Text-To-Speech Synthesis System for Marathi Language Using Concatenation Technique

Submitted to
Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, INDIA
For the award of
Doctor of Philosophy

by

Mr. Sangramsing Nathsing Kayte

Supervised by: Dr. Bharti W. Gawali
Professor

Department of Computer Science and Information Technology
DR. BABASAHEB AMBEDKAR MARATHWADA UNIVERSITY,
AURANGABAD

November 2018

Abstract

A speech synthesis system is a computer-based system that should be able to read any text aloud with a particular language or multiple languages. This is also called as Text-to-Speech synthesis or in short TTS. Communication plays an important role in everyones life. Usually communication refers to speaking or writing or sending a message to another person. Speech is one of the most important ways for human communication. There have been a great number of efforts to incorporate speech for communication between humans and computers. Demand for technologies in speech processing, such as speech recognition, dialogue processing, natural language processing, and speech synthesis are increasing. These technologies are useful for human-to-human like spoken language translation systems and human-to-machine communication like control for handicapped persons etc., TTS is one of the key technologies in speech processing.

TTS system converts ordinary orthographic text into acoustic signal, which is indistinguishable from human speech. For developing a natural human machine interface, the TTS system can be used as a way to communicate back, like a human voice by a computer. The TTS can be a voice for those people who cannot speak. People wishing to learn a new language can use the TTS system to learn the pronunciation. The TTS system can be used to read text from emails, SMSs, web pages, news, articles and so on. In such reading applications, the TTS technology can reduce the eye strain. Speech synthesis systems can be extremely useful to people who are visually challenged, visually impaired and illiterate to get into the mainstream society. The TTS systems can also be integrated to work with systems that recognize text from scanned documents. More recent applications include spoken dialogue systems and communicative robots.

A TTS synthesis system is the artificial production of human speech. Text and voice models are given as input to this system, which in turn generates speech as output corresponding to the given voice models. The input text can be in various forms such as keyboard input, text or word documents, emails, web pages, blogs, Short Message Service (SMS) etc. The input speech is converted to parametric representation by analysis and

later the parametric speech is converted back to original speech. There will be small difference between the original speech and speech produced by resynthesize. However, TTS is more challenging as compared to analysis-by-synthesize because; we have to model the prosody from the text in TTS.

There are mainly used speech synthesis techniques are articulatory, formant, Concatenative methods. Out of all, Unit selection synthesis which comes under the family of concatenative synthesis, helps to synthesis the fluent speech for the available recorded databases. The database is designed to the utterance of the phonemes and di-phones. The implementation of the proposed system is done in Linux based operating system. The setup is prepared with the various commands for the building the Text-To-Speech system using Festvox and Speech tools. The MOS analysis results the speech produced using Unit based TTS system. Then the implementation of the proposed work is done in two ways: by MATLAB Marathi talking calculator and by Building APP for Marathi talking calculator. The Matlab is used for initially developing the console for Speech synthesis. This work is implemented using the designing of calculator. For this particular application, the total number of words with probability are 121, utterance and the data was collected in 1 sessions so, the overall 121/- vocabulary size are collected for the database. The speech is recorded using CSL machine in sound and noise free environment. The Marathi numerals and the specific operation recordings are interfaced. The quality of speech is evaluated performing the MOS analysis. The calculations shows that the speech produced using Matlab application gives the 85% of understandable and quality speech. In further implementation, the Android based APP is developed for Marathi calculator. The Special features of developed android APP are : Speaks out each number, Speaks out the operations, Speaks out the result with proper digit places in Marathi and the voice produced is clear and correct. The analysis of the synthesized speech shows that the 95.5% of individuals rated the the quality of synthetic speech is good and understandable. The only 4.5 % speech is perceptible and not clearly produced. Thus Unit selection method provides the naturalness and understandability, the two important parameters of TTS system.

Approval

This is to certify that the present work in the form of thesis entitled **Text-To-Speech Synthesis System for Marathi Language Using Concatenation Technique** is an original work carried out by **Mr. Sangramsing Nathusing Kayte.** The work included in this thesis is original, unless stated otherwise and has not been submitted for the other degree of Dr. Babasaheb Ambedkar Marathwada University or any other University. References made to the work of others have been cited in the text.

Head

Dr. B.W. Gawali
Professor & Head
Department of Computer Science & I.T.
Dr. Babasaheb Ambedkar Marathwada University,
Aurangabad

Date: 12-10-2017

Declaration

I hereby declare that the present work in the form of thesis entitled **Text-To-Speech Synthesis System for Marathi Language Using Concatenation Technique** is an original work carried out by me at the Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, under the guidance of **Prof. Bharti W. Gawali** for the Ph.D. in Computer Science and not previously submitted to this or any other University for the fulfillment of any other degree or any other similar to this work.

Mr.Sangramsing Nathusing Kayte
Date: 12-10-2017

Acknowledgments

At the moment of accomplishment of the thesis, I would like to thank all the hands who were directly and indirectly involved in supporting me during my research years. I would like to express my sincere gratitude to all of them. First of all, I am extremely grateful to my research guide, mentor and supervisor, **Prof. Bharti W. Gawali,** Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, for her valuable guidance, scholarly inputs and consistent encouragement which I received throughout my research work. This feat was possible only because of the unconditional support provided by mam. A person with an amicable and positive disposition, mam has always made herself available to clarify my doubts despite her busy schedules, because of her motivation and kindheartedness I was able to complete my doctoral degree, and I consider it as a great opportunity to do my doctoral programmer under her guidance and to learn from her research expertise. Thank you so much mam, for always been with me, all your help and support in my academic and even my personal life matter a lot to me. Her mentorship is a paramount in providing a well-rounded experience consistent in my long-term career goals. A huge thanks mam from the bottom of my heart.

I would like to thank the Department of Computer Science and Information Technology and especially the System Communication and Machine Learning Research Laboratory for providing me equipment and required support during my research work.

Immortal thanks to my parents **Mr. Nathusing Kayte** and **Mrs Naganebai Kayte,** without their support, backing, assistance my work would not have been completed. Special thanks to my brother **Dr. Charansing Kayte** and **Jaypalsing Kayte** for their support and help whenever required.

An immeasurable thanks to my best-friend **"MLA"** who has sacrificed a lot and always tried to help me. I was able to do my work with the constant support and encouragement. And at last but not the least I bow my head and thank to my dear God for giving me everything I deserve. Very big thanks to all again...

Table of Contents

List of Figures

List of Tables

1 Introduction

Speech is the vocalized form of human communication. It is based upon 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. These vocabularies, the syntax which structures them, and their set of speech sound units differ, creating the existence of many thousands of different types of mutually unintelligible human languages [1]. Humans interact with information processing systems in many ways; and the interface between humans and the computers they use is crucial to facilitate this interaction. Voice user Interfaces (VI) are utilized for speech recognition and synthesizing systems. Voice communication is the enunciated form of human linguistic communication and is the key means of communication between people. It is as well the most natural and effective type of exchanging information among human. Speech processing is the study of speech signals and the processing methods of these signals. It comes as a front end to a growing number of language processing applications. It includes various areas of study such as It includes various areas of study such as:

- Speaker recognition
- Speech recognition
- Speech coding
- Speech enhancement
- Speech compression
- Speech synthesis.

Speaker recognition: Speaker recognition is the identification of a person from characteristics of voices (voice biometrics). It concentrates on who is speaking.

Speech recognition: Speech recognition is the ability of a machine or program to identify words, phrases and sentences in spoken language. Thus it deals with what is being spoken.

Speech coding: Speech coding is the art of creating a minimally redundant representation of the speech signal that can be efficiently transmitted or stored in digital media, and decoding the signal with the best possible perceptual quality.

Speech enhancement: The objective of enhancement is improvement in intelligibility and/or overall perceptual quality of degraded speech signal using audio signal processing techniques.

Speech compression: Encoding digital speech to take up less storage space and transmission bandwidth. The PCM, ADPCM, CELP and LD-CELP methods are commonly used for speech compression.

Speech Synthesis: Speech synthesis is a procedure of automatic generation of spoken language by computer. It is also referred as text-to-speech system. In this process a written text is changed into voice communication. The goal of speech synthesis is to produce a machine having an understanding and natural sounding voice for communication.

1.1 Speech Synthesis

Speech synthesis systems first convert the input text into its corresponding linguistic or phonetic representations and then produce the sounds corresponding to those representations. With the input being a plain text, the generated phonetic representations need to be augmented with information about the intonation and rhythm that the synthesized speech should have. This task is done by a Natural Language Processing (text analysis) module in most speech synthesizers. The phonetic transcription and prosodic information obtained from the text analysis module, is then given to a Digital Signal Processing (DSP) module that produces synthetic speech. The figure 1.1 below describes the architecture of Text-to-Speech synthesis system

1.1.1 Natural Language Processing

Natural language processing module involves in the conversion of the given text into its corresponding phonetic information and predicts prosodic information. In this process

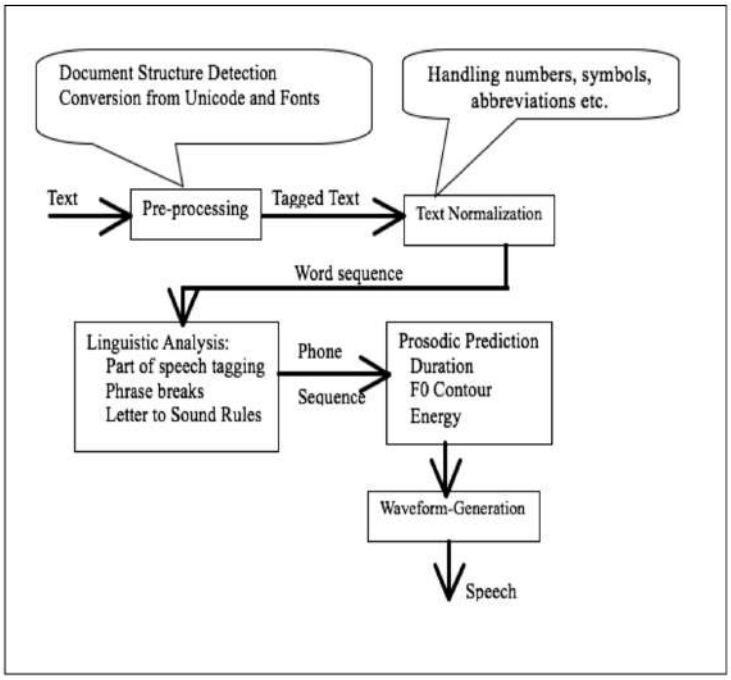


Figure 1.1: Architecture of Text-to-Speech synthesis system

it will depends on the language. In some languages all the possible linguistic units are have their own speech sound [2]. However, in languages like English some linguistic units may have different pronunciations depending on the context in that sentence, this is called homograph disambiguation. Hence these languages require large number of pronunciation rules to pronounce the words exactly matches with the context. But in case of Marathi language, linguistic units have fixed pronunciations for each unit that may ease the speech synthesis process. All these issues are resolved in natural language processing modules with sub blocks like

- **Text analysis**: A systematic analysis of the content rather than the structure of a communication, such as a written work, speech, or film, including the study of thematic and symbolic elements to determine the objective or meaning of the communication.

- **Phonetic analysis:** The science or study of speech sounds and their production, transmission, and perception, and their analysis, classification, and transcription, the science or study of speech sounds with respect to their role in distinguishing meanings among words.

- **Prosodic analysis:** analysis of a language based on its patterns of stress and intonation in different contexts. In systemic grammar, prosodic analysis is regarded as an essential foundation for the analysis of syntax and meaning.

1.1.2 Digital Signal Processing

The operations involved in the Digital Signal Processing module are based on the parameters of controlling the articulatory muscles and the vibratory frequency of the vocal folds, so that the output signal matches the input requirements. In order to do it properly, the DSP module should obviously, in some way, take articulatory constraints into account, since it has been known for a long time that phonetic transitions are more important than stable states for the understanding of speech. This, in turn, can be basically achieved in two ways:

- **Synthesis-by-rule:** How long will it be before your computer gazes deep in your eyes and, with all the electronic sincerity it can muster, mutters those three little words that mean so much: "I love you"! In theory, it could happen right this minute: virtually every modern Windows PC has a speech synthesizer (a computerized voice that turns written text into speech) built in, mostly to help people with visual disabilities who can't read tiny text printed on a screen.

- **Synthesis-by-concatenation**: TTS systems use some type of synthesis-by- concatenation method. Concatenation techniques take small units of speech, either waveform data or acoustically parameterized data, and concatenate sequences of these small units together to produce either time varying acoustic parameters or, alternatively, waveforms.

1.2 Speech Synthesis Techniques

The Text-to-Speech (TTS) synthesis is a process to modify the input text into speech. The TTS is developed for the system with particular knowledge of a language. The accuracy measurement of speech synthesis system depends on the quality of the speech recorded in speech database. It also makes use of linguistic analysis for correct pronunciation, prosody (pitch, duration etc.,) and acoustic representations of speech to generate waveforms.

The working of TTS system includes two main components: the front-end and the back-end. The front-end converts the text into required text format. Here the various symbols like date, phone number are even converted into text. So here every phoneme is converted back into required text format. The back-end is the part of the system that is closer to the speech output which converts the output of the front-end (phonetic transcriptions and prosodic information) to the corresponding waveform. Here also the prosody and voice characteristics of each phoneme is considered before producing the output. There are different kinds of synthesis methods that can be used when building a TTS synthesis system. Some of these methods require a set of rules to drive the synthesizer whereas others depend on parameters exercised from the recorded speech database. These classifications are called rule driven and data driven or corpus based synthesis respectively. Examples of rule driven synthesis include articulatory synthesis and formant synthesis. On the other hand, examples of data driven synthesis include concatenative synthesis and HMM based synthesis [1].

Producing synthetic speech segments from natural language text utterances comes with a unique set of challenges and is currently under serviced due to the unavailability of a generic model for all available languages. The classification of different standard speech synthesis techniques are presented in Figure 1.2 This chapter also discusses about the current status of the text to speech technology in Indian languages focusing on the issues to be resolved in proposing a generic model for different Indian languages.

*1.2.1 **Articulatory Synthesis***

Articulatory synthesis is a technique for synthesizing speech based on human speech production model directly. It is designed by how the human articulators such as vocal tract, nasal tract, lungs and larynx generate speech. The output synthetic speech produced by this model is most natural but it is the most challenging method and computationally very expensive [3]. The working process of this synthesizer completed depends on the articulatory muscles and organs. The study shows that the vocal tract tube is divided into many small sections, and each section carries information with the help of nervous involved in the speech production system. The figure 1.3 represents the human speech production organs [4]. Typical Articulatory synthesis system uses seven to eleven parameters to adequately describe motion of various articulators. The use of eleven features for this synthesizer are as follows: one parameter for controlling velum opening, one for lip rounding, one for lip closure, two each for the tongue body and tongue tip as they have both vertical and horizontal degrees of freedom, one each for jaw height, pharynx width, and larynx heigh [5]. The various are produced depending on the interaction with vocal chords and

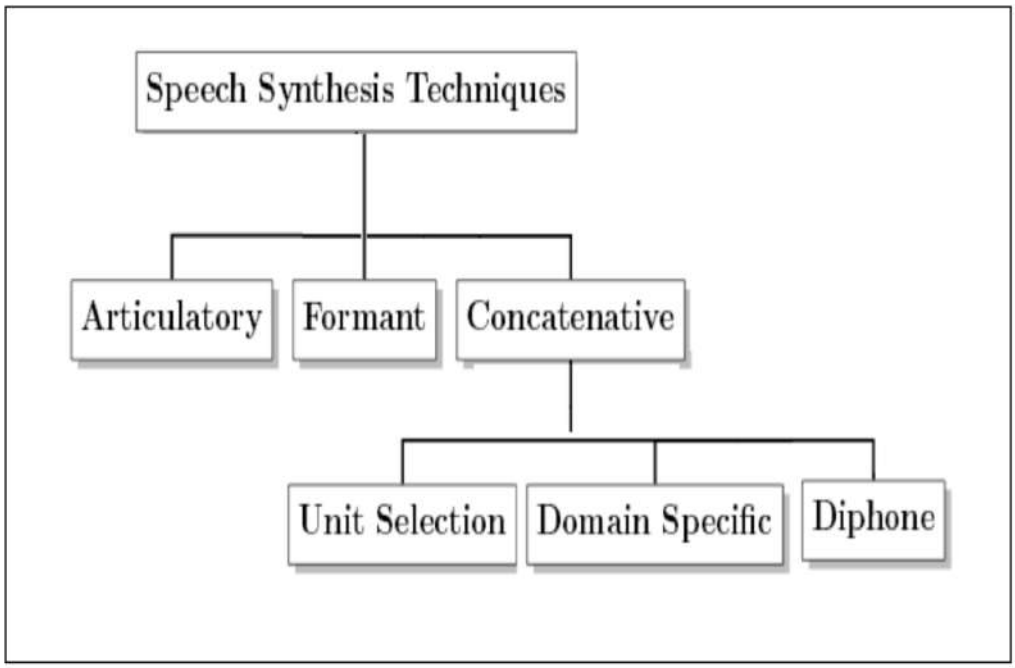


Figure 1.2: Classification of speech synthesis techniques

articulatory system. Like for example, in producing pulmonic sounds, the breathing muscles act as an energy source and the lungs provide storage of pressurized air. The vocal fold or vocal chords separates the lungs from the vocal tract. The signals generated by the vocal folds are filtered by the vocal tract and are then radiated to the surroundings via the mouth and/or nostrils as speech signals [6].

Articulatory synthesis models this natural speech production process as accurately as possible by creating a synthetic model of human physiology and making it speak [7]. The rule-based synthesis allows the articulatory control parameters may be the lip aperture, lip protrusion, tongue tip height, tongue tip position, tongue height, tongue position, velic aperture, etc [8]. and the excitation parameters may be the glottal aperture, cord tension, and lung pressure. While speaking, the vocal tract muscles cause articulators to move and change shape of the vocal tract which causes different sounds [9].

The study of articulatory speech synthesis helps to understand the speech production process at the deepest levels for construction of speech synthesizer indistinguishable from a natural speaker. Articulatory models are also used for speech recognition for understanding the process of speech production and recognition by modeling it as a whole [10]. The applications that make widely use of this technique are virtual language tutor, in speech therapy, in general purpose audio-visual speech synthesis [11], in speech encod-

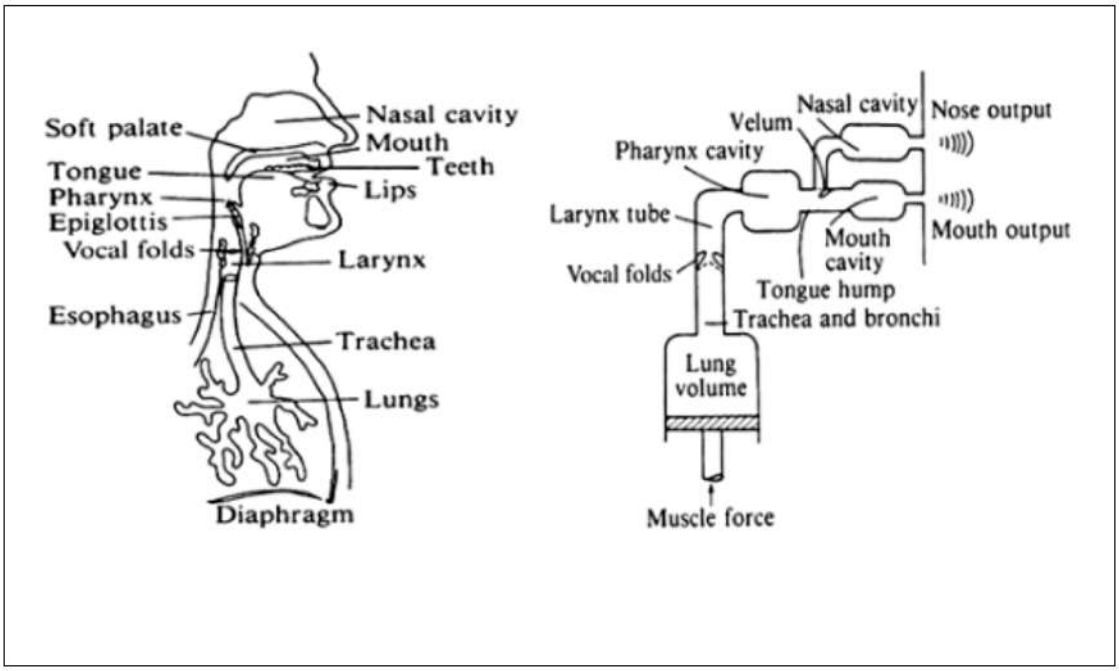


Figure 1.3: The human speech production organs (left) and an idealized model of speech production (right)

ing, in imitation of real speakers, as the speech engine of virtual actors and even as a toy [4].

*1.2.2 **Formant Synthesis***

This is the oldest method for speech synthesis, and it dominated the synthesis implementations for a long time. The working of formant synthesis is based on the well-known source filter model which indicates that the idea is to create periodic and non-periodic source signals and to feed them through a resonator circuit or to develop a model that resembles the vocal tract. The figure 1.4 represents the architecture of formant synthesizer. The principles are thus very simple, which makes formant synthesis flexible and relatively easy to implement. Formant synthesis have the capacity to produce the any type of sounds [12].

