaker's data UOH lexicon, the WA is 89% and CMU lexicon it 52 % WA. It is also observed that more than 50% substitution errors are reduced by handcrafted UOH lexicon. Hence Telugu phoneme based BASR system is giving better performance than the CMU phonemes. The observation also done by collecting the different environmental noisy conditions to bring the robustness to the system.

8.2 Future work

For getting a good accuracy in speech recognition have a large speech corpus is essential. Since this is the first time a research on speech recognition for Banjara Language is done, collection of data was difficult. Hence collection of large speech corpus for Banjara language will be one of the future works. This work is performed using the HMM based Sphinx-III speech recognition Engine. Research using ANN and other toolkits will be another scope for future. Translation from banjara to Telugu Language is the application work for this project. A more comprehensive research and better translation with large speech corpus, and wide varieties of sentence file involved will be the extension work for this project.

References

[1] W.A.Lea, "Trends in speech Recognition", Prentice Hall, 1980.

[2] http://www-2.cs.cmu.edu/~robust/TUtorial

[3] Internet accessible speech recognition technology. http://www.cavs.msstate.edu/hse/ies/projects/speech/index.html.

[4] Samudravijaya K and Kumar and Maria Barot. A Comparison of public domain software tools for speech recognition. Pages 125-131. Workshop on spoken language processing, 2003.

[5] N.Rajput, M.Kumar and A.Verma, A large vocabulary continuous speech recognition system for Hindi. IBM Journal for Research and Development.

[6] Davis, S., Mermestein , P.,"Comparison of Paramatric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences" IEEE Transactions on Acoustics, Speech, and Signal Processing , v. ASSP -28,n.p

[7] www.speech.cs.cmu.edu

[8] A article in The Hindu paper

[9] J. Laver. Principles of Phonetics. Cambridge University Press, 1994

[10] D. O'Shaughnessy. Speech Communications – Human and Machine. IEEE press, second edition edition, 2000

[11] S. S. Stevens. On the psychophysical law. Psychological review, 64:153–181 1957.

[12] S. Buus and M. Florentine. Modifications to the power function for loudness In Fechner Day 2001. Proc. of the 17th Annual Meeting of the International Society of Psychophysics, pages 236–241, 2001

[13] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Stat., 37:1554–1563, 1966

[14] F. Jelinek, L. R. Bahl, and R. L. Mercer. Design of a linguistic statistical decoder for the recognition of continuous speech. IEEE Trans. Information Theory, IT-21:250–256, 1975

[15] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286, 1989

[16] B. H. Juang and L. Rabiner. Hidden Markov models for speech recognition Technometrics, 33(3):251–272, 1991.

[17] K.-F. Lee, H.-W. Hon and R. Reddy. An overview of the SPHINX speech recognition system. IEEE Trans. Acoust. Speech Signal Processing, 38(1):35–45, April 1990.

[18] M.-Y. Hwang. Subphonetic Acoustic Modeling for Speaker-Independent Continuous Speech Recognition. PhD thesis, Carnegie Mellon University, 1993

[19] Abjit Debbarma. "Speech recognition for kokborok language" Master's thesis 2008

[20] N. Rambabu. "Banjara language Tulaanatmaka parishilana". M.phill thesis 2008

[21] Gona Naik. "Bharata desham lo Banjaraalu". Telugu Academy book.

[22] M. Nagamani, B.S.R Krishna. "Speech interactive system for physically challenged people to fill railway reservation form in Telugu language" conference paper-2009.

[23]Pamela_McKenzie_long_paper_RamU1. "Multilingual Education among Minority Language Communities" .

[24] K. Ramesh kumar. "Multilingual Education project of AP". India workshop on multilingual education with special focus on tribal education. 25-27 oct-2005 @CIIL Mysore.

[25] Leila Schroeder. "Bantu Orthography Manual", SIL e-Books 9 @2008 SIL international Library of congress.

[26] Markku Ursin. "Triphone clustering in Finnish continuous speech recognition" July-2002 A Masters thsis.

Sphinx-III User manual

HOW TO RUN SPHINX-III

The procedure, how to run Sphinx-III as shown below.

/**** Procedure Starts****/

1. First go to the Path, where the Sphinx-III

TUTORIAL is load by using the Command prompt.

[root@localhost ~]# cd /home/.../TUTORIAL/

2. Then we create new experiment name as "example" by using the below Command prompt.

[root@localhost TUTORIAL]# ./create_newexpt.csh example

3. Created one folder name 'example' in TUTORIAL folder. Enter example folder.

4. enter example folder by Command prompt.