On the other hand, the simplifications made in the modeling of the source signal and vocal tract inevitably lead to somewhat unnatural sounding result [13]. The synthesized speech is also generated using an additive synthesis and an acoustic model. The acoustic model uses parameters like, voicing, fundamental frequency, noise levels, etc that varied

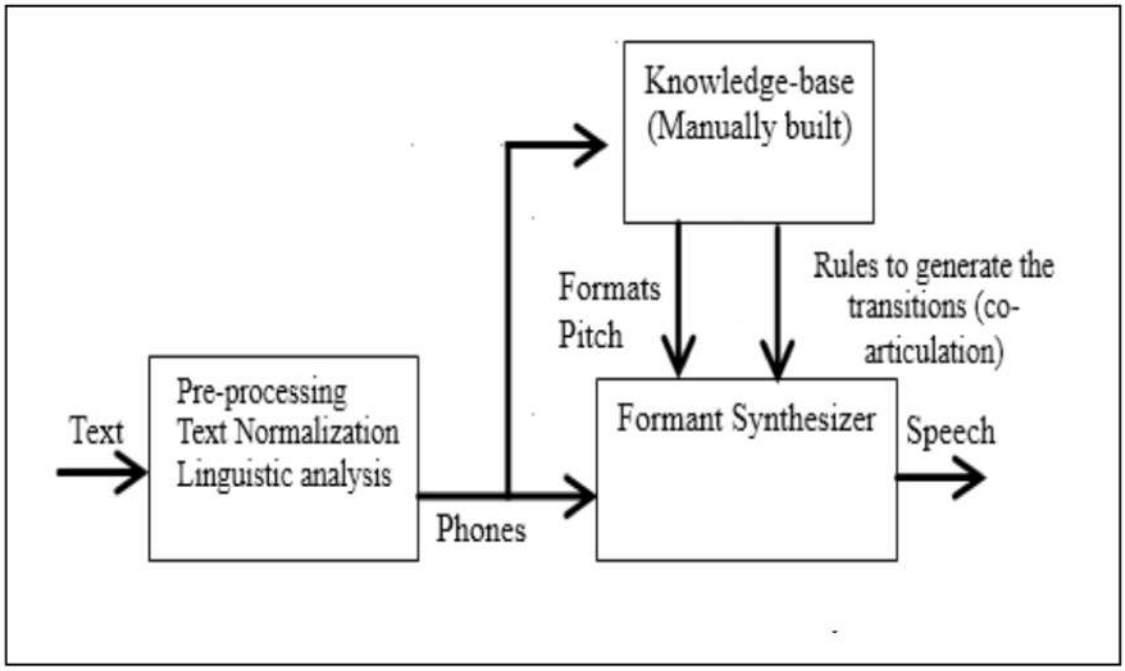


Figure 1.4: A Typical Architecture of Formant Synthesizer

over time [3]. The technique produces highly intelligible synthesized speech, even at high speeds, avoiding the acoustic glitches commonly plagued in concatenative synthesis systems (discussed next). These are usually smaller programs compared to the concatenative techniques as they do not depend on a speech corpus to produce the output speech. Therefore, formant synthesis is widely used speech synthesis technique for embedded systems, where memory and microprocessor power are limited. However, the major drawback of the technique is, the system generates artificial, robotic-sounding speech that is far from the natural speech spoken by a human. Also, it is relatively difficult to design rules that specify the timing of the source and the dynamic values of all filter parameters for even simple words [14]. The overview of the rule based formant synthesis technique is presented in Figure 1.5. With the indent of formant-based systems have complete control on all aspects of the output speech.

It helps to produce the ariety of emotions and different tone voices with the use of prosodic and intonations modeling techniques. The formant synthesis technique are widely being used for utterance copy i.e. for mimicking the voice features that takes speech as input and find the respective input parameters that produces speech, mimicking the target speech . Mimicking the voice characteristic is relatively a difficult and ongoing area of research these days. However, a lot of further research may be undertaken to

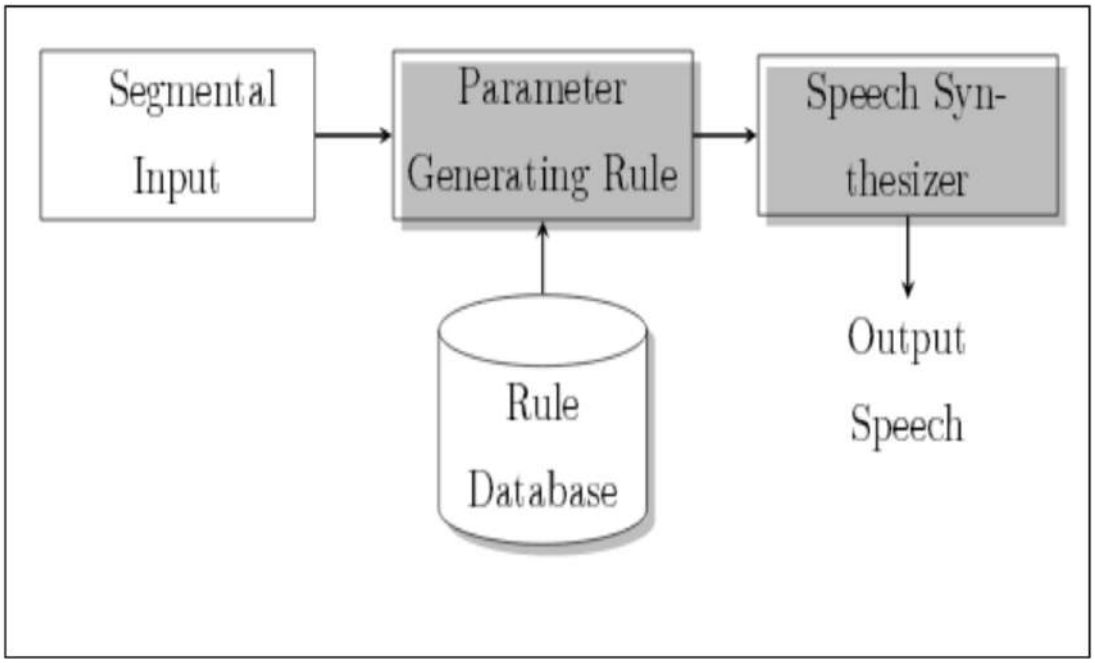


Figure 1.5: Rule-based synthesis approach

obtain more natural sounding speech segments by optimizing different speech parameters.

Formant synthesis glues together the physical and spectral modeling approaches. It is a physical model in that there is an explicit division between glottal-flow wave generation and the formant-resonance filter, despite the fact that a physical model is rarely used for either the glottal waveform or the formant resonator. On the other hand, it is a spectral modeling method in that its parameters are estimated by explicitly matching short-time audio spectra of desired sounds [13].

1.2.3 ***Concatenative Synthesis***

Concatenative synthesis is a technique for synthesizing sounds by concatenating short samples of recorded sound present in speech database (called units). The duration of these speech units is not strictly defined and may vary according to the implementation, roughly in the range of 10 milliseconds up to 1 second. It is used in speech synthesis and music sound synthesis to generate user-specified sequences of sound from a database built from recordings of other sequences.

Concatenative Synthesis is based on the joining of segments of recorded speech. In this method waveform segments are stored in a database. For a given text, these segments

are joined based on some joining rules. Connecting prerecorded natural utterances is probably the easiest way to produce intelligible and natural sounding synthetic speech. One of the most important aspects in concatenative synthesis is to find correct unit length. The selection is usually a trade-off between longer and shorter speech units. With longer units high naturalness, less concatenation points and good control of co articulation are achieved, but the amount of required units and memory is increased. The figure 1.7 shows the architecture of concatenative synthesis. With shorter units, less memory is needed, but the sample collecting and labeling procedures become more difficult and complex [15]. There are three main sub- type of concatenative synthesis.

The Concatenative speech synthesis technique is a corpus-based technique that makes use of pre-recorded speech samples (words, syllables, half-syllables, phonemes, di-phones or tri-phones) in a database and produces the output speech by concatenting or joining the appropriate units based on the entered text utterances [16]. The simplicity of the model and highly natural speech production quality makes it suitable for its use in designing human computer interactive systems for different domains [17]. The overview of a concatenative speech synthesis system based on unit selection technique is presented in Figure 1.6 and is discussed next. The quality of the synthesized speech is affected by the unit length in the database [18]. The naturalness of the synthesized speech increases with longer units while by using longer units less concatenation points are there reducing the formation of unnatural segments at concatenation points.

However the disadvantage of this system is more memory is needed and the number of units stored in the database becomes very numerous. On the other hand with shorter units, the memory requirement is less but the complexity of sample collection and labeling techniques increases. The concatenative technique may broadly be classified into the following three type basis on the unit type stored in its database.

1.2.3.1 ***Unit selection synthesis***

Unit selection synthesis is the so called cut and paste synthesis in which short segments of speech are selected from a prerecorded database and concatenated one after another to produce the desired utterances. The short segments of speech varies from phones to phrases. The longer the selected units are, the fewer problematic concatenation points will occur in the synthetic speech. In unit selection synthesis large databases of recorded speech are used. The primary motivation for the use of large databases is that with a large number of units available with varied prosodic and spectral characteristic it should be possible to synthesize more natural-sounding speech than that can be produced with

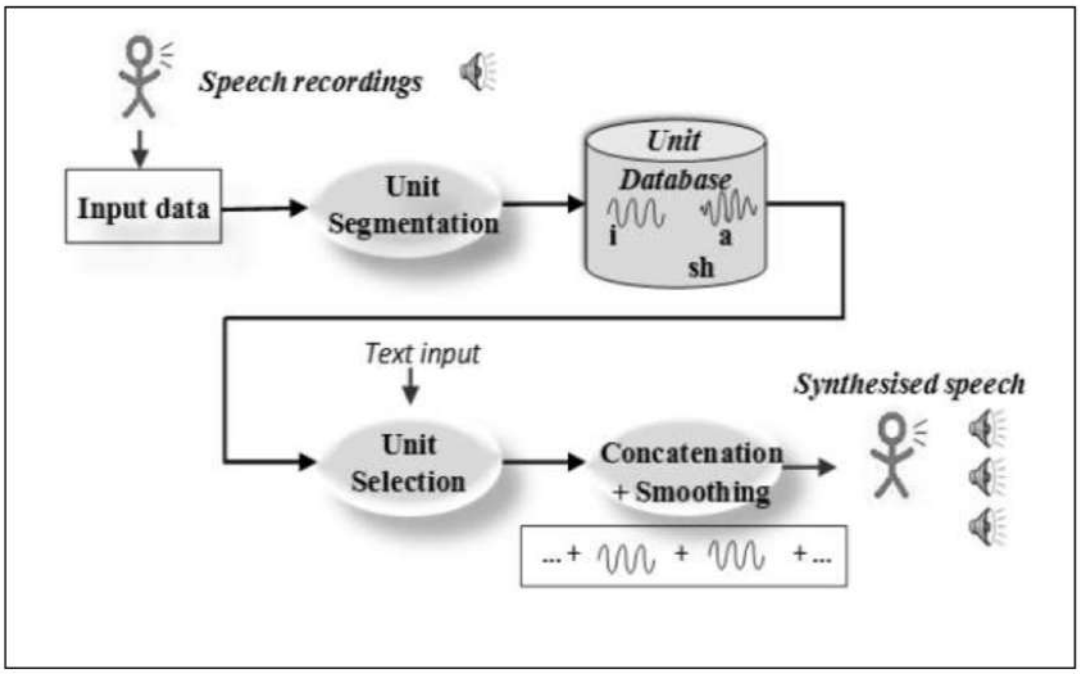


Figure 1.6: Unit selection approach in concatenative synthesis

a small set of controlled units [19]. During database creation, each recorded utterance is segmented into some or all of the following: individual phones, diphones, half-phones, syllables, morphemes, word, phrases, and sentences [20].The figure 1.7 represents the typical architecture of Unit- Selection synthesis. Unit selection synthesis requires a large databases of recorded speech. During database creation, each recorded utterance is segmented into individual phones, syllables, morphemes, words, phrases, and sentences. Typically, the division into segments is done using a specially modified speech recognizer set to a forced alignment mode with some manual correction afterward, using visual representations such as the waveform and spectrogram. An index of the units in the speech database is then created based on the segmentation and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and neighboring phones [21]. At runtime, the desired target utterance is created by determining the best chain of candidate units from the database. This process is typically achieved using a specially weighted decision tree [22]. In the unit selection scheme, by using the target cost and the concatenation cost, speech units are selected from the whole speech database, and concatenated in run-time. In this scheme, a heuristic distance is defined between contexts to measure the target cost. To avoid this, a clustering-based scheme may be used which clusters the contexts in advance, and selects each unit from a cluster. A typical decision

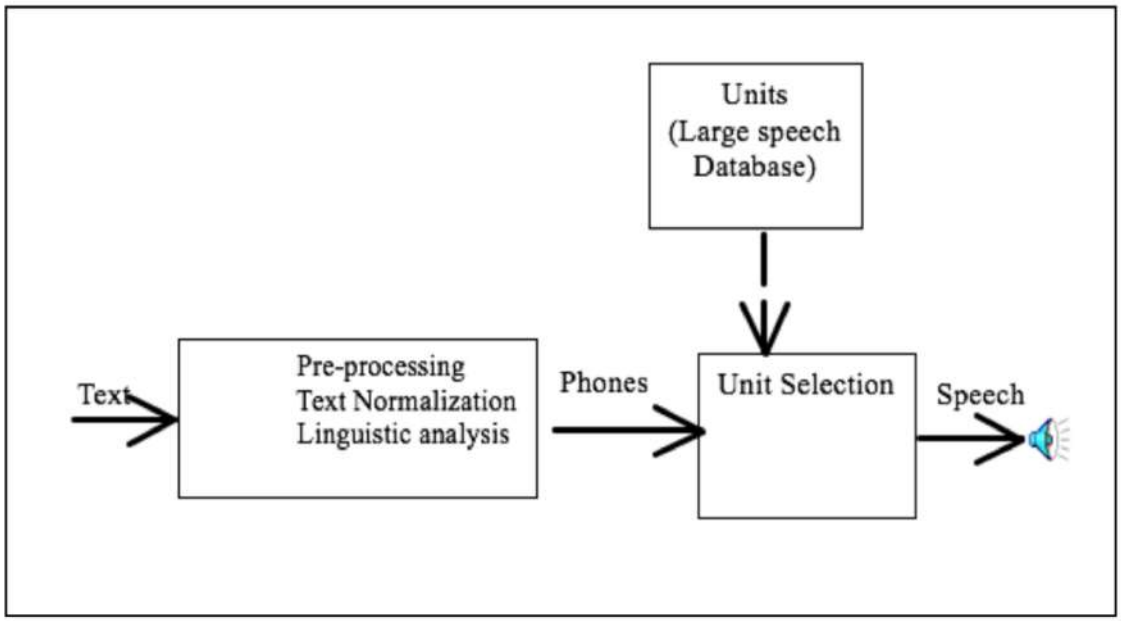


Figure 1.7: A Typical Architecture of Unit-Selection Synthesis

tree based on concatenation cost for unit selection and is presented in Figure 1.8 along with an overview of the clustering based unit selection scheme. As only a small amount of digital signal processing is applied to the recorded speech, unit selection technique produces highly natural speech segments. However, maximum naturalness typically requires unit-selection speech databases to be very large.

Also, unit selection algorithms have been known to select segments from a place that results in less than ideal synthesis. For example minor words become unclear, even when a better choice exists in the database. Recently, researchers have proposed various automated methods to detect unnatural segments in unit-selection speech synthesis systems [22] however, a lot of work may further be done to achieve better performance.

1.2.3.2 ***Di-phone synthesis***

Di-phone synthesis uses a minimal speech database containing all that occurring in a language. In di-phone synthesis, only one example of each di-phone is contained in the speech database. The quality of the resulting speech is generally worse than that of unit-selection systems, but when compared to format synthesis it is more natural-one [23]. Compared to unit selection synthesis technique, di-phone synthesis uses a minimal speech database containing all the di-phones occurring in a given language. The figure 1.9 de-

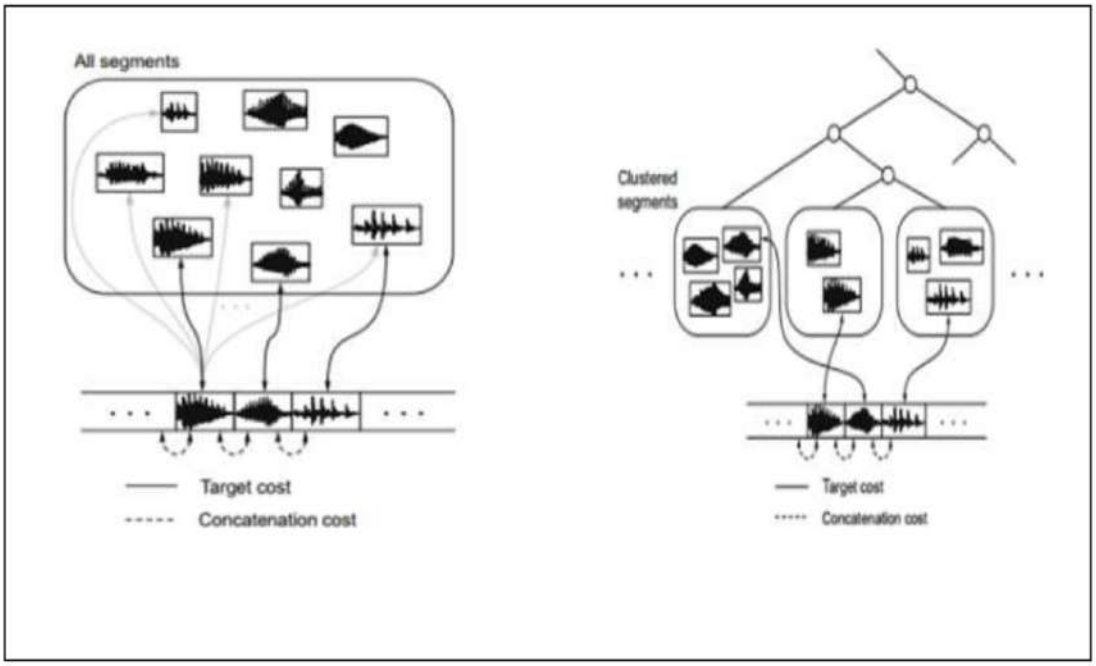


Figure 1.8: Overview of general unit-selection scheme (left) and clustering-based unit Selection scheme (right)

scribes the typical architecture of Di-phone synthesis process. In diphone synthesis, with only one instance of the speech unit being available, extensive prosodic modifications have to be applied to obtain good quality speech. Diphone synthesis have a drawback of robotic-sounding synthesized speech. Although due to a number of freely available software implementations, it continues to be used in research but its use in commercial applications is declining [24]. TTS systems based on diphone synthesis need prosodic models to produce good speech output. The prosodic analysis for these models require a database of speech annotated with linguistic and prosodic labels.

Tools are also required to generate appropriate linguistic information essential to predict prosody from text [25]

1.2.3.3 ***Domain-specific synthesis***

Domain-specific synthesis combines the pre-recorded words and phrases to create complete utterance. It is widely applied in areas like transit schedule announcements or weather reports, as the output is limited to particular domain and also the text to be produced is also limited one. Because the variety of sentence types is limited, and they

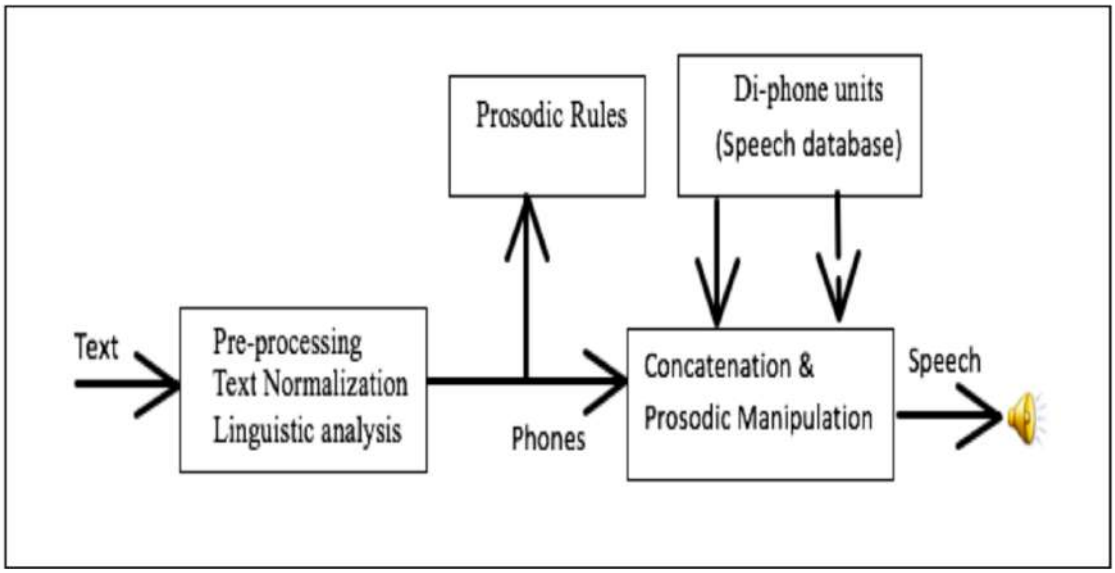


Figure 1.9: A Typical Architecture of Di-phone Synthesis

closely match the prosody and intonation of the original recordings, the level of naturalness is excellent in the speech produced from this method.

Table 1 .1 presents a comparative study of the discussed speech synthesis techniques with respect to their advantages and disadvantages. Different studies have investigated speech synthesis in different cultures, most of which have focused on achieving natural sounding speech in the respective languages. There are fewer models available in different Indian languages for text to speech. Synthesis studies demonstrated that, although speech synthesis in different languages may have different issues due to the pronunciation variations, the quality of the synthesized speech is regulated by intelligibility and naturalness of the produced speech. While intelligibility refers to the understandability of the artificially produced speech, naturalness refers to how closely it seems like a human generated speech.

Individuals are able to produce intelligible speech in different regional languages [26]. In Xias study [12], the unit selection scheme achieves better results for languages like, English, French, Japanese, etc. where the major objective is to achieve highly natural speech. However, when the judged expressions were short the unit selection scheme provides highly natural speech compared to other approaches. At the same time by increasing the length of the text, the processing overhead increases. A discussion in [27] is presented for the size of the speech database in Indian languages. The researchers in [28] presented the efforts to build a high quality syllable-based framework for unit selection for 13 Indian

Table 1.1: Comparisons of speech synthesis techniques

Technique	Method Used	Advantages	Disadvantages
Articulatory	Mathematical model of human speech production	Needs no speech database	Robotic sounding Speech
Formant	Rule-based	Needs no speech database	Robotic sounding Speech
Unit Selection	Concatenative	highly natural speech	Database dependency
Unit Selection	Concatenative	highly natural speech	Database dependency
Domain Specific	Concatenative	highly natural speech	Speech for limited word may be produced

languages. However, the appropriate size of the speech database needed for unit selection synthesis in all official Indian languages is still an open question. Kishores study in [29] discusses about the appropriate unit size for unit selection speech synthesis. A discussion in [30, 31] is presented on the associated challenges for designing corpus based speech synthesis systems. Murty in [32] discusses about the initiative in building unit selection speech synthesis system in Indian languages. However, the larger size of the unit selection speech database increases the difficulty for its use in small hand held evices having limited storage resources. The Festival speech synthesis system [33]. Provides an open architecture for multi lingual speech synthesis research, where some Indian languages like, Hindi, Marathi, Tamil and Telugu are included along with their western languages (English). A comparison of the existing speech synthesis techniques with respect to the considered parameters are presented in Table 1.2.

Table 1.2: Comparison of speech synthesis techniques with respect to considered parameters

Technique	Corpus Size	Complexity	Intelligibility	Naturalness
Articulatory	Corpus independent	Very high	High	Low
Formant	Corpus independent	High	High	Low
Concatenative	Large	Low	High	High
Syllable-based	Large	Low	High	High

1.3 Importance of speech synthesis

As a result, advanced developments in speech synthesis and natural language processing techniques and mostly due to the emergence of new technologies in Digital Signal

Processing to do the efficient computations, the concept of high quality speech synthesis systems appeared in the mid-eighties. It is now a must for the family of speech products, because they can reduce human strength while executing the tasks related to speech and their enormous applications [34] like

- **Telecommunications services:** In telecommunication, a telecommunications service is a service provided by a telecommunications provider, or a specified set of user-information transfer capabilities provided to a group of users by a telecommunications system.

- **Language education:** Language education refers to the process and practice of acquiring a second or foreign language. It primarily is a branch of applied linguistics, however can be considered an interdisciplinary field.

- **Aid to handicapped persons:** disability is an impairment that may be cognitive, developmental, intellectual, mental, physical, sensory, or some combination of these. It substantially affects a person's life activities and may be present from birth or occur during a person's lifetime.

- **Talking books and toys:** An audiobook (or talking book) is a recording of a text being read. A reading of the complete text is noted as "unabridged", while readings of a reduced version, or abridgement of the text are labeled as "abridged".

- **Vocal Monitoring:** An audio-vocal monitoring system plays an important role in speech production, based on auditory feedback about the speakers own voice.

- **Multimedia, man-machine communication:** Multimedia is content that uses a combination of different content forms such as text, audio, images, animations, video and interactive content. Multimedia contrasts with media that use only rudimentary computer displays such as text-only or traditional forms of printed or hand-produced material.

1.4 Objective of the work

The main aim of the research is to develop a speech synthesis system using festival and speech tools for Marathi speech database with Indian dialect. Most of the research work is concentrated on configuring the computer as compatible to run the speech synthesis

system and recording speech corpus without noise to generate voice models, to achieve good accuracy, natural speech as output for the given input text. A speech synthesis system is used to generate synthetic speech using voice models, in this project; Festival, MATLAB, Android, a speech synthesis framework is used to synthesize the speech for the given input text with the generated voice models. Speech tools are heart of the festival framework which is used to process the speech signals. Festival, Festvox and MATLAB, Android tools are together used to generate the new unit selection and statistical parametric synthetic voice models using particular persons speech data base. Festival and speech tools are the open source packages which are available in Edinburgh University, centre for speech technology research (CSTR) website.

- Thus the proposed research aims at development of an Marathi Text-to- speech synthesis
- Analyse the performance of festival and Festvox for Marathi Language.
- To develop Text to Speech Synthesis application in Marathi Language.
  - Marathi Talking Calculator using MATLAB for Computer System
  - Building Android application for Marathi Talking Calculator

1.5 Marathi Language

Classification: Indo-European, Indo-Iranian, Modern Indo-Aryan, South-western. Other languages of the south-western group are Gujarati and Konkani. Overview. Marathi is a regional language of central-western India spoken in the state of Maharashtra. It descends from the Middle Indo-Aryan Maharashtra, the most literary of the vernacular languages known as Prakrits. It has been influenced by Kannada and Telugu, two neighboring Dravidian languages.

Distribution and Speakers. Marathi is spoken in India, in the state of Maharashtra and adjacent regions (Gujarat, Madhya Pradesh, Chhattisgarh, Andhra Pradesh, Karnataka, and Goa) by around 83.5 million people: Table 1.3 shows the population speaking Marathi language in each state in India.

Status. Marathi is the official language of the central Indian state of Maharashtra.

Varieties. Deshi spoken in the city of Pune, Varhaddi and Nagpuri spoken in the city of Nagpur and surrounding area, and Kokni prevalent in the coastal regions.

Table 1.3: Distribution and Speakers

Maharashtra	77,300,000
Karnataka	2,194,000
Madhya Pradesh	1,470,000
Gujarat	885,000
Andhra Pradesh	706,000
Goa	352,000
Chhattisgarh	163,000
Other states	400,000

1.5.1 Features of Marathi Language

Indian languages have a more sophisticated notion of a character unit or Akshara that forms the fundamental linguistic unit. An Akshara consists of 0, 1, 2, or 3 consonants and a vowel. Words are made up of one or more Aksharas. Each Akshara can be pronounced independently as the languages are completely phonetic. Aksharas with more than one consonants are called samyuktaksharas or combo-characters. The last of the consonants is the main one in a samyuktakshara. All Indian languages have essentially the same alphabet derived from the Sanskrit alphabet. This common alphabet contains 33 consonants and 15 vowels in common practice. Additional 3-4 consonants and 2-3 vowels are used in specific languages or in the classical forms of others. This difference is not very significant in practice. Individual consonants and vowels form the basic letters of the alphabet [35].

Marathi is an Indo-Aryan language spoken by the Marathi people of western India (Maharashtrians). It serves as the official language of the state of Maharashtra, with roughly ninety million fluent speakers worldwide. Marathi ranks 4th in India with respect to the number of people who claim it as their primary language. Along with Bengali, Marathi is the oldest of the regional literatures in Indo- Aryan languages, dating from about AD 1000 [36]. Marathi is at least fifteen hundred years old, and derives its grammar and syntax from Pali and Prakrit. The Marathi language was earlier known as Maharashtra, Maharathi, Malhatee or Marathi in ancient times. Marathi is usually written in the Devanagari script, a character-set consisting of 36 consonant letters and 16 initial-vowel letters. It is written from left to right. The Devanagari alphabet used to write Marathi is slightly different from the Devanagari alphabets of Hindi and other languages: there are a couple of additional letters in the Marathi alphabet, and Western punctuation is used.

1.5.2 Marathi Phonology

The phoneme inventory of the Marathi language is similar to that of many other Indo-Aryan languages. Like other alpha syllabaries, Devanagari writes out syllables by adding vowel diacritics to consonant bases. Marathi retains several features of Sanskrit that have been lost in north-Indian Sanskrit-based languages such as Hindi and Bengali, especially in terms of pronunciation of vowels and consonants. For instance, Marathi retains the original Sanskrit pronunciation. Spoken Marathi allows for conservative stress patterns in words like (ram) with an emphasis on the ending vowel sound, a feature that has been lost in Hindi. The figure 7 depicts the Character set of vowels and consonants present in Marathi Language [37].

1.5.3 Letters and Symbols used in Marathi language

Marathi is written in Devanagari alphabet and draws vocabulary from Sanskrit. Devanagari is a form of alphabet called an abugida, as each consonant has an inherent vowel (a) that can be changed with the different vowel signs. Most consonants can be joined to one or two other consonants so that the inherent vowel is suppressed. The resulting form is called a ligature. Devanagari is written from left to right. Devanagari has no case distinction, i.e. no majuscule and minuscule letters [38]. and figure 1.10 describes the number in Marathi.

Figure 1.10: Numbers

1.5.3.1 Marathi pronunciation

Sanskrit spelling was phonetic, but the spelling of modern languages written in Devanagari may only be partly phonetic in the sense that a word written in it can only be pronounced in one way, but not all possible pronunciations can be written perfectly.

1.5.3.2 ***Consonants***

Aspiration means with a puff of air, and is the difference between the sounds of the letter p in English pin (aspirated) and spit (unaspirated). Retroflex consonants, on the other hand, are not really found in English. They should be pronounced with the tongue tip curled back. Practice with a native speaker, or just pronounce as usual you'll usually still get the message across. The figure 1.11 represents the character set of consonants in Marathi Language.

क	ख	ग	घ	ङ	च	छ	ज	झ
ञ	ट	ठ	ड	ढ	ण	त	थ	द
ध	न	प	फ	ब	भ	म	य	र
ल	व	श	ष	स	ह	ळ	क्ष	ज्ञ

Figure 1.11: Character set of consonants in Marathi Language

1.5.3.3 ***Vowels***

The Marathi, vowels are added to consonants. Most of them are easy to pronounce, is slightly challenging. Marathi vowels retain much of their original Sanskrit pronunciation making some of them different from their Hindi counterparts. A notable example is (au), pronounced as owl in Marathi but as Oxford in Hindi. (Ao) is a special vowel used for loan English words, and is pronounced as in doctor [37]. The figure 1.12 represents the character set of consonants in Marathi Language

अ	आ	इ	ई	उ	ऊ	ए	ऐ
ओ	औ	अं	अः	ॲ	ऑ	ऋ	

Figure 1.12: Character set of consonants in Marathi Language

1.6 Organization of the thesis

The thesis is organized in six chapters.

Chapter 1, Introduction

This chapter introduces the speech production, speech synthesis system along with its classification.

Chapter 2, Literature Review

This chapter begins by reviewing the relevant findings for the speech synthesis system for languages spoken all over the world. Then it manily focueses on the work carried out for various Indian languages using Unit Selection Sythesis.It also produces the relevant work done for Indian languages using same method. This chapter also provides review about speech database.

Chapter 3 Speech Synthesis System For Marathi Accent Using Festvox

In this chapter, the system is developed for Marathi based speech synthesis techniques using festival and Festvox. It entails the procedure and step by step approach for the same. It explains briefly the commands involved in development of Unit based Marathi Speech synthesis System. It also conatins the highlights of Unit based Speech Synthesis System.

Chapter 4 Implementation Experimental Analysis

This chapter presents the design of Marathi Talking Caluclator. It discusses the Android platform and the needed API for developing the same. The calculator performs the basic level calculations and speaks the text out in Marathi. The functionality and design of the developed App is even presented here. The screen shots and performance of the APP is formatted in this chapter.

Chapter 5 Conclusion, Limitation And Future Scope

In this chapter, the Experimental observations and analysis are included. The various test and their respective results are perfomed to check the performance and integrity of the system.

Chapter 6, Conclusion and Future work

This chapter presents the outcome of research and its discussion along with future work.

2 Literature Review

Synthetic speech has been a dream of the humanity for centuries. To see how the present schemes, work and how they have grown to their present form, a historical review may be useful. The history of synthesized speech from the first mechanical efforts to systems that form the foundation for today's high-quality synthesizers are discussed in this chapter some milestones in synthesis-related methods and techniques will also be discussed briefly. In the 1930s, Bell Labs developed the vocoder, which automatically analyses speech into its fundamental tone and resonances. From his work on the vocoder, Homer Dudley developed a keyboard-operated voice synthesizer called The Vocoder (Voice Demonstrator) [5].

In late 1970's and early 1980's, considerably amount of commercial text-to-speech and speech synthesis products were introduced. The first integrated circuit for speech synthesis was probably the Votrax chip which consisted of cascade formant synthesizer and simple low-pass smoothing circuits. In 1978 Richard Gagnon introduced an inexpensive Votrax-based Type-n-Talk system. Two years later, in 1980, Texas Instruments introduced linear prediction coding (LPC) based Speak-n- Spell synthesizer based on low-cost linear prediction synthesis chip (TMS-5100). It was used for an electronic reading aid for children and received quite considerable attention. In 1982 Street Electronics introduced Echo low-cost diphone synthesizer which was based on a newer version of the same chip as in Speak-n-Spell (TMS-5220). At the same time Speech Plus Inc. introduced the Prose-2000 text-to-speech system. A year later, first commercial versions of famous DECtalk and Infovox SA- 101 synthesizer were introduced. Modern speech synthesis technologies involve quite complicated and sophisticated [5].

Dominant systems in the 1980s and 1990s were the MITalk system, based largely on the work of Dennis Klatt at MIT, and the Bell Labs system the latter was one of the first multilingual language-independent systems, making extensive use of natural language processing methods. Early electronic speech synthesizers sounded robotic and were often barely intelligible. The quality of synthesized speech has steadily improved, but output

from contemporary speech synthesis systems is still clearly distinguishable from actual human speech.

2.1 Speech Synthesis Efforts In Indian Language

Indian languages are syllabic in nature. The design and implementation of a unit selection based text-to-speech synthesizer with syllables and polysyllables as units of concatenation. The syllable based synthesis does not require signicantly prosodic medication, the prosodic medication that needs to be performed in the context of syllable is signicantly different from that of conventional diphone based synthesis [39].

The approximate language speakers for various languages (Speakers in Million) in India are : Assamese(13), Bengali(180), Bodo(1.2), Dogri(0.1), Guajarati(46), Hindi(422), Kannada(38), Kashmiri(5.5), Konkani(2.5), Maithili(12), Malayalam(33), Manipuri(1.5), Marathi(72), Nepali(2.5), Oriya(33), Punjabi(29), Sanskrit(0.05), Santhali(6.5), Sindhi(2.5), Tamil(61), Telugu(74), Urdu(52). The scripts in Indian languages have originated from the ancient Brahmi script. The basic units of the writing system for Indian languages are referred to as Aksharas. Aksharas are divided into Consonants(C) and vowels (V) as shown in figure 2.1

The properties of Aksharas are as follows:

- An Akshara is an orthographic representation of a speech sound in an Indian language.
- Aksharas are syllabic in nature.
- The typical forms of Akshara are V, CV, CCV and CCCV, thus have a generalized form of C*V.

Consonant- Consonant is a sound in spoken language that is characterized by a constriction or closure at one or more points along the vocal tract such as the lips, tongue and teeth.

Vowel- A vowel is a sound in spoken language, pronounced with an open vocal tract so that there is no build-up of air pressure at any point above the glottis. Consonants are divided into Gutturals, Palatals, Cerebrals, Dentals, Labials, Semi- Vowels, Sibliants/Aspirants, Miscellaneous and Compound. Vowels are most interesting class of sound in any language. Their duration in any word is also of most significance. They play a major role

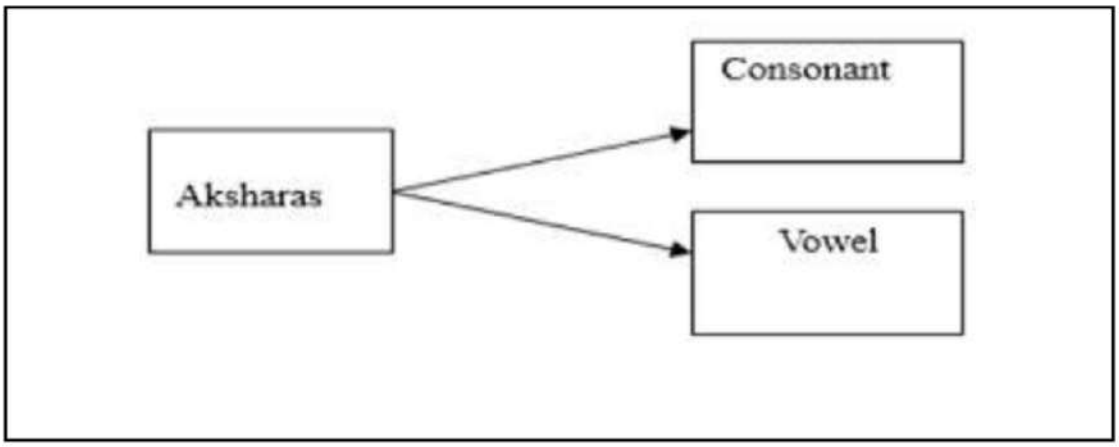


Figure 2.1: Components of Akshara

in the pronunciation of any word. Vowels are always voiced sounds and they are produced with the vocal cords in vibration, while consonants may be either voiced or unvoiced.

HINDI - Hindi is the national language of India, which is spoken by 33% of people as first language. Grapheme-to-Phoneme (G2P) converter is presented for the design and development of a good quality Hindi Text-to-Speech (TTS) system. The experimental work is carried out on sentences containing 3485 words with phonetically rich text corpora [40]. The consonant graphemes in Hindi are associated with an inherent schwa vowel which is not represented in orthography. The inherent schwa is associated with each of these consonants and the schwa is pronounced while pronouncing the word. As a rule of thumb in Hindi, a word ending schwa is always deleted. Vowels other than the schwa are explicitly represented in orthographic text with the help of specific ligatures or matras around the consonant. The algorithm is designed to perform schwa deletion which can handle schwa deletion for independent words. The efforts are taken at HP labs, to build Hindi based TTS using festival framework [40]. Indian Consortium developed TTS using Unit Selection synthesis for six major languages spoken in India i.e. Hindi, Telugu, Tamil, Bengali, Marathi and Malayalam. The key issues that are addressed in this project are varying size units of the database, Automatic segmentation, Prosody which focuses on stress patterns, Text processing, Evaluation using subjective measure and lastly aims to integrate the TTS systems in screen readers such as Non Visual Desktop Access (NVDA), ORCA and Reading Aid for Visually Impaired [41].

The quality of speech synthesis systems for varying coverage of units in the speech

database by observing the scores (response) given by the subjects is experimented. It was observed that when the coverage of units is small, the synthesizer is likely to produce a low quality speech, and there would be high variance among the scores given by different subjects. As the coverage of units increases, it increases the quality of the synthesizer and there would be less variance in the scores given by different subjects. This processing is done on Hindi Text-to-Speech Synthesis system. The observations was found that the list of words indicated the effect of loan 5 words should be considered while building the speech corpus, and also highlighted the need for good text processing front end for Indian language synthesizers [42]. Hindi speech synthesizer is developed with the different sizes of units i.e. syllable, diphone, phone and half phone. In this study the Perceptual tests evaluates that the quality of the synthesizers with different unit size indicate that the syllable synthesizer performs better than the phone, diphone and half phone synthesizers, and it also concluded that half phone synthesizer performs better than diphone and phone synthesizers [43]. Hindi TTS is developed using syllable level database for unit selection technique. An important advantage of this approach leads to reduced prosody mismatch and spectral discontinuity that occurs during syllable concatenation. The results obtained from the proposed system are far superior compared to the traditional unit based Text to Speech (TTS) synthesis system. The most important quality of this system is the improved naturalness in the synthesized speech. This is implemented using .NET and MATLAB programming languages. The noise present in synthesized speech is removed using two audio features i.e. signal energy and spectral centroid [44].

TAMIL- A polysyllabic speech synthesis system for Indian languages using the Festival framework especially for Hindi and Tamil is studied. In this study the criterion for unit selection synthesis is presented which helps in improving the quality, since the number of concatenation points would be greatly reduced. Also, the prosodic variations across the smaller units which make up the polysyllabic units would remain intact. At the application level, the polysyllable TTS built was integrated with ORCA, a free and open source screen reader software for the Linux platform. This system is applied for Hindi and Tamil TTS system [45]. The speech units are annotated with associated prosodic information about each unit, manually or automatically, based on an algorithm. An annotated speech corpus utilizes the clustering technique that provides way to select the suitable unit for concatenation, depends on the minimum total join cost of the speech unit. The entered text file is analysed first, this syllabication is performed based on the linguistics rules and the

syllables are stored separately. Then the syllable corresponding speech file is concatenated and the silence present in the concatenated speech is removed. After that discontinuities are minimized at syllable boundaries without degrading the quality. Smoothing at the concatenated syllable boundary is performed and changing the syllable pitches by time scale modification. Here the syllable is used as basic unit for speech corpus. A database has been created from various domain words and syllables. Syllable pitch modification is performed based on time scale modification. The speech files present in the corpus are recorded and stored in PCM format in order to retain the naturalness of the synthesized speech. The given text is analysed and syllabication is performed based on the rules specified. The desired speech is produced by concatenative speech synthesis approach such that spectral discontinuities are minimized at unit boundaries. It is inferred that the produced synthesized speech is preserving naturalness and good quality based the subjective quality test results [46].