[root@localhost TUTORIAL]# Cd example

in that folder we get six folders. We delete the three link folders; those are feature_files, s3trainer and util.

4. Create three folders and assign names as feature_files, wav and raw respectively.

5. Copy s3trainer, util, lm3g2dmp, wavtoraw.csh, dictophones.c folders from r TUTORIAL and paste in 'example' folder.

6. Put all wav files in wav folder (Note: those wav files must be saved with extension of .wav)

ex: 1.wav, 2.wav, 3.wav, ...

7. In example folder create example file/**** Procedure Starts****/

In that file write text form of wav file

(ex: A

AA

I)

Note: must care about spaces don't give Space. And at one line only one sentence with corresponding to wav file.

8. In example folder create example.trans file in that file write textform of wav file with .wav file.

(ex: <s> A </s> (1.wav)

<s> AA </s> (2.wav)

<s> I </s> (3.wav))

Note: must care about spaces. Between <s> and A one Space, A </s> one space, and </s> and (1.wav) two spaces. And at one line only one sentence with corresponding to wav file.

9. Open wavtoraw.csh file and change the wavdir and rawdir path

Ex: like as set wavdir = /home/. . . /TUTORIAL/example/wav

Set rawdir = /home/. . . /TUTORIAL/example/raw

10. Then run wavtoraw.csh file by using the Command prompt.

[root@localhost example]# ./wavtoraw.csh

11. Then observed r raw folder. Definitely got .raw files in r raw folder.

12. Create one raw.ctl (control file) by using the below command.

[root@localhost example]# ls /home/eslab/TUTORIAL/example/raw/.raw > raw.ctl*

13. Give r example.Sent file to CMU file by using link

http://www.speech.cs.cmu.edu/tools/lmtool.html

get one .tar file. Extract that file and we copy only '.dic' and '.lm' files and paste in r 'Example' folder.

And give name as example.dic and example.lm respectively.

14. open dictophones.c file in example folder. And Change the path of *argv[] as

*argv[] = { "","/home/eslab/TUTORIAL/example/example.dic", "" };

15. compile dictophones.c by using the command prompt.

[root@localhost example]# cc dictophns.c

16. and getting output by using command prompt.

[root@localhost example]# ./a.out

17. Copy that output from command prompt and save as example.phonelist in r example folder.

18. sort out that exaple.phonelist file using below command.

[root@localhost example]# Sort –u example.phonelist

After sorting add *SIL* at end of example.phonelist file.

19. Then enter in c_scripts by using the command

[root@localhost example]# cd c_scripts/

a) open compute_mfcc.csh file and change outdirctory as

Set outdir = /home/eslab/TUTORIAL/example/feature_files/$x

b) Open variables.def file and change

Set base_dir = /home/eslab/TUTORIAL/example.

20. Run the compuet_mfc.csh using the command as

root@localhost c_scripts]# ./compute_mfcc.csh /home/eslab/TUTORIAL/example/raw.ctl

21. Then we convert mfc.ctl(control) file by using command as

[*root@localhost c_scripts]# cd ..*

[root@localhost c_scripts]# ls /home/eslab/TUTORIAL/example/feature_files/raw/.mfc >mfc.ctl*

22. Then open mfc.ctl file and delete path and extension of .mfc for all files before deleting it as

/home/.../TUTORIAL/raw/1.mfc

after deleting it as *raw/1*

/**** Training part Starts ****/

23. After that once again open the variables.def file in c_scripts change path as

Input to the trainer.

set listoffiles = $base_dir/mfc.ctl

set transcriptfile = $base_dir/example.trans

set dictionary = $base_dir/example.dic

set fillerdict = $base_dir/lists/filler.dict

/**** Procedure Starts****/

set phonefile = $base_dir/example.phonelist

24. After that enter 01.cichmm

Folder in c_scripts folder and put # symbol before unlimit, for

All files in the folders 01.cichmm,

02.cichmm,

03.cichmm,

04.cichmm

And 05.cichmm.

26. Then enter c_scripts by using command as

[root@localhost c_scripts]# cd c_scripts/

for training the data we do as two types. First one as

27. runall .csh files by using below command.