In a Dravidian Tamil text-to-speech system based on the concatenative synthesis approach the database consist of the units along with their annotated information is called as the annotated speech corpus is described. The entered text file analysed first, syllabication is performed based on the syllabification rules and the syllables are stored separately. Then the corresponding speech file for the syllables are retrieved, concatenated and the silence present in the concatenated speech is removed and the synthesized speech is produced with good quality [47]. The framework of a TTS system for Tamil language is built using concatenative speech synthesis approach. Major issues considered in developing TTS are text corpus collection, recording and labeling the speech corpus, deriving letter to sound rules and prosody modeling for Tamil language can be solved and attempts to produce a naturalness in speech. The dictionary-based approach is quick and accurate. The prosody is incorporated with an intonation model using feed forward neural network (FFNN) for syllable based text to speech (TTS) synthesis system for Tamil. The features used to model the neural network include set of positional, contextual and phonological features [48].

GUJARATHI - Gujarati Text to speech (TTS) synthesis system was a built up using concatenation of Gujarati phonemes for unrestricted input Gujarati text. In this methodology, Gujarati phonemes are recorded and stored as speech database. The main advantage of this method is simple to implement which uses less memory space. The paper also describes the Gujarat alphabets with their ASCII values. The phoneme matching and speech unit searching becomes very easy with this method

[49].

TELUGU- An application was designed to build hand held devices and talking tourist aid for Hindi and Telugu languages using Festvox. The duration of Telugu speech data recorded for this purpose is around 110 minutes, while the duration of Hindi speech data is around 96 minutes. The Telugu speech corpus contains 33,417 realizations of 2,291 syllable units, and the Hindi speech corpus contains 23,179 realizations of 2,391 syllables. Festvox is used to extract pitch markers and Mel-cepstral coefficients, and then to build a decision tree for each unit (phone) based on questions concerning the phonemic and prosodic context of that unit [50]. The study proposes the data-driven synthesis method for Indian languages using syllables as basic units for concatenation. In this approach the co-articulation effects and prosodic mismatch between two adjacent units. Selection of a unit is based on the phonetic context and the prosodic parameters of its adjacent (previous or succeeding) unit. The procedure implies that a unit must be selected in way that it satisfies the local phonetic constraints and prosodic matching function to synthesize the high quality output. The experimental observation shows that the efficiency and the performance of this approach is better than that of other data-driven synthesis techniques adapted for Indian languages [51].

Unit selection synthesis inventories have coverage issues, which lead to missing syllable or diphone units. In the conventional back-off strategy of substituting the missing unit with approximate unit(s), the rules for approximate matching are hard to derive. The work is done to propose a back-off strategy for Telugu TTS systems emulating native speaker intuition. It uses reduced vowel insertion in complex consonant clusters to replace missing units. The inserted vowel identity is determined using a rule-set adapted from L2 (second language) acquisition research in Telugu, reducing the effort required in preparing the rule-set. If a consonant cluster in a syllable violates the Phonotactics constraints of Telugu, it is broken using epenthesis. The identity of this epenthetic vowel is determined by the vowel harmony rule .The vowel harmony rule states that , if the epenthetic vowel is inserted in a word medial consonant cluster, its identity is dependent on the identity of the vowel following the cluster. If the epenthetic vowel is inserted in a word final consonant cluster, its identity is determined based on the word final consonant [52].

Telugu TTS is implemented with Telugu as Unicode input. The quality of the synthesized speech is reasonably natural. The proposed approach minimizes the co articulation effects and prosodic mismatch between two adjacent units. Selection

of a unit is based on the presiding and succeeding context of the syllables and the position of the syllable. The system implements a matching function which assigns weight to the syllable based on its nature and the syllable with maximum weight is selected as output speech units [53].

SANSKRIT - Festvox tool is designed which is freely available software suite for creation and analyses of large scale speech corpora for enabling research, development and instruction in speech technologies. These include software providing state-of-the-art implementations for text analysis normal- ization, phonetically balanced subset selection, audio recording, phonetic alignment for both sentence level and multi-paragraph speech data, building durational and international models, synthetic voice building for desktop mobile applications and voice conversion support. In this the unit selection technique is performed by clunit and statistical speech generation is implemented using clustergen algorithm [54].

The work is performed for building a prototype text to speech system for Sanskrit. A basic prototype text-to-speech is built using a simplified Sanskrit phone set, and employing a unit selection technique, where pre-recorded sub-word units are concatenated to synthesize a sentence. It is important to have an optimal text corpus balanced in terms of phonetic coverage and the diversity in the realizations of the units. Sanskrit has a vast literature starting from Vedic text to latest classical literature. Sanskrit is also known for Sandhi rules. In future, more focus need to be given for implementation of these Sandhi rules for Sanskrit TTS [55].

PUNJABI - The work is presented to develop Punjabi TTS system using syllable based choice of units. While creating database the utmost care is taken to cover maximum of syllable with minimum size of database. The quality of TTS is dependent on the quality of speech corpus and quality of speech corpus depends on the speech units and the number of speech units stored in the database. In phonetics, the syllables are defined based upon the articulation. However in phonological approach, the syllables are defined by the different sequences of the phonemes. So, combination of phonemes gives rise to next higher unit called syllable. Further, combination of syllables produces larger units like morphemes and words. So, syllable is a unit of sound which is larger than phoneme and smaller than word. In every language, certain sequences of phonemes and hence syllables are recognized. The database is statistically studied with the position of syllable at three places i.e. First, Second and Middle. This statistical analysis helped to select a relatively small syllable set (of about first ten thousand syllables, that are about 0.86% of total syllables) of most

frequently occurring syllables having cumulative frequency of occurrence less than 99.81%, out of 1156740 available syllables for Punjabi language [56].

BENGALI - Bengali speech synthesizer is implemented on a mobile device with the help of technique called Epoch Synchronous Non Overlap Add (ESNOLA) based on concatenative speech synthesis technique which uses the partnemes as the smallest signal units for concatenations [57].

In Bengali TTS system the characters are converted into English ones by using Unicode texts to ASCII codes transliteration. These are converted into phonetic representation after text analysis and Grapheme to Phoneme (G2P) conversion. Prosodic analysis and waveform synthesis are done in this representation. Finally the speech is generated from the given texts [58]. Unit selection produces more natural sounding speech, so it has an advantage over the diphone concatenation. Another advantage of unit selection systems is that the database can be created automatically. Bengali TTS is developed using Unit selection and Multisyn selection algorithm [59].

The work is also presented for the design and development of unrestricted text to speech synthesis (TTS) system in Bengali language by Indian Institute of Technology Kharagpur. Unrestricted TTS system is capable to synthesize good quality of speech in different domains. In this module syllables are used as basic units for synthesis [39]. In 2011, a Bangla TTS was developed using Di-phone synthesis system with festival platform. It designed system that can convert a Unicode encoded Bangla text into human speech. Since the voice developed here is diphone concatenation based and it lacks proper intonation modeling so it produces robotic speech which degrades the naturalness quality for the synthesized speech [60].

URDU - A bi-lingual novel algorithm for the synthesis of Urdu and Sindhi language text. The devised bi-lingual algorithm uses knowledge based approach along with the hybrid rule based and concatenative acoustic methods to provide efficient and accurate conversion of Urdu and Sindhi text into the high quality speech. The algorithm has been implemented in the VB programming language with a GUI based interface [61].

MALAYALAM -The proposed Malayalam text-to-speech system is implemented in Java multimedia framework (JMF) and runs on both in Windows and Linux platforms. The proposed system provides utility to save the synthesized output for

Malayalam language [62]. Grapheme to Phoneme conversion is the process of assigning phonetic transcription to words. The study proposes a rule based G2P for Malayalam language is implemented the system has been implemented in Python. Rule-based method works better than the manual editing as well as the dictionary-based approach [63].

MARATHI - Quality of the synthesizer with different unit size indicates that the word synthesizer performs better than the phoneme synthesizer. The most important qualities of a speech synthesis system are naturalness and intelligibility. A concatenative method is applied to synthesis Marathi speech. The experimental result states that 81% of speech synthesized by the proposed method was preferred to that by the conventional method, the error rate of TTS synthesizer is around 8.22%, also the significant approach of this method is that Speech synthesis runtime was reduced for proposed method. This method suggested that greatest naturalness can be achieved by improvement in text analysis, prosody and creation of big speech database [64].

A global syllable set is designed for three Indian languages using basic languages like Hindi, Tamil and Telugu. A unit based syllable system is constructed for this study. The quality of the unit selection voices depends to a large extent on the variability and availability of representative units. Here the same syllable set is used for different types of voices. The attractive feature of this study it shows the technique to transform multiple voices to one speaker. This is achieved using ANN algorithm for voice conversion [65].

Quality of the synthesizer with different unit size indicates that the word synthesizer performs better than the phoneme synthesizer. The concatenative text-to-speech system speech synthesizer using different choice of units: Words, di-phone and tri-phone as a database. The Marathi based speech synthesis is developed using Di-phone and Trip hone syllable units. This method gives 95% of best quality of voice with more naturalness and intelligibility [1].

A new view of a synthesis database for use in unit concatenative speech synthesis is presented. The units in a synthesis database can be considered as a state transition network in which the state occupancy cost is the distance between a database unit and a target, and the transition cost is an estimate of the quality of concatenation of two consecutive units. The network can be decoded using a pruned Viterbi algorithm [12].

There are many assertive devices built up with the speech synthesis for various

languages. These devices are helpful for people with disability to live their life normally like others. A Low Memory Device Synthesizer (LMDS), which is small and fast enough to fit into small devices was developed for several commercial and non-commercial interests. This goal is achieved by using Syllable as the basic unit of synthesis to design LMDS. In this method one syllable and phone is selected as a time (the basic units) out of their multiple instances in the database to form a scaled down database. Synthesis was done merely by breaking up the given text to a sequence of available units and then concatenating them. Unit Selection algorithm is designed here for inventory building process rather than speech synthesis [66]. The speech is synthesized using artificial neural networks. There are two methods proposed to use this technique. The first method is predicting Mel-Cepstral Coefficients and synthesize speech using MLSA vocoder. The second method is building a statistical parametric synthesis using formant features. The study shows that formants are more flexible parameters than cepstral coefficients. Formants allow simple transformation to simulate several aspects of voice quality, speaker transformation etc., and also the level of understanding of speech production mechanism is better in terms of formants and their bandwidths. Mel cepstral coefficients, fundamental frequency (f0) and duration are the main components for synthesizing speech in statistical parametric synthesis. The current study mainly concentrates on Mel cepstral coefficients. Durations and f 0 are taken from the original data [67].

The Universal Digital Library (UDL) is to capture all books in digital format. A text to speech (TTS) interface for UDL portal would enable access to the digital content in voice mode, and also provide access to the digital content for illiterate and vision-impaired people. The work focuses on design and implementation of text to speech interface for UDL portal primarily for Indian languages i.e. Marathi, Hindi, Tamil and Telugu. This system is implied using festvox engine for unit speech synthesis technique. The process of digitization begins with the book being scanned page by page. Thus a digitized book is a series of images where each image corresponds to a page in the book. Each digitized page is processed using Optical Character Recognition (OCR) to obtain text in ASCII or Unicode format. The digitized text is stored in the UDL portal. As per the request from a user, this text is sent to a text to speech (TTS) system for conversion into a speech signal [68]. Phrase break prediction is very important for speech synthesis. Traditional methods of phrase break prediction have used linguistic resources like part-of-speech (POS) sequence information for modeling these breaks. In the context of Indian languages, this method implies at syllable level features and explore the use of word-terminal syllables to model

phrase breaks. A hypothesis is stated that these terminal syllables serve to discriminate words based on syntactic meaning, and so can be used to model phrase breaks. We utilize these terminal syllables in building models for automatic phrase break prediction from text and demonstrate by means of objective and subjective measures that these models perform as well as traditional models using POS sequence information. The advantage of such method is that the terminal syllables, is that they can be directly derived from the text under consideration, thus eliminating the need for additional linguistic resources like shallow parsers or POS taggers, while also eliminating the need to model phrase breaks by computationally expensive unsupervised model [69]. At IIIT- Hyderabad the Indic speech database is developed. A common set of speech databases act as benchmark speech databases to compare, evaluate and share knowledge across the institutions. The speech databases is designed since long for languages like Bengali, Hindi, Kannada, Malayalam, Marathi, Tamil and Telugu [68]. Many Indian languages shares a unique quality except (English and Urdu) with a common phonetic base, i.e., they share a common set of speech sounds.

This common phonetic base consists of around 50 phones, including 15 vowels and 35 consonants. While all of these languages share a common phonetic base, some of the languages such as Hindi, Marathi and Nepali also share a common script known as Devanagari. But languages such as Telugu, Kannada and Tamil have their own scripts. The property that separates these languages can be attributed to the Phonotactics in each of these languages, rather than the scripts and speech sounds. Phonotactics are permissible combinations of phones that can co-occur in a language. This implies that the distribution of syllables encountered in each language is different. Prosody (duration, intonation, and prominence) associated with a syllable is another property that separates these Indian languages significantly. The table 2.1 shows the languages and total duration of data available and the utterance of each particular sentence [68].

The synthetic voice is prepared for audio books using large audio database. These audio data files and text transcriptions can be used for development of Unit and Statistical based synthesizer. The text transcriptions are sentences (often incomplete) with five to ten words in each sentence. These sentences are typically recorded by a professional native speaker in a noise-free environment or in a recording studio. The total duration of audio files varies from two to ten hours. The selection of sentences used for recording is based on optimal coverage of phones in a language and hence there may not be a semantic relationship between any two successive

Table 2.1: Duration of speech databases

Language	Duration (hh:mm)	Avg. Dur of each utterance(sec)
Bengali	1:39	5.94
Hindi	1:12	4.35
Kannada	1:41	6.05
Malayalam	1:37	5.83
Marathi	1:56	6.98
Marathi	1:28	5.27
Telugu	1:31	5.47

sentences. The recording of audio files are done in a noise-free environment by a single speaker and the aspects of prosody beyond an isolated sentence are encapsulated in speech files [69]. TTS using festvox is developed for many languages all over the world. Amharic is the official language of Ethiopia. A transliteration scheme to work with Amharic scripts and incorporated Amharic phone set, syllabification rules, letter to sound rules into Festvox are developed. The voice is synthesized through unit selection method [70]. Blizzard 2008 focused on building three different voices: English full voice, English ARCTIC voice and Mandarin voice. The syllable based synthesizers are build using these three databases.

2.2 Speech Synthesis In International and National Scenario

There are various foreign languages in which work has been done or going on such as American English, Japanese, Portuguese, Arabic, Polish, Korean, German, Turkish, Mongolian, and Greek. While focusing towards Indian languages there are total 22 official languages out of which Hindi, Malayalam, Kannada, Bengali, Oriya, Punjabi, Gujarati, Telugu, and Marathi are being in focused. Different systems have been developed in these languages such as Dhvani, Shruti, HP Lab, Vani. Various institutions has been working on speech synthesis such as IIIT-H, CDAC-Mumbai, CDAC-Pune, and IIT-Madras in different languages. They have built several applications such as e-speak, a-speak, I-speak, Sandesh Pathak but in Hindi, Telugu and other languages. Marathi is an Indo- Aryan language. It is the co-official language in Maharashtra and Goa states of Wereten India and is one of the 23 official languages of India. The basic unit of Marathi writing system are the Aksharas which are an orthographic representations of speech sounds. An Aksharas are the combination of consonants and vowels. As it seen very less work has been done in Marathi [38] [39] [40]. C-DAC is the institution which has been working in the area of speech synthesis since 25 years. This institute has developed text-to-voice communication

system in Hindi, Malayalam, Bangla, Mizo and Nepali. They have developed ESNOLA based Bangla synthesis techniques. It is called as BANGLA VAANI. Other systems are Mizo Text reading system, Mozhy TTS [1], [71], [64], [72].

After Going through the efforts in speech synthesis by various researchers and scientist in national and international languages, It is also seen that not only theoretical studies are accomplished but even applications are been developed. The section below presents the speech synthesis systems for international and national scenario.

2.2.1 Application in International languages

(a) **SVOX:** It is a Text To Speech Engine from SVOX, in combination with 40+ male/female voices in more than 25 languages that allows to read aloud texts from ebook, navigation, translation and other apps. The languages in which it is available are Arabic (male), Australian English (female), Brazilian Portuguese (female), Canadian French (male/female), Cantonese (female), Czech (female), Danish (female), Dutch (male/female), Finnish (female), French (male/female), German (male/female), Greek (female), Hungarian (female), Italian (male/female), Japanese (female), Korean (female), Mandarin (female), Mexican Spanish (male/female), Norwegian (female), Polish (female), Portuguese (male/female), Russian (male/female), Slovak (female), Spanish (male/female), Swedish (female), Thai (female), Turkish (male/female), UK English (male/female), US English (male/female).

(b) **IVONA:** It is a TTS system available in 17 languages. It gives natural sounding and more accurate voices. It is compatible with Windows, UNIX, Android, Tizen, iOS based systems. The compatible languages are American, Australian, British, Welsh, German, French, Castilian, Icelandic, Italian, Canadia, Dutch, Euro pean, Brazilian, Polish, Romanian, Russian, and Danish.

(c) **CereVoice Engine:** It gives support to 9 languages. It can be easily deployed with any variety of English voices. It is available in English, French, Spanish, Italian, German, Portuguese, Japanese, Dutch, and Catalan.

(d) **eSpeak:** It supports 51 languages over the world. It allows different voices, whose features can be changed. It partially supports SSML (Speech Synthesis Markup Language).The languages supported by eSpeak are Afrikaans, Albanian, Aragonese, Bulgarian, Danish, Dutch, Cantonese, Catalan, Croatian, English, Esperanto, Estonain, Farsi, Finnish, French, Georgian, German, Greek, Hindi, Hungarian, Icelandic, Indonesian, Irish, Italian, Kannada, Kurdish, Latvian, Lojban, Malayalam,

Malaysian, Nepalese, Norwegian, Polish, Portuguese, Punjabi, Romanian, Russian, Serbain,

(e) **Google Text-to-Speech:** This TTS system supports 15 languages. It is a low level speech synthesis. The languages supported by this TTS are Dutch, English (India), English (United Kingdom), English (United States), French, German, Hindi, Italian, Japanese, Korean, Polish, Portuguese, Russian, Spanish.

(f) **Natural Reader:-** Another great text to speech software with Optical Character Recognition for both Windows and Mac users. Natural Reader also offers the ability to change speed of speech. Vast Range Of Voices US English, British English, French, German, Italian, Spanish, Swedish, Arabic, Chinese and Japanese to name a few. Optical Character Recognition (OCR) OCR enables scanning of character your text and then convert it into digital text which you can listen to in audio form or save in your computer. Great Aid For People With Learning Disabilities.

(g) **Zabaware:** Text To Speech Reader: Great text to speech software with a speech synthesizer that reads many types of documents. Various Compatible Text Types, Documents, emails, websites, clipboard contents as well as Windows dialog boxes. Great Tool For Learning Disabilities Reading disabilities as well as various concentration problems are tackled by Zabaware features.

(h) **iSpeech:** is a text to speech software with the ability to convert text to numerous audio formats not limited to only one device. Blackberry, iPhone and Android applications are available. Multiple Formats Wav, MP3, ogg, wma, aiff, alaw, ulaw, vox, MP4 to name a few. Give Voice to Any Text Dont limit yourself to PDFs or copy and pasted text. iSpeech open-source text to speech function allows you to voice-enable even text from chat applications. Multiple Devices Nobody owns just one mobile device nowadays. iSpeech has you covered with its responsive design.

(i) **Acapela Group Virtual Speaker:** It is One of the best text to speech software tools the market has to offer, particularly useful for eLearning purposes, with many compatible formats, languages and voice properties. Immense Variety Of Voices More than 70 voices and 30 languages with additional voice mood range as well as voice frequency to indicate sadness, happiness, whispering or screaming.

(j) **TextSpeech Pro:** Download-only text to speech software that reads the most popular document types such as PDFs, MS Word files and HTML. It also provides

support to impaired users.

(k) **Audio Book Maker:** If youre on the lookout for free and reliable text to speech software, then look no further. Audio Book Maker is probably the best free text to speech software. Multi-Lingual Interface English, Russian and simplified Chinese included. Customizable Speech Parameters Change speed, pitch and volume as you see fit. Highlighted Spoken Text Significantly important for online learners with learning disabilities.

(l) **TextAloud :** Developed by NextUp, TextAloud 3 is one of the most professional text to speech software tools, featuring 29 languages. Massive Range Of Voices TextAloud has partnered with ATT Natural Voices, Acapela Group, Ivona, and Nuance Vocalizer bringing you many voice options and different accents to choose from.

(m) **Read The Words:** Online text to speech software with various language options and easy-to-use interface with free version available.3 Language Options Have your text translated and read to you in English, French and Spanish.

(n) **Voice Reader:** Linguatec has produced this excellent text to speech software tool with numerous functional features. 45 Languages English, German, Italian and Spanish among others, depending on the membership you choose.

2.2.2 Application available in Indian languages

(a) **aSpeak:** It is an application available in 2 Indian languages such as Telugu and Hindi. The speech produced is intelligible, natural, and clear and can be used at different speeds.

(b) **Sandesh Pathak:** It is an application which supports 5 Indian languages namely Hindi, Marathi, Tamil, Telugu, and Gujarati. It is mainly used in agricultural based application. The text can be heard at various speeds.

(c) **Shruti:** It is a TTS system developed using Concatenative speech synthesis for two languages namely Hindi and Bengali. It is designed in such a way that it can be extended in any other languages.

(d) **HP Labs:** It is a TTS system developed in Hindi language.

(e) **Vani:** It is a system to be developed in Hindi language

(f) **Dhvani:** It is a TTS designed for 11 Indian languages such as Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, Telugu, and Pashto. The development of this system is still in progress. It is based on diphone concatenation algorithm.

(g) **Text-to-Speech Communication Systems:** These products are voice output systems for people who are able to spell out what they want to say. Messages that are typed out by the user are read aloud by the device. Many rate enhancements, like abbreviation/expansion, word prediction, stored phrases, and dialog scripts, help the user compose messages quickly, thus speeding up conversations.

(h) **ESNOLA:** It is a concatenative speech synthesis system which uses a new set of signal units in sub-phonemic level, namely, partneme as the smallest signal units for concatenation. The Epoch Synchronous Non Overlap Add (ESNOLA) algorithm is developed for concatenation, regeneration as well as for pitch and duration (prosodic) modification. The methodology of concatenation provides adequate processing for proper matching between different segments during concatenation.

(i) **Mizo Text Reading System:** Mizo Text Reading System is a Text to Speech System integrated with Mizo Optical Chracter Recognizer (OCR). The developed system is able to meet the real-life requirements of visually impaired persons with a given level of accuracy. It is a good example of software that promotes self-reliance, improves productivity and changes lives of blind, visually impaired, illiterate men and women.

2.3 Summary

This chapter overviews the literature work available for TTS. It also studies the work realted to development of TTS for various Indian languages. The study entails the focus and progress of research for primary Indian languages like Hindi, Marathi, Sanskrit, Telugu, Malayalam, Tamil, Punjabi, Urdu, and Sindi. The research related to International languages is even described. The chapter sumups the perceptive scenario for building TTS in various languages. The chapter 3 provides the tools required for building the Marathi TTS system. support. In this the unit selection technique is performed by clunit and statistical speech generation is implemented using clustergen algorithm [17].

3 Speech Synthesis System for Marathi Accent Using Festvox

3.1 Introduction

The existing application of speech is detailed in Chapter 2. In order to design and develop the proposed application, there is a need of understanding the tools available for the synthesis of speech.

3.1.1 Following are the tools for synthesis the speech system

(a) **Android:** Android is a mobile operating system developed by Google. It is used by several smartphones and tablets. Examples include the Sony Xperia, the Samsung Galaxy, and the Google Nexus One. The Android operating system (OS) is based on the Linux kernel. Unlike Apple's iOS, Android is open source, meaning developers can modify and customize the OS for each phone. Therefore, different Android-based phones often have different graphical user interfaces GUIs even though they use the same OS. Android phones typically come with several built-in applications and also support third-party programs. Developers can create programs for Android using the free Android software developer kit (SDK). Android programs are written in Java and run through a Java virtual machine JVM that is optimized for mobile devices. The "Dalvik" JVM was used through Android 4.4 and was replaced by Android Runtime or "ART" in Android 5.0. Users can download and install Android apps from Google Play and other locations. If you are unsure what operating system your phone or tablet uses, you can view the system information by selecting "About" in the Settings menu. This is also a good way to check if your device meets an app's system requirements. The name "Android" comes from the term android, which is robot designed to look and act like a human.

(b) **Xcode:** Xcode. The integrated development environment (IDE) from Apple that is used to create, compile and test Mac OS X and iOS (iPhone/iPad/iPod) applications. Xcode is an integrated development environment for macOS containing a suite of software development tools developed by Apple for developing software for macOS, iOS, watchOS and tvOS. First released in 2003, the latest stable release is version 8 and is available via the Mac App Store free of charge for macOS Sierra users. Registered developers can download preview releases and prior versions of the suite through the Apple Developer website. However, Apple recently made a beta version of version 8.0 of the software available to those of the public with Apple Developer accounts [73].

(c) **MATLAB:** MATLAB is a tool used for matrix calculations along with user interface environment. Its GUI facility helps to develop algorithms and perform various experimental and numerical analysis on dataset. The MATLAB multi- paradigm numerical computing environment allows developers and researchers to interface with programs developed in different languages, which makes it possible to harness the unique strengths of each language for various purposes. The applications of MATLAB can be found in area such as image and signal processing, communications, control systems for industry, smart grid design, robotics as well as computational finance. required speech data for generating voice is recorded in noise less environment. The voice models can be generated by unit selection or clustergen modules present in festvox. It is observed from the generated voices that clustergen voices are better than unit selection voices [73].

3.1.2 ***Festival Framework for Speech Synthesis***

The festival is a speech synthesis system [74] and it is developed in CSTR (Centre for Speech Technology Research), university of Edinburgh. Festival is compatible to work with all types of voices and also in different platforms. It is having the open source license, so any one can use freely. The core system consists of the following features [74].
d) Festival: - Festival in Linux environment. Festival is a general pre-packaged tool for development of multi-language speech synthesis systems; and it will support most of the languages in the text to speech conversion. In this project, the speech generation process is done by using Festival frame work and speech tools. The voice model is generated by using festvox frame framework, festival and speech tools.
The core system consists of the following features [74].

- Availability of Scheme interpreter makes the representation of parameters and flow of control easy. Thus recompilation of system is not needed

- C/C++ core modules: the core modules are written C/C++ this provides flexibility of implementation.

- General utterance representation: It include the flexible and powerful represent of utterances which makes it easy effective for writing function using these utterances

- Waveform I/O, formants, re-sampling: it supports common waveform for mach it also include the facility of Re-sampling and changing formats.

- Utterance, relations, and features, I/O: The structure of utterance easy to understand and implement.

- Standard data tools: A number of basic standard data tools are available like viterbi decoder.

- Audio device access and spooling: The Edinburgh speech tools library offers direct and indirect support for many types of output audio device. Also spooling is supported, allowing synthesis to continue while playing a file.

- Server/client model: client mode is provided so that a larger more powerful machine might be used as server remotely by smaller programs saving on both start up time, and resources required on the client end.

3.2 Festival Architecture

A festival speech synthesis system consists of different modules and they all together produce the synthetic speech. The modules present in festival are:

- Text Analysis and processing

- Tokenization

- Token identification

- Token to word

- Linguistic/prosody processing
- Wave form generation
- Render waveform

In festival framework utterance plays an important role in generation of synthetic speech. This framework takes an utterance and each of the modules present in it, manipulate in some way and pass on to the next module in it. Utterance consists of a set of items which are related through a set of relations. Each relation consists of a list or tree of items. Items are used to represent objects like words or segments. Relations are used to link items together in a useful way. An item may have one or more relations [74], [75].

3.2.1 ***Text analysis and processing***

The first of the three major tasks in speech synthesis is the analysis of raw text into acceptable format that can be processed in a more reasonable manner. The text analysis block takes the raw input text and produces the pronouncing format. In this all the abbreviations and numbers are expanded with respect to the context in the given text. It consists of three steps: Identifying tokens, Normalization of non- standard words (expansion of Tokens), Homograph disambiguation and chunk of utterance with sequence of pronouncing words.

3.2.2 ***Text analysis and processing***

The first of the three major tasks in speech synthesis is the analysis of raw text into acceptable format that can be processed in a more reasonable manner. The text analysis block takes the raw input text and produces the pronouncing format. In this all the abbreviations and numbers are expanded with respect to the context in the given text. It consists of three steps: Identifying tokens, Normalization of non- standard words (expansion of Tokens), Homograph disambiguation and chunk of utterance with sequence of pronouncing words.

(a) **Identifying tokens:** The text is converted to tokens depending on the white spaces and punctuation marks. Whitespaces can be viewed as separators; Punctuation can also be separated from the raw tokens. Festival converts text into an ordered list of tokens, each with its own preceding whitespace and succeeding punctuation as features of the token.

(b) **Normalization of non-standard words:** In the given input text, all the words, which are available in the dictionary, are called standard words. Numbers, symbols, abbreviations etc, which are not available in dictionary for their pronunciation, are called as non-standard words. These words are converted into full pronunciation with the following stages.

- Splitter: It will split the token not only white space but also with the punctuation.
- Type identifier: It will identify the token type for expansion.
- Token expander: the identified token is expanded depending on the context.
- Language modelling: Language model is then used to select between possible alternative punctuations of the output.

(c) **Homograph disambiguation:** In some languages like English some words are same but having different pronunciations, so there is the need of categorizing those words depending on the context in that sentence. This should be solved in text processing module.

(d) **Chunking into utterance:** In festival, chunking tokens into utterances are recognized as sentences so ideally chunks should be prosodic phrases. In some languages festival may face some problems like tokenization without white space and number pronunciation to resolve these issues; festival will follow the letter to sound rules instead of lexicons to pronounce the words.
The following examples show how the raw text is converted into sequence of pronouncing words.

- This is a pen $->$ This is a pen
- He stole \$100 from the bank $->$ He stole hundred dollars from the bank
- He stole 1996 cattle on 25 Nov 1996 $->$ he stole One thousand Nine Hundred Ninety Six cattle on twenty fifth November Nineteen Ninety Six.

3.2.2.1 ***Linguistic/Prosodic processing***

Pronounceable words are the input to this stage duration and time (f0 contours) are the parameters required to concert the pronounceable word into segment with prosody. This stage includes the following modules [74], [75].

- **Linguistic processing:** Deep linguistic processing is a natural language processing framework which draws on theoretical and descriptive linguistics. It models language predominantly by way of theoretical syntactic/semantic theory (e.g. CCG, HPSG, LFG, TAG, and the Prague School).

- **Lexicons:** Lexicons contain the mapping between the written representations and the pronunciations of words or short phrases. Speech recognition engines and speech synthesis engines each have an internal lexicon that specifies which words in a language can be recognized or spoken. The lexicon specifies how the engine expects a word to be pronounced using characters from a single phonetic alphabet.

- **Letter to sound rules:** For some languages the writing of a rule system is too difficult. Although there have been many valiant attempts to do so for languages like English life is basically too short to do this. Therefore we also include a method for automatically building LTS rules sets for a lexicon of pronunciations. This technique has successfully been used from English (British and American), French and German. The difficulty and appropriateness of using letter-to-sound rules is very language dependent,

- **Part of speech tagging :** A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'.

- **Prosodic processing:** Spoken sentence, in addition to semantic and syntactic information, carry prosodic information. Two types of prosodic information can be distinguished: linguistic prosody and emotional prosody which is non- linguistic. Here only linguistic prosody will be considered

- **Prosodic phrasing:** In linguistics, a prosodic unit, often called an intonation unit or into national phrase, is a segment of speech that occurs with a single prosodic contour (pitch and rhythm contour).

- **Intonation Accent:** Pitch accent (intonation) Pitch accent is a term used in auto segmental-metrical theory for local into national features that are associated with particular syllables. Within this framework, pitch accents are distinguished from both the abstract metrical stress and the acoustic stress of a syllable.

- **Tune (F0) o Duration of Post-lexical rules:** In fluent speech word boundaries are often degraded in a way that causes co-articulation across boundaries. A lexical entry should normally provide pronunciations as if the word is being spoken in isolation. It is only once the word has been inserted into the context in which it is going to spoke can co-articulary effects be applied.

3.2.2.2 Waveform synthesis

This is the final and most important part of festival speech synthesis system. This receives phone information, prosody for synthesis from previous block and existed voice models. By combining all these parameters, it will produce synthetic speech as output. Depending on the voice models, the waveform synthesizer is differed to access the relevant and required information from voice models and produce synthetic speech.

3.3 Installation of tools for speech synthesis

3.3.1 Setting up system (Ubuntu)

While updating Ubuntu some of the packages like Bison, synaptic manager may not installs automatically but they are required. In this case install them from the Ubuntu software centre.

3.3.2 Installation of tools

Edinburgh Speech Tools provides a set of executable, which offer access to speech tools functionality in the form of a standalone program. As far as possible, the programs follow a common format in terms of assumptions, defaults and processing paradigms. Some of the common features of these tools are

- Arguments to functions can be specified in any order on the command line. Most programs can take multiple input files, which by default have no preceding argument.

- Output by default is to standard out. The -o flag can be used to specify an output file.

- Often programs can read many formats of input file. Wherever possible, this is done automatically. If this can't be done, the -type flag can be used to specify the input file format.

- The output file format is specified by type fringe client manual section; Installation of festival speech synthesis system tool requires the sources of festival frame work and speech tools. Festvox project is another frame work to generate the new voice models from the recorded speech data base, developed in CMU (Carnegie Mellon University). To check whether the installation of festival is working or not, use the existed lexicons and voice models. Below mentioned release version tools and frame works are used for arctic speech data bases and its corresponding voice models. To generate voice models and use those models to synthesize the input text for Indic speech data base, current versions of festival, speech tools and festvox frame work are used [76], [77].

1. Download for the tools

   - $festvox-2.7.0.tar.gz$
   - $speech-tools2.1.tar.gz$
   - $festvox-2.1-release.tar.gz$
   - $festlex-CMU.tar.gz$
   - $festlex-OALD.tar.gz$
   - $festlex-POSLEX.tar.gz$
   - $festvox-kallpc.tar.gz$
   - $festival-2.1.1-current.tar.gz$
   - $speech-tools-2.1.1-current.tar.gz$
   - $festvox-2.5.3-current.tar.gz$

2. Create a folder on Desktop and copy all the above downloaded tools and enter into the folder with root permissions.

3. Install all the tools which are downloaded above

   (a) Speech-tools

      i. tarxvf speech-tools-2.1-release.tar.gz

ii. cd speech-tools

iii. ./configure

iv. make

If the errors occurs in speech_tools programming section while running make command, follow the below corrections.

- speech_tools/include/EST_Titerrator.h at line number: 212:7, 292:7 write this-¿ and save.
- speech_tools/include/EST_TNamedEnum.h at line no 133:64 write this-¿ and save.
- speech_tools/base_class/EST_Tsimplematrix.cc at line no 132:4, 130:11, 101:4 use this-¿ before set_values, just_resize and also add #include ¡string.h¿ in the header file section because this program consist of memcpy function.
- speech_tools/base_class/EST_Tsimplevector.cc add #include¡string.h¿ in the header files section because this program consist of memset function and at line no 74:7 use this-¿ before just_resize

After all the modifications are completed run "make" command. After completion of this installation run make command. This completes installation of speech tools.

1. Speech-tools

   (a) tarxvf speech-tools-2.1-release.tar.gz

   (b) cd speech-tools

   (c) ./configure

   (d) make

   speech_tools/include/EST_Titerrator.h at line number: 212:7, 292:7 write this$->$ and save.
   speech_tools/include/EST_TNamedEnum.h at line no 133:64 write this$->$ and save.
   speech_tools/base_class/EST_Tsimplematrix.cc at line no 132:4, 130:11, 101:4 use this$->$ before set_values, just_resize and also add #include ¡string.h¿ in the header file section because this program consist of memcpy function.
   speech_tools/base_class/EST_Tsimplevector.cc add #include¡string.h¿ in the header

files section because this program consist of memset function and at line no 74:7 use this-¿ before just_resize

After all the modifications are completed run ”make” command. After completion of this installation run make command. This completes installation of speech tools.

2. festival

   (a) tarxvf festival-2.1-release.tar.gz

   (b) cd festival

   (c) ./configure

   (d) make

3. festvox

   (a) tar xvf festvox-2.1-release.tar.gz

   (b) cd festvox

   (c) ./configure

   (d) make

If we run make install command all the executable tools are copied into usr/local/bin directory, we can use these tools directly from the usr/local/bin directory.

Checking festival installation

Now copy the existing voice in to festival library which is downloaded earlier by extracting using below commands.

- tarxvf festlex_CMU.tar.gz
- tarxvf festlex_OALD.tar.gz
- tarxvf festvox_kallpc16k.tar.gz

While doing this make sure that cmu dictionary, lexicons and voices are copied into the festival library now the festival is ready to generate speech for the given input text with existing voice.

To run festival follow the below commands

To make this all easier a small function doing these three steps exists. SayText simply takes a string of text, synthesizes it and sends it to the audio device.

festival>>(SayText ”Good morning, welcome to Festival”)

3.4 Recording speech data base

In order to generate synthesised speech we require the recording of database the important following are input required for database recording. The text should be selected in such a way covers the all.

3.4.1 Text selection

Text should be selected in such a way that it covers as much as possible, all phone symbols belongs to that particular language. Here a Marathi-English prompts are taken for recording, which is prepared by the people System communication machine learning research laboratory, in Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India. The most of the text in Indic English set is taken from Wikipedia dump and the text is converted to prompts by using festvox, hence all the prompts satisfy the conditions like

- Each utterance should consist of 5-15 words.
- Each word in the utterance should be among the 500 most frequent words in the text collection.
- No strange punctuation, capitals at the beginning, and punctuations at the end.

3.4.2 Speaker selection

Speaker specification depends upon the age, fluency, education, and mother etc. The selected speaker should have some awareness about the selected specification to avoid mistakes in the recording process.

3.4.3 ***Recording equipment***

To record an efficient and noise-less speech corpus, high quality recording equipment is required. A noise less environment like a specially designed recording studio is required to avoid background noise while recording the speech files. A good quality speech recorder with expandable memory slot and a headset are preferred.

3.4.4 ***Speech Recording***

The speech files are recorded in recording studio at System communication machine learning research laboratory, in Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India. A handy zoom recorder and a Logitech head-set are used for recording. Handy zoom recorder is look like a mobile and easy to operate. The distance from the microphone to a mouth and speaking volume, style is kept constant up to the end of the recording. A set of 50 or 100 utterances recorded in a single wave file. After each utterance, the speaker has to pause briefly and start the next utterance. This avoids the start-stop for each utterance. Hence the recording was probably clear and had minimal noise. During recording care must be taken about the mistakes in utterances due to wrong pronunciation or repeated pronunciation of a word while recording. If any mistakes occurred while recording are recovered either by re-recording those utterances or by correcting the corresponding transcription to suit to the utterance [68]. The recording of speech files is done with 48 kHz sampling frequency and number of bits per sample is 32-bits float type. An H4n handy zoom recorder is having the capability of recording with 4-channels; here a stereo recording (2-channels) is used for recording. The speech files are stored in Microsoft PCM wav format.

After the completion of recording of all the prompts, now all the wave files are consist of 50 or 100 utterances, so it is important to make single utterance wave files from the existed wave files. There exist many options to make slices the wave file, Zero Frequency Filtering technique (ZFF) [78], Edinburgh speech tools and etc. Zero frequency filtering technique is used to detect the voiced and unvoiced regions in a speech file, by making the duration of unvoiced region as threshold. A set of 50 or 100 utterance wave files are converted to 50 or 100 single utterance wave files.

The speech files are recorded with dual channel (2 channels) mode, so these are converted to single channel, using a tool called Audacity [79] Edinburgh speech tools can also be used for conversion. Audacity is an open source freely available tool it can be directly installed from the Ubuntu software centre and operate very easily. After the installation of audacity, open the original wav file using it. The tool shows the dual channel speech

file waveform, be obtained from and can clicking on name of the speech file and selecting split stereo to mono option from the drop down list. For making multiple channels click on file option and select the export multiple option from the drop down list. A new window opens and asks for output wave file format and output file storing directory and for output file format, need to select Microsoft PCM 16-bit format, press on export option and give the appropriate file names for the left and right channels then press ok option. This will creates the two single channel files in the output directory. Now select an exact channel signal which is nearer to mouth of the speaker. Audacity tool main window is shown below in figure 3.1.

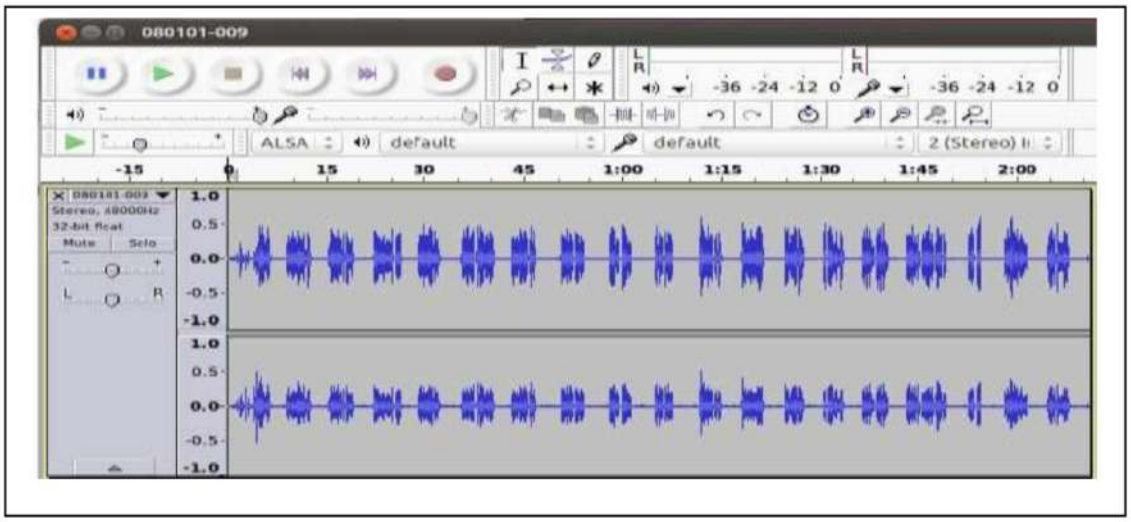


Figure 3.1: Audacity window showing the dual channel wave file

The speech files are recorded at high sampling frequency (48 kHz), it needs to be down sampled to 16 kHz sampling frequency.