[root@localhost c_scripts]# .runall.csh

and the second one as

[root@localhost c_scripts]# cd 01.cichmm

[root@localhost 01.cichmm]#./slave_convg.csh

if ask y/ n enter y

[root@localhost c_scripts]#cd ../ cd 02.cichmm

[root@localhost 02.cichmm]#./slave_convg.csh

if ask y/ n enter y

[root@localhost c_scripts]#cd ../ cd 03.cichmm

[root@localhost 03.cichmm]#./slave_treebuilder.csh

if ask y/ n enter y

[root@localhost c_scripts]#cd ../ cd 04.cichmm

[root@localhost 04.cichmm]#./slave_tiestate.csh

if ask y/ n enter y

[root@localhost c_scripts]# cd ../cd 05.cichmm

[root@localhost 05.cichmm]#./slave_convg.csh

if ask y/ n y

[root@localhost 05.cichmm]#cd ..

/**** Two ways are completed and complete the training the data is completed ****/

[root@localhost c_scripts]# cd ..

28. At present are in exmaple in command prompt.i.e. [root@localhost example]#

29. convert the .dmp(lagugemodel dump) by using

[root@localhost example]# ./lm3g2dmp /home/eslab/TUTORIAL/example/example.lm

/**** Decoder part Starts ****/

30. Once again enter variables.def file in c_scripts folder and change as

Input to the decoder

set ctlfn = $base_dir/mfc.ctl

set testref = $base_dir/example.sent

set cepdir = $base_dir/feature_files

set lmfile = $base_dir/example.lm.DMP

set dictfn = $base_dir/example.dic

set fdictfn = $base_dir/lists/filler.dict

31. Enter decoding in command prompt using command as

[root@localhost example]# cd decoding/

[root@localhost decoding]# ./compute_acc.ci.csh

32. Run gaumodels file using below command

[root@localhost decoding]# ./launch_decode.ci.1gaumodels

33. compute accuracy using below command

[root@localhost decoding]# ./compute_acc.ci.csh

Then finally get the word accuracy.

/**** Decoder part Ends ****/ /**** Procedure Ends****/

Banjara Language Data

దిత్వార్	భాణజీ	ఘోళియ	వాళోళి	వాందర్	కటుత్రే
సోమవార్	పూపా	సాళ్, ధాన్	అంబాడి భాజి	ఘోడా	ఊంట్
మంగళవార్	పూపీ	ఘౌ, గహూ	నస్సణ్	ఘోడి	జనావర్
బద్వార్	నణద్	ఆటో	ధణియ	గద్దా	హణుత్
వరస్ పత్	నణదోయి	భాజీ	భేజ్రా	గద్దీ	సాళియా
సక్రవార్	దేవర్	పాలోభాజీ	జూర్	ససియా	కిర్ కిండో
ధావార్	జేట్	బోటి	మణగా	వాఘ్	తిత్తక్
దాది	దర్ వాణి	మాళి-మాచళి	చణా	హాతి	క్యాంకోడో
దాదా	బ్వాడి	ఈండా	ఫళి	సిం హ్	కోడియా
నానా	భయ	అచార్	మండవా	రీంచ్	కీడి
నానీ	నాతో	ఖోడి	లూంగ్	హణ్ణి	ఈంచు
ధణి	సం దీ	నూణ్	కాకడి	బళద్	సాప్ సాంప్
గొణ్ణి	సం దణ్	జీరో	లీంబూ	గావుడి	నాగ్ సాప్
ఘరవాళో	వీర	హళద్	నారళ్	సాండ్యా	మొకోడా
ఘరవాళి	జటాణి	గోళ్	సకర్ గంధ్	ఛేళి	సేల్డి
బేటా	నాతి	అంలీ-ఆంబ్లీ	తీల్, తల్లి	బకరా	బిస్సి
బేటీ	మామా	సక్కర్	మక్కా	ఊందర్	ధోళో
భియ్యా	మామి	కాళీ మరచ్	ఆంబా	ఊందరి	కాళో

భాయి	కాక	అద్రక్	తూండి	గోర్లీ	నీళో
భేన్	కాకి	కోతిమిర్	గోలె	గోర్లా	పీళో
బాయి	మొటాప్	పాన్ సపారి	అరెట్ల	ఢోర్	హర్రో