3.5 Synthesis of New Voices using Recorded Speech Database

To generate the synthesized speech for the given input text, we need pre trained voice models in the speech synthesis system library. The generation of new voices involves various tasks and can be done by using Festvox tools, Festival and speech tools. The voices are generated with different synthesizing techniques like clunits methods. Clunits voice models are generated using sphinx trainer (sphinxtrain-beta), decoder (sphinx2) labeling. Speech tools available in festival are used to analyse the speech files. Festvox is a set of tools that scripts to generate new synthetic voices by using different methods.

3.5.1 Three Modules to synthesis the new voice Unit Selection

The steps involved in building a unit selection voices are basically the same as that for building a limited domain voice. Though in for general voices, in constrast to ldom voice, it is much more important to get all parts correct, from pitchmarks to labeling.

- Design the prompts: In this say time example the basic format of the utterance is the time is now, ¡exactness¿ ¡minute info¿ ¡hour info¿, in the ¡day info¿. The time is now, a little after five to ten, in the morning. Although it would technically be possible to record all of these we wish to reduce the amount of recording to a minimum.

- Record the prompts: The best way to record the prompts is to use a professional speaker in a professional recording studio (anechoic chamber) using dual channel (one for audio and the other for the electroglottograph signal) direct to digital media using a high quality head mounted microphone.

- Autolabelling the prompts: The recorded prompt can be labelled by aligning them against the synthesize prompts.

- Extract pitchmark and build LPC coefficients: Getting good pitchmarks is important to the quality of the synthesis, For the limited domain synthesizers the pitch extract is a little less crucial that for diphone collection.

- Building a clunit based synthesizer from the utterances: Building a full clunit synthesizer is probably a little bit of over kill but the technique basically works

- Testing and tuning: synthesizer can only say the phrases that it has phones for which basically means it can only say the time in the format given at the start of this chapter. Thus although you can use **SayText** you can only five it text in the write form if you expect it to works

Set the environment variables **ESTDIR** and **FESTVOXDIR** to the compiled versions of speech tools and Festvox respectively by using export command as given below

- export ESTDIR=/directory to speech tools in the system

- export FESTVOXDIR=/directory to Festvox in the system

By defining the alias for directory, we can easily refer the directory name in the use of Festvox and speech tools so many times in the process of building synthetic voices. So no need to write the entire directory every time, just we can use environment variables.

Now create a new folder/directory in the working directory to generate voice models by using the below command

mkdir institutename_dictonarylanguage_speakarname_type

like (bamu_mar_sing_cg)

Now enter into the created directory to make setup, to do this process run the below commands cd institutename_dictonarylanguage_speakarname_type

$FESTVOXDIR/src/clustergen/setup\_cg <> institutename <> dictionarylanguage <> speakername\_type$

After this, we have all the supported folders and script files. i.e. Required folders will be created and required shell script files will be copied in to the newly created directory to build the new synthetic voices. All the required script files are copied into the bin folder in current directory. Phone sets, lexical rules and all related to generating voices are generated in Festvox folder which is present in current directory. Now we need to generate prompt file, it consist of file id followed by transcription, transcription is in double quotes and entire line is enclosed in brackets. For example prompt file structure is shown below

(marthi0001 "मला वाटलं, की पाणी माझ्या हाता खाली श्वास, घेत होते.")

The prompt file is copied into the etc folder in current directory for the input wave files and it is renamed as ”txt.done.data”.

Now copy the wav files into recording folder which is already created in current directory. These wav files may not be compatible for labeling, so we need to modify these wav files in to required format, for this we have a script file in bin folder in current directory.

To run that script file, we need to run the below command in the terminal.

./bin/get_wavs recording/*.wav

By running this command (to execute the shell script), Wav files are extracted one by one from the given directory, that is recording and given to the ch_wave executable file, which is part of the speech tools in exported environment variable. This executable file takes the wav as input and converts the wav file to required format; here the scripts

requires a wav file with 16000 Hz sampling frequency and mono sound and an output wave is stored in wav folder in current directory.

Firstly we need to generate prompt files for each and every speech file present in wav folder using txt.done.data prompt file which is in etc folder. We can achieve this by using script called do_build.

./bin/do_build build_prompts

This can generate prompt-wav files, prompt-lab files and prompt-utts files in existing folders. These are used to generate festvox compatible fileids and transcription. Uniphone this contains three sentences which contain each of the Marathi phonemes once (if spoken appropriately). This prompt set is hopelessly minimal for any high quality synthesis but allows us to illustrate the process and allow you to build a voice quickly.

FESTVOXDIR/src/unitsel/setup_clunits bamu marathi sing

Alternatively you can copy in a prompt list into the etc directory. The format of these should be in the standard ”data” format as in Note the spaces after the initial left parenthesis are significant, and double quotes and backslashes within the quote part must be escaped (with backslash) as is common in Perl or Festival itself. The next stage is to generate waveforms to act as prompts, or timing cues even if the prompts are not actually played. The files are also used in aligning the spoken data.

festival -b festvox/build_clunits.scm ’(build_prompts_waves ”etc/txt.done.data”)’

Use whatever prompt file you are intending to use. Note that you may want to add lexical entries to festvox/WHATEVER_lexicon.scm and other text analysis things as desired. The purpose is that the prompt files match the phonemes that the voice talent will actually say. You may now record, assuming you have prepared the recording studio, gotten written permission to record your speaker (and explained to them what the resulting voice might be used for), checked recording levels and sound levels and shield the electrical equipment as much as possible.

3.5.2 ***Labeling***

An EHMM labeler itself consist of three modules which are used for feature extraction and HMM implementation of the speech database. A feature extraction module will generate MFCC files for corresponding speech files and ehmm, edec modules are used to perform Baum-Welch algorithm for HMM implementation and it stops Baum-Welch iterations when the difference in the average likelihood is$<$0.001. All these modules are operated by using do_ehmm script file in bin folder, by running step by step and this will generate

appropriate files. do_ehmm scripts file having several stages to label the speech data. At the starting point of labeling, all the required folders and signal processing settings file (the typical signal processing settings given in below table 5.1) are copied into the ehmm folder, which is generated in the current directory. This procedure comes under setup as the input argument. After the completion of setup an EHMM tools extracts the phone sequences and base HMM state numbers (phone list) using the festival and prompt file (txt.done.data). The phone sequences, HMM states for each and every sentence in prompt file are generated and these should be look like When an input argument is feats for the do_ehmm, it will call the feature extraction module available in festvox/ehmm folder, extracts and stores the features from the speech files in feats folder. Feature extraction tool takes feature extraction settings and wave list file as the input arguments. The typical feature extraction settings are given below table 3.1

Table 3.1: Typical signal processing settings

WaveDir	wav/
HeaderBytes	44
SamplingFreq	16000
FrameSize	160
FrameShift	80
LP order	12
CepsNumber	16
Output Feat Directory	./ehmm/feats
Input file Extension	.wav
Output file Extension	.ft

The features in the feature files are normalized to some scaling factor, here 4 is used as the scaling factor and number of Gaussians, number of connections and feature dimension files are generated. All these files are used in Baum-Welch re-estimation. Iterative Baum-Welch re-estimation is executed up to the maximum likelihood condition is satisfied. The re-estimated features are given to edec tool for aligning, this ends the labeling procedure and it will generates the label files in. /lab.

The above entire labeling procedure can be run by the below command.

./bin/make_labs prompt-wav/*.wav

Here, we can label each of the phones present in speech files and writing those labels into .lab files. These .lab files are stored in lab folder. After this process, generate utterance files for speech files by using below command, this will create all utterance files. After recording the recorded files should be in wav/. It is wise to check that are actually

there and sound like you expected. Getting the recording quality as high as possible is fundamental to the success of building a voice. Now we must label the spoken prompts. We do this my matching the synthesized prompts with the spoken ones. As we know where the phonemes begin and end in the synthesized prompts we can then map that onto the spoken ones and find the phoneme segments. This technique works fairly well, but it is far from perfect and it is worthwhile to at least check the result, and most probably fix the result by hand.

festival -b festvox/build_clunits.scm '(build_utts "etc/txt.done.data")'

Especially in the case of the uniphone synthesizer, where there is one and only one occurrence of each phone they all must be correct so its important to check the labels by hand. Note for large collections you may find the full Sphinx based labeling technique better the Section called Labeling with Full Acoustic Models in the Chapter called Labeling Speech). After labeling we can build the utterance structure using the prompt list and the now labeled phones and durations. The next stages are concerned with signal analysis, specifically pitch marking and cepstral parameter extraction. There are a number of methods for pitch mark extraction and a number of parameters within these files that may need tuning. Good pitch periods are important. See the Section called Extracting pitchmarks from waveforms in the Chapter called Basic Requirements. In its simplest case the follow may work

./bin/make_pm_wave wav/*.wav

The next stage it find the Mel Frequency Cepstral Coefficents. This is done pitch synchronously and hence depends on the pitch periods extracted above. These are used for clustering and for join measurements.

*3.5.3 **Prosody Extraction***

Now we need to calculate frequency contour (F0) and Mel cepstral parameters for the speech files with sentences in prompt file. The frequency contours are generated by using F0 parameters, if the parameters are existed in etc folder, then this process uses those F0 parameters otherwise it will take the default F0 parameters, this can be run by using the following command.

./bin/make_mcep wav/*.wav

Now we can do the main part of the build, building the cluster unit selection synthesizer. This consists of a number os stages all based on the controlling Festival script. The parameters of which are described above.

festival -b festvox/build_clunits.scm '(build_clunits "etc/ txt.done.data")'

For large databases this can take some time to run as there is a squared aspect to this based on the number of instances of each unit.

My question is how can I use the voice I built in festival? In another word how can I load the voice.

Either run festival as:

festival festvox/you_voice_name_clunits.scm

festival>(voice_sing_mar_bamu_cg)

festival>(SayText "This is a little example.")

Now, this is the time to play the text with the generated models. For this we need to run festival with and give the generated voices to play the text. And then type commands or copy the folder with your voice to

Festival/lib/voices/your_language/your_voice_name_clunits

This new voice generation procedure using some speech database. Here new voices are created using festvox tools, festival and speech tools.

The resulting voice should now work

festival festvox/bhart_indic_arctic_cg.scm

festival¿ (voice_bharti_indic_arctic_cg)

festival> (SayText " मला वाटलं, की पाणी माझ्या हाता खाली श्वास, घेत होते ")

3.6 Results

In speech synthesis, the output of the system is a synthesized speech. The below diagrams show the original and its corresponding synthesized waveforms. The new voice is generated using clustergen module with Marathi-English speech data base which is recorded in noiseless studio. Here the input to the festival is author of the danger trail, Philip steels etc, the recorded wave form for the input is shown in figure 3.2 and synthesized speech wave form with marathi voice is shown in figure 3.3

Here the input given to the festival is All you need is to have tests you can perform that will show the theory is false. and its corresponding recorded wave form is in figure

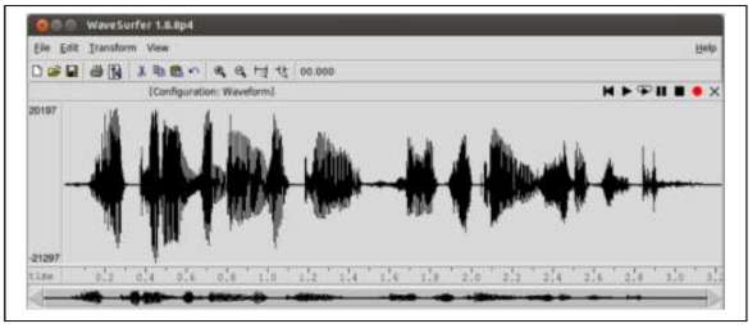


Figure 3.2: Recorded wave form bharti_arctic transcription

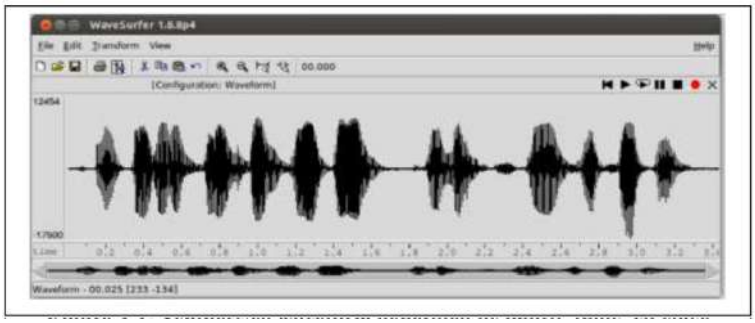


Figure 3.3: Synthesized waveform generated by bharti_indic_cg voice

3.5 and its synthesized wave forms shown in figures 3.6 and 3.7. These are generated by using voices, generated using clustergen module with bharti_arctic and sing_arctic speech data bases which is recorded in a normal room and studio respectively.

Here the input given to the festival is author of the danger trail Philip steels etc and its corresponding recorded wave form is in figure 3.8 and its synthesized wave form shown in figure 3.9 generated by using voice, which is generated by using unit selection module with Marathi speech data base, is recorded in noise less studio. Here the input given to the festival is All you need is to have tests you can perform that will show the theory is false. and its corresponding recorded wave form is in figure 3.10 and its synthesized wave forms shown in figures 3.11 and 3.12. These are generated by using voices generated, using unit selection (clunits) module with bharti_arctic and sing_arctic speech data bases which is recorded in a normal room and studio respectively.

Mel Cepstral Distortion is a technique to find the accuracy of a synthesized speech file. In this, Mel cepstrum (mcep) is calculated for both the originally recorded test files (not present in training speech data files) and synthesized speech files. Here the test transcription is used for synthesizing the speech.

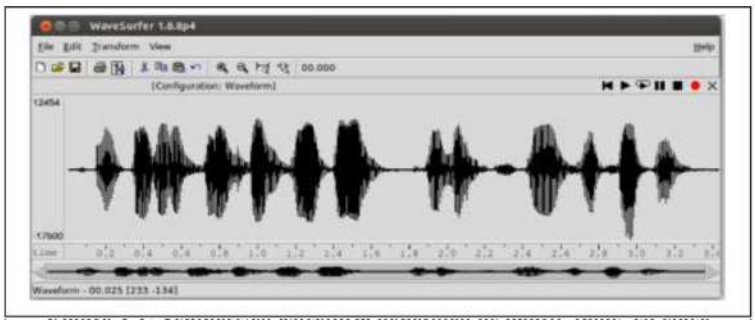


Figure 3.4: Synthesized waveform generated by sing_indic_cg voice

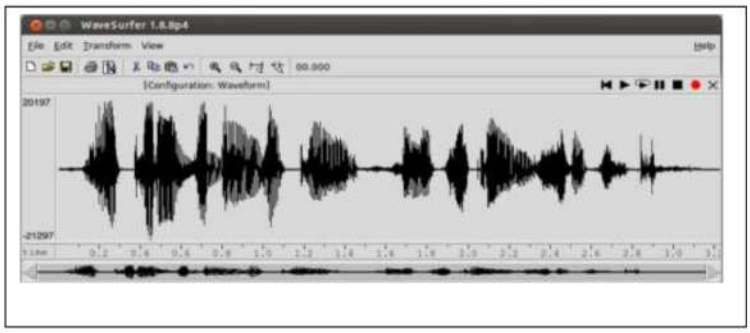


Figure 3.5: Recorded wave form bharti_arctic transcription

3.7 Summary

This chapter outlines the method of unit selection based concatenation speech synthesis technique is used to synthesize the output speech. The flow diagram for building the Unit selection based Marathi TTS system is discussed here. The frame work of Festival and Festvox and its detailed description is also mentioned in this chapter. Festival speech synthesis system is used for this purpose. The creation of database and the setup to build the recording of utterance is also described.

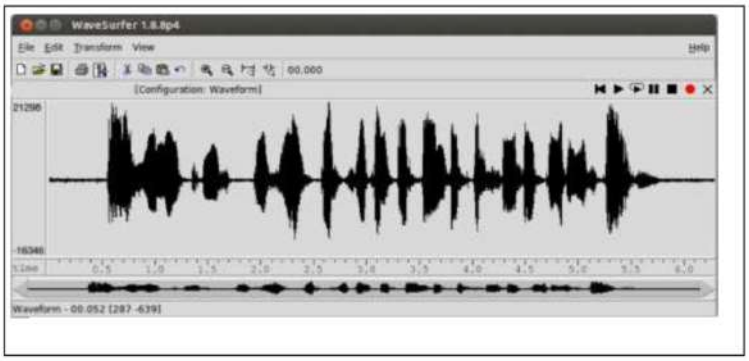


Figure 3.6: Recoded waveform from sing_indic transcription

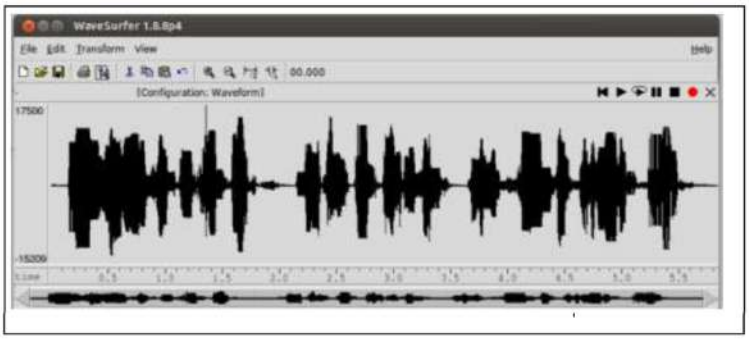


Figure 3.7: Synthesized waveform generated by using bharti_arctic_clunits voice

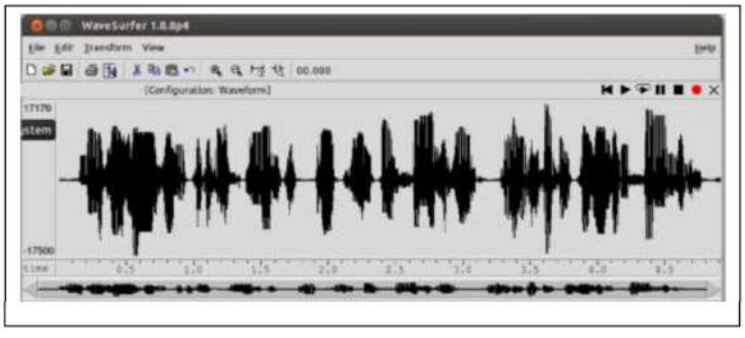


Figure 3.8: Synthesized waveform generated by using sing_arctic_clunits voice

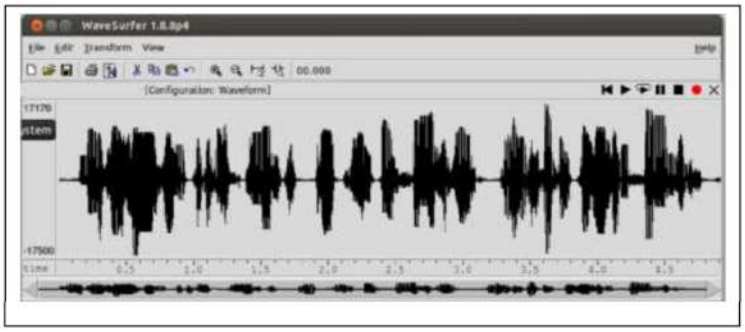


Figure 3.9: Synthesized waveform generated by using bharti_arctic_clunits voice

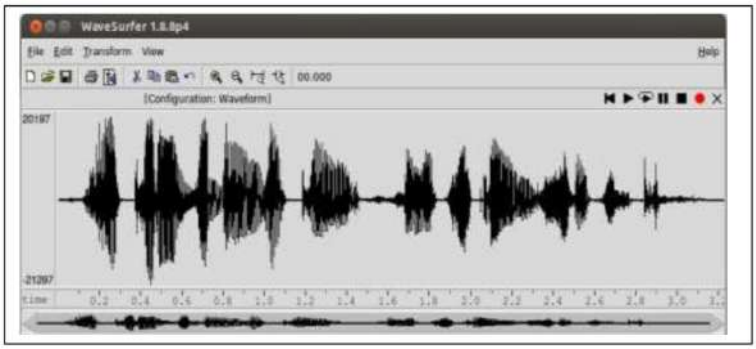


Figure 3.10: Recorded waveform bharti_arctic transcription

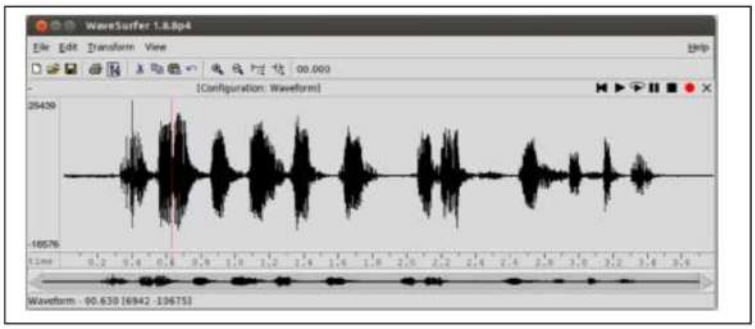


Figure 3.11: Synthesized waveform generated by using bharti_indic_clinits voice

4 Implementation

4.1 Introduction

Text-to-speech synthesis (TTS) is an important component of the speech interfaces in which the low bandwidth text is converted into speech on the assertive devices. It is also useful in applications like visual interface for embedded devices. In this chapter, the experimental analysis is performed to synthesize the Marathi speech with help of festival and also by building the Marathi calculator. Festival provides a highly flexible and modular architecture for TTS conversion. Speech synthesis necessarily produces artificial human speech. The quality of synthesized speech can be evaluated using different methods. Several methods like segmental evaluation methods, sentence level tests, comprehension tests, prosody evaluation, intelligibility of proper names and field tests etc. have been suggested to evaluate speech quality in general and these methods are also suitable to measure overall quality or acceptability of synthetic speech.

There experiments were perfomed for analysis the synthesis of speech.

4.2 Experimental Analysis for the festvox and speech tools

4.2.1 Experiment for first objective to analyse the performance of festival Festvox for Marathi Language.

Database Creation: - The speech files are recorded in recording studio at System communication and machine learning research laboratory, in Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India. A handy zoom recorder and a Logitech head-set are used for recording. Handy zoom recorder is look like a mobile and easy to operate. The distance from the microphone to a mouth and speaking volume, style is kept constant up to the end of the recording. The recording is done with utmost are and with noiseless

Table 4.1: Design Speech database

Sr.No	The Raw text
1	kaarand∼a aapalayaakad:ei tii padadhata naahii
2	paraachiina bhaaratiiya arathavayavasathaa atishaya sun:dara hootii
3	san:darabha maatara jaramanii yeithiil
4	kaahii vishishhat:a dhayeiyaanei pareiriita aajakaala phaaracha kamii lookan: saapad:atiil
5	asoo haa leikha ajhat:eika asuuna a vara chan:dara yeitoo
06	savatah:chayaa sadasaya paanaavara ashii sahii karand ayaachii garaja naahii
07	in:galan:d:a yaa gat:aata vijaya mil:avuuna dusarayaa pheiriita paraveisha keilaa
08	hei dhayaanaata gheiuuna tulaa jei yoogaya vaat:eila tei kar
09	karanaat:akaata keival:a kananad:a adhikrxta aahei
10	ghara deitaanaa saravaparathama maraat:hii maand asaalaa paraadhaanay
11	tumahii kaahii maaragadarashana karuu shakaala kaa dhanayavaad
12	pund ei hei mahatavaachei audayoogika kein:dara aahei
13	in:garajii bhaashhaa aahei roomana lipii aahei
14	chhatarapatii shivaajii mahaaraaja yaan:chayaa patanii
15	vikipiid:iyaa haa eika janj aanakoosha aahei
16	poopa kaleimein:t:a paachavaa haa chaudaavayaa shatakaatiila poopa hootaa
17	tayaanan:tara tayaan:nii bhaata sheitii va itara kalaa kaushalaya mil:avalei
18	pud:hei uchacha shikashhand aasaat:hii tayaa jaramaniilaa geilayaa
19	hei kaagadapatara aahei saachaa deisha maahitii o sat:areiliyaa vishhayii
20	kaarand a tuu maajhii aaii aaheis

background environments. A set of 500 utterances recorded in a single wave file. the speaker was asked to pause inbetween start the next utterance. This avoids the start-stop for each utterance. The recording of speech files is done with 48 kHz sampling frequency and number of bits per sample is 32-bits float type. An H4n handy zoom recorder is having the capability of recording with 4-channels; here a stereo recording (2-channels) is used for recording. The speech files are stored in Microsoft PCM wav format. After the completion of recording of all the prompts, all the wave files are consist of 500 utterances, so it is important to make single utterance wave files from the existed wave files. There exist many options to make slices the wave file, Zero Frequency Filtering technique (ZFF), Edinburgh speech tools and etc. Zero frequency filtering technique is used to detect the voiced and unvoiced regions in a speech file, by making the duration of unvoiced region as threshold. A set of 1000 utterance wave files are converted to 1000 single utterance wave files. Table: 4.1 show that for example files of speech database.

From the database the ten sentences selected for the analysis of the festvox tool. The

synthesised speech is evaluated for naturalness and undestandability.

4.3 Synthsized speech Evaluation methods

With respect to TTS system, testing evaluation is significant; as it is important to test the system before deployment. For example: a screen reader application which is mainly designed for people with visual disability should fulfill their requirement of reading out the digital content. Testing and Evaluation of TTS system will be useful until the quality of synthetic speech becomes natural. The quality of synthesis speech output is calculated at phoeneme, word and sentence level. There are varoius test for segmental evaluation level. The test like diagnostic rhyme test (DRT) , modafed rhyme test MRT, are used for this type of testing of TTS system. The test are also performed at sentence level like Harvard Psychoacoustic Sentences, Haskins Sentences, Semantically Unpredictable Sentences (SUS). The test is also performed at comprehension which works for single phoneme. This test involves the identifying each word it is difficult to get the deeper information about the context, because in intelligibility test evaluator will emphasize only on recognition of each word without concentrating on the meaning of the sentence. Comprehensibility test should be carried out when the system achieves the intelligibility up to acceptable level otherwise it is meaningless to carry out comprehensibility test for unintelligent system; as intelligibility has a strong impact on comprehension.Questions can be factual and inferential. For comprehensibility test, data should be wisely created/chosen as there may be chance that evaluator has prior knowledge of the paragraph/story played and they may answer it based on their prior knowledge. Data for such test can be picked from stories of one language e.g.: Punjabi translate it in required language e.g.: Hindi, by this method chances of prior knowledge of such data reduces.Prosody evaluation is again an important part of evaluation techniques. Prosody is also one of the least developed parts of existing TTS systems and needs considerable attention for the research in the future.

So out of all existing test, we have used MOS test as it the most widely used. The test also performs the overall quality of evaluation of speech data produced. MOS can be performed for both the TTS parameters i.e naturalness and intelligibiility. MOS. The Intelligibility refers to the accuracy with which each word is pronounced so that normal listener can understand the spoken word or phrase. In this method, focus of evaluator should be on intelligibility of synthetic speech.MOS is the arithmetic mean of the scores given by all the evaluators. The experimental analysis is performed for Marathi language.. The quality of synthesized speech is evaluated with the three methods.

4.3.1 Mean Opinion Score (MOS)

The mean Opinion Score (MOS) is test used to measure the quality of output synthesized speech in voice communications. It is subjective test which uses the average to calculate the best performance of the system. The bandwidth quality can be determined using the Compressor/decompress or (codec) systems and digital signal processing (DSP). It is given in terms of the best codecs provide the most bandwidth conservation while producing the least degradation of the signal. With regards to determine the best voice quality it requires the human interpretation. In order to calculate the MOS test, a number of listeners rate the quality of variuos sentences read aloud over the communications circuit by male and female speakers.

A listener gives each sentence a rating which is as follows:
(1) bad;
(2) poor;
(3) fair;
(4) good;
(5) excellent.
The MOS is the arithmetic mean of all the individual scores, and can range from 1 to 5

4.3.1.1 Analysis of Mean Opinion Score (MOS)

MOS is calculated for subjective quality measurement. It is calculated for the synthesized speech using the Unit selection synthesis. It was counseled to the listeners that they have to score between 01 to 05 (1) bad; (2) poor; (3) fair; (4) good; (5) excellent for understandability. The mean of the scores given by ten individual for ten sentences of the Unit selection approach is shown in table 4.2. The mean and variance of the score obtained according to the subject using unit selection it is observed that from table 4.3 and table 4.4 mean scores increases with the increase in the syllable coverage. The detail sentences and label used for the unit selection based speech synthesis is described in figure 4.1.

4.3.1.2 Peak Signal-To-Noise Ratio (PSNR) and Mean Squared Error (MSE)

PSNR is defined as the ration of the maximum possible power of a signal to the power of corrupting noise that affects the quality of signal affecting of its representation. The calculation is done with the original signal to the error noise which is introduced during the process. MSE is a risk function, corresponding to the expected value of the squared

Sr.No	The Original Sentence	Label Used for Original Speech File	Label Used for Synthesis Speech File
1	म्हणून मी खोटे बोलणे योग्य नाही	A001	a001
2	त्याचप्रमाणे सर्वोच्च गोलंदाजी ऐवजी सर्वोत्तम गोलंदाजी पाहिजे	A002	a002
3	पुणे हे महत्वाचे औद्योगिक केंद्र आहे	A003	a003
4	पुणे शहर हे महाराष्ट्र राज्याची सांस्कृतिक राजधानी म्हणून ओळखले जाते	A004	a004
5	छत्रपती शिवाजी महाराज यांच्या पत्नी	A005	a005
6	विकिपीडिया हा एक ज्ञानकोश आहे	A006	a006
7	पोप क्लेमेंट पाचवा हा चौदाव्या शतकातील पोप होता	A007	a007
8	त्यानंतर त्यांनी भात शेती व इतर कला कौशल्य मिळवले	A008	a008
9	पुढे उच्च शिक्षणासाठी त्या जर्मनीला गेल्या	A009	a009
10	हे कागदपत्र आहे साचा देश माहिती ऑस्ट्रेलिया विषयी	A0010	a0010

Figure 4.1: Sentences and label used for unit selection based speech synthesis

error loss or quadratic loss. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate.

The PSNR and MSE method was used for subjective quality measure of speech synthesis based on hidden Markov model and unit selection approach. Table 4.4 represents the MSE and PSNR values for unit selection based speech synthesis. UNIT based speech synthesis using MSSE and PSNR is shown in table 4.3.

*4.3.2 **Mel-frequency cepstral coefficient***

In speech processing, the mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a speech, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency. Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC. The signal based synthesis quality measure is experimented for unit selection and Hidden Markov Model based speech synthesis. For the performance variation, we calculated the mean, standard deviation and variance as a statistical measure. The detail sentences and label used for the unit selection based speech synthesis is described in figure 4.2.

The MFCC-mean based performance of unit selection based synthesis is shown in table 4.6. Table 4.7 represents the detail of standard deviation of MFCC for unit selection speech synthesis.

Table 4.2: Unit selection speech synthesis of the scores given by each subject for each synthesis system

Sentence	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9	Sub10
1	5	5	5	5	4	4	5	4	4	5
2	5	5	4	5	5	4	5	4	4	5
3	4	4	5	4	3	3	4	2	5	4
4	5	4	4	5	4	4	5	5	5	5
5	5	5	5	5	4	4	5	3	3	5
6	5	4	5	5	5	4	5	4	4	5
7	4	5	4	4	4	4	4	4	4	4
8	4	4	5	4	4	5	4	5	5	4
9	5	3	5	5	3	4	5	3	5	5
10	5	5	4	5	4	4	4	4	5	4

Table 4.3: Mean and variance of the scores obtained across the subjects from Unit selection

Subject	Unit Selection Method	
	Mean Score	Variance
Sub 1	4.56	0.25
Sub 2	4.23	0.52
Sub 3	4.03	0.79
Sub 4	4.56	0.25
Sub 5	4.10	0.43
Sub 6	4.03	0.37
Sub 7	4.56	0.25
Sub 8	4.96	0.72
Sub 9	4.16	0.62
Sub 10	4.63	0.24

4.4 Text To Speech Synthesis Application In Marathi Language

4.4.1 Marathi Talking Calculator using MATLAB for Computer System

A talking calculator has a built-in speech synthesizer that reads aloud each number, symbol, or operation key a user presses in Marathi. Marathi Talking Calculator in MATLAB, a talking calculator's main feature is the ability to talk in Marathi. The idea behind it is to make simple calculating tasks much more convenient and efficient to someone who cannot easily read a standard display. One does not need to give up any features and functions to get a calculator with Marathi talking capabilities.

Table 4.4: MSE and PSNR values for Unit selection based speech synthesis

Sr. No	Original Speech File	Synthesized File	M.S.E	P.S.N.R
1	A1	a1	7.94	3.30
2	A2	a2	4.57	6.72
3	A3	a3	1.02	3.21
4	A3	a4	3.70	4.20
5	A5	a5	7.61	2.57
6	A6	a6	4.32	1.26
7	A7	a7	8.06	7.56
8	A8	a8	7.20	1.29
9	A9	a9	9.25	3.24
10	A10	a10	7.01	4.08
Average			6.168	3.743
Quality (100-Average)			93.83	96.26

Table 4.5: Comparative result of Unit speech synthesis

Sr. No	Approach of Synthesis	MSE (%)	PSNR (%)
1	Unit Selection	93.83	96.26

The talking calculator is implemented using MATLAB. It is very easy to learn, versatile and very useful for engineers and other professionals. MATLAB is a special-purpose language that is an excellent choice for writing moderate-size programs that solve problems involving the manipulation of numbers. MATLAB is viewed by many users not only as a high-performance language for technical computing but also as a convenient environment for building graphical user interfaces. The figure 4.1 decipts shows the computer based Marathi speech talking calculator application.

4.4.1.1 ***Database Creation MATLAB***

The total number of words with probability 121, utterance and the data was collected in 1 sessions so, the overall 121/- vocabulary size are collected for the database.

4.4.1.2 ***Acquisition setup***

To achieve a high audio quality, the recording took place in the normal room without noisy sound and effect of echo. The sampling frequency for all recordings was set to 0to sit in front of the microphone with the distance of about 12-15 cm. The speech data was collected with the help of microphone reattach and Matlab software using the single channel.

Sr.No	The Original Sentence	Label Used for Original Speech File	Label Used for Synthesis Speech File
1	त्यांचे गायक म्हणून गाजलेले काही चित्रपट	A1	a1
2	ऑस्ट्रेलियाला देशांतर करण्या आधी आय	A2	a2
3	दिंडोरी तालुक्यातील एक गाव आहे	A3	a3
4	ही स्पर्धा चेन्नई व चंदिगड येथे होईल	A4	a4
5	आणि ती तुला ठाऊक नाही	A5	a5
6	माणूस हा वाघांचे नैसर्गिक भक्ष्य नाही	A6	a6
7	मध्यप्रदेश राज्यातील एक प्रमुख शहर आहे	A7	a7
8	आणि ती तुला ठाऊक नाही	A8	a8
9	मायक्रोसॉफ्ट विंडोज अधिक प्रगत आहे	A9	a9
10	म्हणून मी खोटे बोलणे योग्य नाही	A10	a10

Figure 4.2: Sentences and label used for unit selection based speech synthesis

Table 4.6: Parameters used for the database creation

Sr. No	Parameter	Value
1	Parameter	Value
2	Sampling Rate	16000
3	Speakers	Dependent
4	Condition of Noise	Normal
5	Accent	Marathi
6	Pre emphasis	1-0.9/(z-1)
7	Window type	Hamming, 25 milliseconds
8	Window step size	20 millisecond

The preprocessing was done with the help of computerized Speech Laboratory (CSL). The CSL is one of the analysis systems for speech and voice. CSL is an input/output recording device for a PC, which has special features for reliable acoustic measurements. The figure 4.2 described the setup of database creation using Computerized Speech Lab.

The parameter considered for recording are described in table 4.6

In this research we have implemented the interface for windows default calculator application via communication interface. The spoken word was successfully recognized and then the action was taken towards the command in the calculator. The table 4.7 presents the Demo values for the MATLAB Marathi Speech Talking Calculator

Synthesized Speech										
Original		a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
Speech	A1	0.12	3.24	2.31	1.39	3.42	3.00	2.98	4.09	7.01	1.23
Signal	A2	1.28	0.00	2.11	2.45	7.63	1.90	4.02	1.22	8.01	3.04
	A3	1.45	3.21	0.08	3.25	1.99	2.84	3.92	2.96	2.09	6.70
	A4	3.02	1.23	2.56	0.89	2.78	4.45	1.67	1.98	2.67	3.45
	A5	2.40	1.45	4.43	2.93	0.02	3.92	6.05	4.67	7.00	3.67
	A6	3.89	8.09	3.98	2.73	3.67	0.67	2.84	4.03	3.89	4.92
	A7	1.89	1.56	2.03	2.10	4.92	3.67	1.46	1.27	3.52	7.38
	A8	2.93	3.67	2.73	3.72	3.56	2.73	3.87	0.78	2.67	3.64
	A9	1.77	2.73	4.33	4.89	2.78	3.87	2.74	4.87	1.09	3.02
	A10	2.87	2.98	1.87	1.90	2.76	1.56	4.02	1.78	2.03	4.32

Figure 4.3: The performance of MFCC-Mean based Unit selection speech synthesis

4.4.1.3 Performance Evaluation MOS on MATLAB based Marathi Speech Talking Calculator

MOS is calculated for subjective quality measurement. It is calculated for the synthesized speech using the Unit selection synthesis. It was counseled to the listeners that they have to score between 01 to 05 (1) bad; (2) poor; (3) fair; (4) good; (5) excellent.The table 4.7 shows the Unit selection speech synthesis of score for each subject.

4.4.1.4 The quality of speech as per the above MOS table is as follows

Graphical Representation. Percentage Evaluation for Synthesized output speech for MATLAB application for figure 4.8. The table 4.8 shows the understandability of quality voice.

The table 4.8 shows that the 85% of individuals rated the quality of synthetic speech is good and understandable. The only 15% speech is perceptible and not clearly produced. Thus Unit selection method provides the naturalness and understand ability, the two important parameters of TTS system.

Synthesized Speech											
Original		a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
Speech	A1	0.45	1.20	1.89	3.90	3.87	2.89	4.78	3.87	4.50	2.67
Signal	A2	1.23	0.63	2.76	2.09	1.98	2.76	1.23	1.65	4.67	6.01
	A3	4.63	3.98	1.05	1.92	2.78	2.78	1.89	1.29	3.02	4.87
	A4	3.09	2.09	4.09	1.83	4.93	3.09	4.78	3.98	2.98	2.98
	A5	1.88	1.02	3.02	3.09	2.87	1.09	4.09	3.98	3.98	4.02
	A6	3.09	4.87	3.98	2.98	1.02	0.93	1.82	2.89	4.09	4.09
	A7	2.98	3.09	2.99	4.98	2.83	8.01	1.92	1.89	3.00	3.09
	A8	4.01	3.92	4.98	4.09	2.93	4.98	1.09	0.98	1.77	4.83
	A9	2.93	4.01	2.93	6.09	2.98	1.92	2.93	4.63	1.09	1.92
	A10	3.09	1.82	1.90	3.92	4.92	3.84	4.98	4.53	2.32	0.21

Figure 4.4: The performance of MFCC-Mean based Unit selection speech synthesis

The table below shows the comparative performance of both Unit for accent recognition using MFCC, MSE and PSNR techniques.

4.5 Building Android application for Marathi Talking Calculator

A talking calculator has a built-in speech synthesizer that reads aloud each number, symbol, or operation key a user presses in Marathi. Marathi Talking Calculator in Android, a talking calculator's main feature is the ability to talk in Marathi. The idea behind it is to make simple calculating tasks much more convenient and efficient to someone who cannot easily read a standard display. One does not need to give up any features and functions to get a calculator with Marathi talking capabilities.

Android Android is basically an operating system for smartphones. But we find now integrated into PDAs, touch pads or televisions, even cars (trip computer) or notebooks. The OS was created by the start-up of the same name, which is owned by Google since 2004.

Low Investment High ROI Android comparatively has a low barrier to entry. Android provides freely its Software Development Kit (SDK) to the developer community which minimizes the development and licensing costs. The development costs can be divided into three stages: Stage1 application development, Stage2 testing, and Stage3 hardware cost for testing and deploying the android mobile application.