పడ్ దాది	మొటాడి	ఫళ్	అరండి	కాకలా	లాలో,రాత్ డో
పడ్ దాదా	జమాయి	వేంగణ్	కేళా	కుకడి	భూరో
పడ్ నానా	సాళొ	కాంద	కరేల	కుకడో	గోలాల్
పడ్ నానీ	సాళి	ఆలు	సాంటా	సమేళి	కాబ్రో
య,యాడి	ఛోరా	భీండా	అంగూర్	కోబల్	లట్టా
బ, బాపు	ఛోరి	దళియ	ఖర్ బూజా	కోయల్	తాళో
మాసీ	భోజాయి	దాళ్	తర్ బూజా	భేం సి	మాతో
మాసా	భెనోయి	నొంగాణ్	గాజర్	భేం సా	బాపిణి
సాసు	రుకడి	నంగాణ్	జాంబూ	కెల్డా	నాక్
ససరో	దహి	రస్	పపాయ	పాడ్ గా	కాన్
సాడు	జామణ్	తూరీర్ దాల్	సేఫ్	పాడ్ గి	దాంత్
డోక్రి	దూద్	మూంగేర్ దాల్	సపారి	మోర్	తాపాళేపర్
డోక్రా	ఘీ	కోడతి, కోడ్డి	కుత్రా	బతక్	జీబ్
పోతో	నూణి	భాజ్రి	కుత్రి	వాగళ్	హోట్
పోతి	తేల్	వడద్	సూర్, సూరి	కమేడి	ఆంకీ
భాణజో	చావళ్	గోర్ ఫళి	వాందరి	చిబ్రి	జాబ్డి

దాడి	ముక్కి	మూట్	ఢేరొ	లక్కో	పడేచి
గాల్	టాంగ్	సాం ళాన్	వావిడి	జో	పడ్ జాయిస్
కన్ పడి	కేవడి	సాం ళాయె	ఝూడ్	బలా	సురుకర్
ముచ్చె	ఛెణి	మలక్	ఘవేరే	బలావో	ముణాగ్ దేక్
గొద్రి, ఘోగ్	కత్	లాజ్	కడబి	కరొ	టేస్
మూండో	జాత్	కరలా	ఖడ్	రం	ఫరన్ ఆజో
నండి	వియా	కాయి కోని	ఖాన్ సేర్	రమ్మో	బలా
లండి	బేటీర్	వర్లా	సూంతిలి	కాట్	బల్లా
గుదడి	కన్నా	జీవణొ	సూవొ	కాటో	కాన్ యి

కాందో	ఆపణ్	ఎక్కజ్	దాతలి	ఎవడి దేక్	కూణు
కంకో	ఛ	వాటేపర్	తవ్వ	వరా	మార్ మత్
పూటో	హర్	గోళా	కాకొటి	డ్రైం ఆజో	పూచ్
హీక్, ఛాతీ	కతా	వాత్	ద	అల	పూంచా
బచ్చి	లా	క	ఓకరా	ఈ లేల	పూంచ్
పేట్	కరియు	కమత్	తాంగడి	జల్ది జో	పూంచ్ డి
కడ్	ఠీక్ ఛ	కేమేలోచ	తాండో	థం	ఖ్యాంచ్
హాత్	కత్రా	దనియ	దవ్ లత్	కొ	థకేల్
ఖుణి	అఖల్	పెహెల్	దవ్ లత్	హుబర	జూం వుంచు
సూంటి	ఛక	కస్సేకో	ఖాన్ సేర్	భూల్ మత్	జూనావుంచు
అంగళి	ఛేని	నాళి	లోటా	హాల్ మత్	జూచిక
నొక్	మర్గి	నామేతి	తర్వార్	ధాం స్ మత్	జూయెస్నిక
తపాళో	యాండో	భాటా	ఛరి	ఎత్ థం	జోమత్
హతేళి	కోయి	విజళి	కొరాడి	హనువరా	మర్ మత్
పూంద్	కూణ్ కో	నంది	దాతలా	ఇద్ సామళ్	ఖోమత్
జూంగ్	కర్	లకడి	కూద్	దేక్	డర్ మత్
పీండి	కరేచిక	రీస్	కూదో	సస్య	ఖీంచ్
గోడా	కరారోచిక	సేన	టోకోణో	హేట జో	కీంచ్
తళవ	కరా	తోన	బోల్	ఊంపర్ జో	కేవడి, కిమ్మ
ఏడి	మర్గో	హమేన	రొ	ఊతర్	హిమ్మత్
నస్	జకో	సారీ	హాం స్	వాతేకర్	డర్
దం	వరస్	హర్దే	దో	మూండో	ఆజ్
హడ్కా	పణ్	ఖాడ్	ఊట్	బగల్ సరక్	సవార్
అంగూట	కేల్దోంచు	భూల్	ఉటో	హటొ ఆ	కాల్
ఖాల్డీ, ఖాళీ	కేల్దోతో	కాంచళి	ఉటాడో	డీలో చాల్	రాత
లోయి	కేలూంచు	జూంగ్య	మార్	పాచ	దిం యె
కొళజొ	కూణ్	ఫేటువా	మారో	దీటో	మీనా
ఆంతర్	సామళ్	ధోతి	లక్	రోమత్	దాడ్