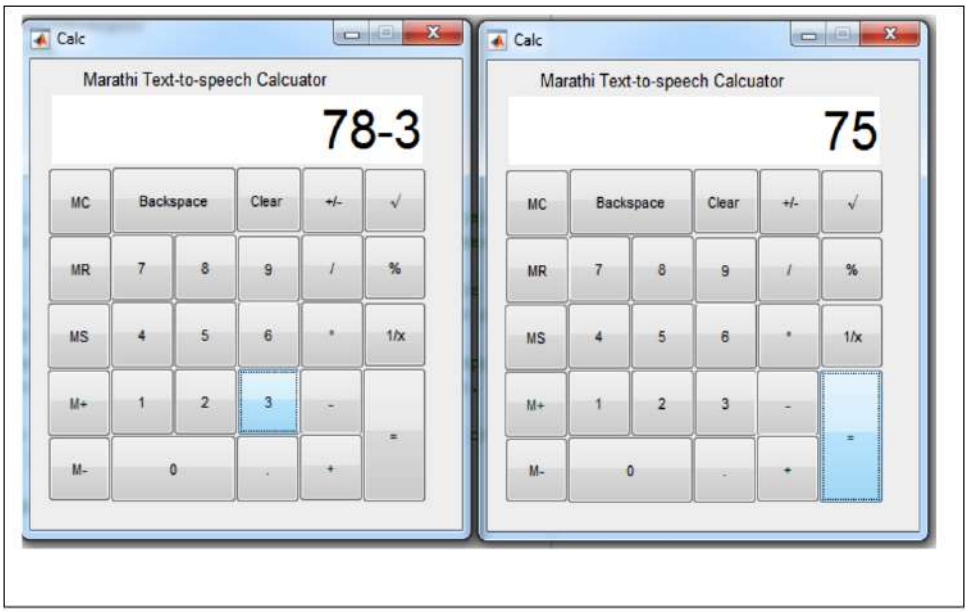


Figure 4.5: Computer Based Marathi Speech Talking Calculator Application

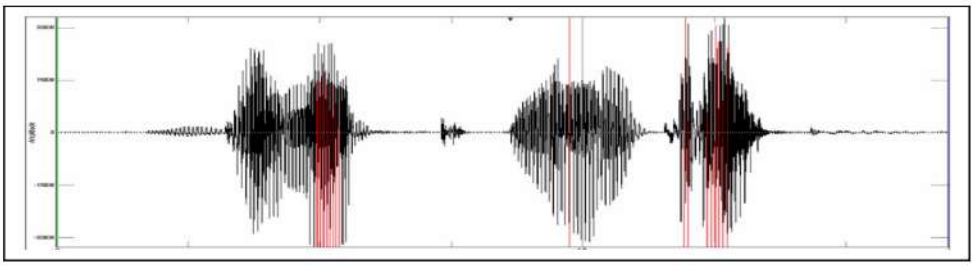


Figure 4.6: Digitization of speech signal vocabulary word

Open Source Get the open source advantage from licensing, royalty-free, and the best technology framework offered by the Android community. The architecture of the Android SDK is open-source which means you can actually interact with the community for the upcoming expansions of android mobile application development. This is what makes the Android platform very attractive for handset manufacturers wireless operators, which results in a faster development of Android based phones, and better opportunities for developers to earn more. Thats the magic of Android.

Easy to Integrate Are you looking for complex technical customization and integration of a web application or just a smartphone application you already have? Yes. Then an android app can be the right solution for you. The entire platform is ready

Figure 4.7: Result for the MATLAB Marathi Speech Talking Calculator

Sr.No	Numb1	Opr1	Numb2	Opr2	Result	Time in Secand	Result in Speech
१	२	+	६	=	८	१.५	आठ
२	५	+	९	=	१४	२.६	चौदा
३	५९	+	८९	=	१४८	३.६	एकशे अठ्ठेचाळीस
४	६६	+	८४	=	१५०	३.६	एकशे पन्नास
५	८७	+	९३	=	१८०	३.६	एकशेऐंशी
६	९६८	+	७६९	=	१८३७	४.२	एकहजार आठशे सदतीस
७	२३६७	+	९५६३	=	११९३०	५	अकराहजार नऊशे तीस
८	५८६	+	३२१	=	९०७	३.६	नऊशे सात
९	५८२	+	६३९	=	१२२१	४.२	एकहजार दोनशे एकवीस
१०	४९३	+	३५७	=	८५०	३.६	आठशे पन्नास
११	१२.१५	+	९५.६९	=	१०७.८४	५	एकशे सात पुर्ंका चौऱ्याऐंशी
१२	७५.९	+	५६.७८	=	१३२.६८	५	एकशे बत्तीस पुर्ंका अडुसष्ठ
१३	१०	-	६	=	४	१.५	चार
१४	१५	-	९	=	४	१.५	चार

for customization. You can integrate and tweak the mobile app according to your business need. Android is the best mobile platform between the application and processes architecture. Most of the platforms allow background processes helping you to integrate the apps.

Multiple Sales Channels Unlike other mobile platforms, Android applications can be deployed in different ways. You do not have to rely on a single market to distribute your applications. You can use third-party application marketplace (especially in Google Android Market), but you can also form your own distribution and sales channel: applications for vertical markets, to develop new application stores, and also place it on your website. You build it, you publish it. With your choice of promotional strategy, you can reach your end users through multiple channels.

Easy Adoption Android applications are scripted in Java language with the help of a rich set of libraries. Anyone can build Android applications with the knowledge of Java. According to a recent survey, a lot of Java programmers find it easy to adopt and script code for mobile applications in the Android OS. It is now very beneficial for Java developers to transition the code script into a mobile application, and can also implement android application development services in the app. The talking

Table 4.7: Synthesis of the scores given by subject

Sr.No	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9	Sub10	Average
1	5	4	5	5	5	5	5	4	5	5	4.6
2	5	4	5	5	5	5	5	4	5	5	4.6
3	5	4	5	5	5	5	5	4	5	5	4.6
4	5	4	5	5	5	5	5	4	5	5	4.6
5	5	4	5	5	5	5	5	4	5	5	4.6
6	5	4	5	5	5	5	5	4	5	5	4.6
7	5	4	5	5	5	5	5	4	5	5	4.6
8	5	4	5	5	5	5	5	4	5	5	4.6
9	5	4	5	5	5	5	5	4	5	5	4.6
10	5	4	5	5	5	5	5	4	5	5	4.6
Average											4.5

Table 4.8: Understandable voice quality

High quality and Understandable	85%
Perceptible Quality speech	15%

calculator is designed especially for Marathi language. The Numbers and operations performed and are spoken in Marathi. The figure below depicts the layout of Marathi talking calculator app. In this APP the input numerals are spoken in Marathi when the number is pressed. The attractive feature of APP is it contains two special buttons for clear and go back which again speaks out when pressed. This again gives the clear instruction to the user for performing the particular operation. The output speech is in c ear and proper format. It performs all the basic arithmetic operations and reads out the each operation in Marathi. The result is even spoken out in Marathi with proper digit place value of the number. The figure 4.9 shows the output screen of android based Marathi speech talking calculator.

The Special features of developed android APP are as follows:

- Speaks out each number
- Speaks out the operations
- Speaks out the result with proper digit places in Marathi
- The voice produced is clear and correct.

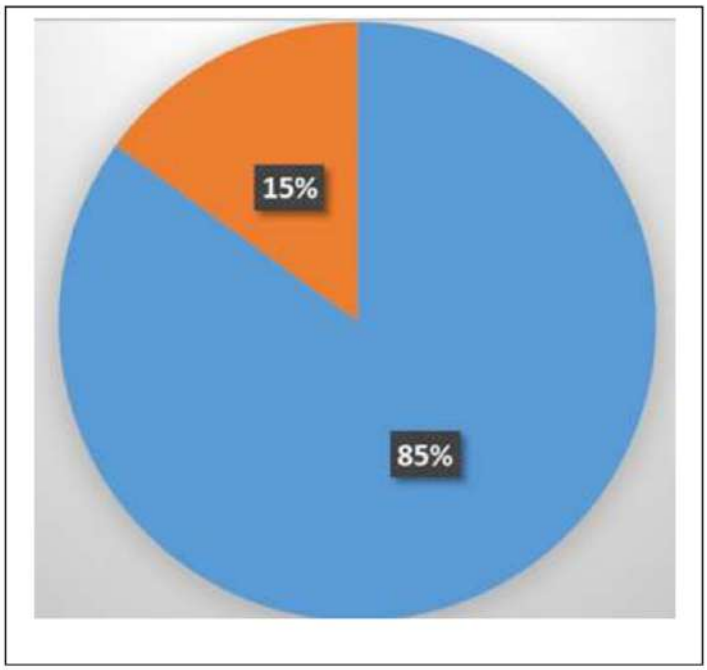


Figure 4.8: Result for the MATLAB Marathi Speech Talking Calculator

4.5.1 Database Creation Android

The total number of words with probability 121, utterance and the data was collected in 1 sessions so, the overall 121/- vocabulary size are collected for the database. The table 4.15 shows the design of database values for the reuired android application.

4.5.2 Acquisition setup

To achieve a high audio quality, the recording took place in the normal room without noisy sound and effect of echo. The sampling frequency for all recordings was set to be 16000 Hz at the room temperature and normal humidity. The speakers were asked to sit in front of the microphone with the distance of about 12-15 cm. The speech data was collected with the help of microphone realtech and matlab software using the single channel. The preprocessing was done with the help of computerized Speech Laboratory (CSL). The CSL is one of the analysis systems for speech and voice. CSL is an input/output recording device for a PC, which has special features for reliable acoustic measurements.

In this research we have implemented the interface for windows default calculator application via communication interface. The spoken word was successfully recognized

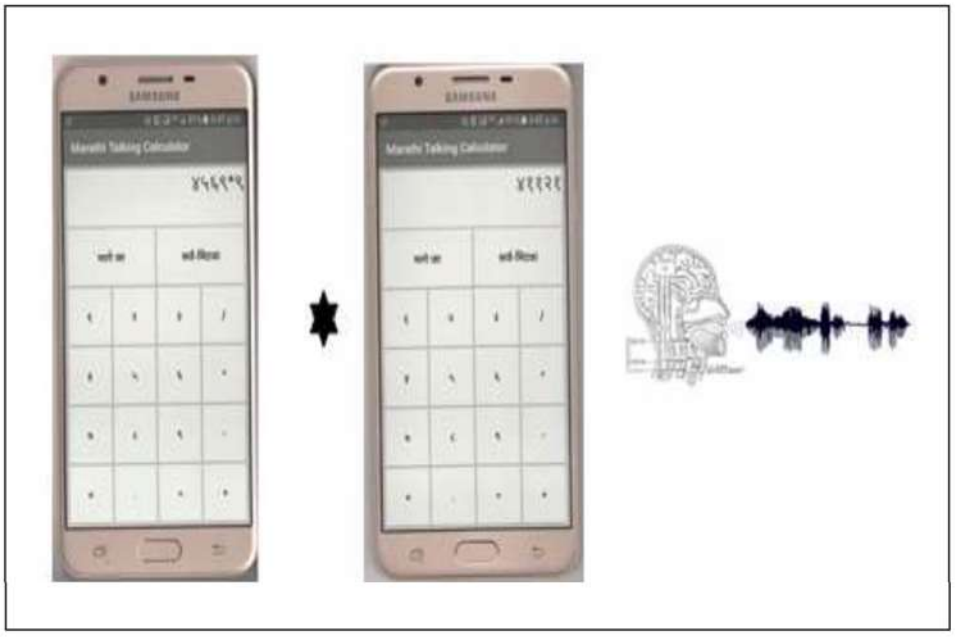


Figure 4.9: Android Based Marathi Speech Talking Calculator Application

and then the action was taken towards the command in the calculator. The figure 4.10 presents Demo database values for the Android Marathi Speech Talking Calculator the basic structure.

4.5.3 Performance Evaluation MOS on Android based Marathi Speech Talking Calculator

MOS is calculated for subjective quality measurement. It is calculated for the synthesized speech using the Unit selection synthesis. It was counselled to the listeners that they have to score between 01 to 05 (Excellent 05 Very good 04 Good 03 Satisfactory 02 Not understandable-01) for understandable. The table 4.10 shows the unit selection synthesis given by each subject.

4.5.4 The quality of speech as per the above MOS table for this is as follows:

Graphical Representation: Percentage Evaluation for Synthesized output speech for Android application

The Graph 4.11 and table 4.11 shows that the 95% of individuals rated the quality of synthetic speech is good and understandable. The only 4.5% speech is perceptible and

Table 4.9: Parameters used for the database creation

Sr. No	Parameter	Value
1	Parameter	Value
2	Sampling Rate	16000
3	Speakers	Dependent
4	Condition of Noise	Normal
5	Accent	Marathi
6	Pre emphasis	1-0.9/(z-1)
7	Window type	Hamming, 25 milliseconds
8	Window step size	20 millisecond

Table 4.10: Synthesis of the scores given by subject

Sr.No	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6	Sub7	Sub8	Sub9	Sub10	Average
1	5	5	5	5	4	4	5	4	5	5	4.7
2	5	5	5	5	4	4	5	4	5	5	4.7
3	5	5	5	5	4	4	5	4	5	5	4.7
4	5	5	5	5	4	4	5	4	5	5	4.7
5	5	5	5	5	4	4	5	4	5	5	4.7
6	5	5	5	5	4	4	5	4	5	5	4.7
7	5	5	5	5	4	4	5	4	5	5	4.7
8	5	5	5	5	4	4	5	4	5	5	4.7
9	5	5	5	5	4	4	5	4	5	5	4.7
10	5	4	5	5	5	4	4	5	4	5	4.7
Average											4.7

not clearly produced. Thus Unit selection method provides the naturalness and understand ability, the two important parameters of TTS system.

4.6 Summary

In earlier chapter 3, the description of tools like festival and festvox are explained. In this chapter we have described the android platform and the experimental analysis for TTS evulataion. The android app is built for Marathi calculator. It performs the basic operations i.e. addition, subtraction, multiplication and division. It reads out every number and input and output in Marathi language. The home screen is built up in Marathi Devanagari Font. MOS test is performed for the built up app which gives the result as 95

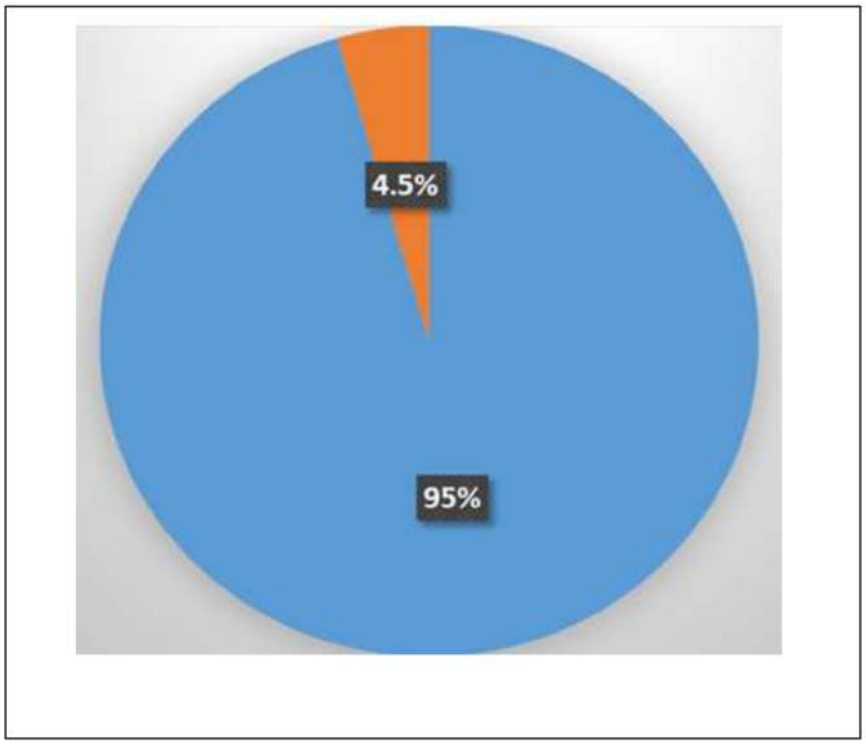
Sr.No	Numb1	Opr1	Numb2	Opr2	Result	Time in Secand	Result in Speech
१	२	+	६	=	८	१	आठ
२	५	+	९	=	१४	१.५	चौदा
३	५९	+	८९	=	१४८	२.६	एकशे अठ्ठेचाळीस
४	६६	+	८४	=	१५०	२.६	एकशे पन्नास
५	८७	+	९३	=	१८०	२.६	एकशेऐंशी
६	९६८	+	७६९	=	१८३७	३.७	एकहजार आठशे सदतीस
७	२३६७	+	९५६३	=	११९३०	४.७	अकराहजार नऊशे तीस
८	५८६	+	३२१	=	९०७	२.६	नऊशे सात
९	५८२	+	६३९	=	१२२१	३.७	एकहजार दोनशे एकवीस
१०	४९३	+	३५७	=	८५०	१.६	आठशे पन्नास
११	१२.१५	+	९५.६९	=	१०७.८४	५	एकशे सात पुर्णका चौऱ्याऐंशी
१२	७५.९	+	५६.७८	=	१३२.६८	५	एकशे बत्तीस पुर्णका अडुसष्ठ
१३	१०	-	६	=	४	१	चार
१४	१५	-	९	=	४	१	चार

Figure 4.10: Result for the Android Marathi Speech Talking Calculator

Table 4.11: Understandable voice quality

High quality and Understandable	94.5%
Perceptible Quality speech	4.5%

Figure 4.11: Result for the Android Marathi Speech Talking Calculator



5 Conclusion

The development of speech synthesis system has been started over few decades. Synthesized speech is generated through the conversion of the written text to speech form. The present work elaborated various method for the synthesis of speech. The articulatory based speech synthesis is complex with high computational load whereas with formant synthesis the quality of synthetic speech is good, but the speech sounds unnatural. Concatenation is one of the most widely used technique of speech synthesis. The Unit selection and domain based subcategories of concatenation are implemented for the proposed study. The benefit of concatenate method is that it provides more natural and individual sounding speech, but the quality with some consonants may vary. In case of longer units the controlling of pitch and duration may be difficult. However, using higher sample rate than necessary, the speech may sound slightly more pleasant. Artificial Neural Networks and Hidden Markov Models speech synthesis have been found promising for controlling the synthesizer parameters, such as gain, duration, and fundamental frequency. The hybrid approach of these basic methods has been used by various researchers. Combining the best properties of the basic methods is a good idea, but practically it will be very difficult to control the synthesis process.

Language is an important part in speech synthesis. Through the Literature review it is observed that much of efforts are already been taken in International and National languages. It is also seen that much of speech technology usually comes with English as it is a global language. Nationally good work is contributed by various researchers in Punjabi, Tamil, Telugu, Bangla, Hindi, Marathi and many more. Academic Institutions like IIIT Hyderabad, CDAC, IIT Mumbai, IIT Madras and many speech research laboratories are continuously working for the development of the speech synthesis. To make the technology reach to common mass it has to be language specific. The proposed research is carried for Marathi Speech synthesis.

Speech synthesis technology is been incorporated into various applications. For most applications, the intelligibility and understandability of synthetic speech have reached the

acceptable level. However, in prosodic, text preprocessing, and pronunciation fields there is still much work and improvements to be done to achieve more natural sounding speech. Natural speech has so many dynamic changes that perfect naturalness may be impossible to achieve. As the application development in speech synthesis are increasing steadily, the efforts into this research area is also increasing.

The present study explores the speech synthesis by understanding the very important tool named FESTIVAL. It is a general prepackaged tool for development of multi-language speech synthesis systems and it supports most of the languages in the text to speech conversion. The voice model is generated by using Festvox framework, festival and speech tools.

The research work starts with the experimentation conducted for analyzing the implementation of Festival for Marathi language with unit selection module of Festival. The database is created with the proper technical specification and is mentioned in chapter 3. The objective of this experiment was to check the detail process of the synthesis as well as the analysis of quality, naturalness and understandability of synthesized voice. For the evaluation of synthesized speech signal the Peak signal to noise ratio is calculated. It is beneficial to use normal speech processing techniques with synthesized speech for evaluating the naturalness parameters. Mel Frequency cepstral coefficients were also calculated for original and synthesized voice. It is observed that:

- The MSE and PSNR ratio for unit selection method are 93.83% and 96.23% respectively which shows that the output speech is very near to normal one.

- The MFCC coefficients also shows that the synthesized speech is sounding to naturalness.

It is observed that controlling prosodic features is very difficult and the synthesized speech still sounds usually synthetic or monotonic. The methods for correct pronunciation have been developed steadily during last decades and the present systems are quite good, but improvements with especially proper names are needed. Text preprocessing with numbers and some context-dependent abbreviations is still very problematic. However, the development of semantic parsing or text understanding techniques may provide a major improvement in high-level speech synthesis.

The application of TTS system can be used to read text from emails, SMSs, web pages, news, articles, blogs, talking books and toys, games, man machine communications etc. Internet revolution made phones smart which became an integral part of life. There

are number of speech driven applications available on smart phones. Siri from iPhone is an example that activates calling, sending texts, emails, web browsing etc. Android operating system is gaining lots of attention as it provides access to various features of the phone like location sensor, TTS and many more. Android being open source provides free development tools, which motivated people to use android system. The key features of Android are:

- It enables free download of development environment for the development of an application.
- Contains rich database of software libraries
- High Quality audiovisual contents
- Ability to test applications on most computing platforms, including Windows and Linux.

There are various speech synthesis application available in Marathi for specific domain like agriculture but this android application in Marathi is designed and implemented for calculation purpose. The developed application will be beneficial to illiterate mass of Maharashtra for calculation. The application is available in play store as Marathi text to speech calculator. The Third experimentation performed was Computer based Marathi talking calculator which is designed with Matlab.

Only creation of the application would have not been enough, the evaluation and assessment play one of the most important roles. Different evaluation methods have been discussed in the previous chapter. Before performing a listening test, the method used should be tested with smaller listener group to find out possible problems and the subjects should be chosen carefully. It is also impossible to say which test method provides the valid data and it is perhaps reasonable to use more than one test. Depending upon the application we have used the Mean Opinion Score method for all the experiments. The MOS score for the experimental analysis is high quality and Understandable speech is 85% and Perceptible Quality speech is 15% for Matlab based Marathi talking calculator. Also for high quality and understandable speech is 95% and Perceptible Quality speech is 4.5% in Android based Marathi talking calculator which found to be an efficient score to rate the synthesized speech natural and understandable.

The experimentation resulted in following observations:

- The MOS score for the experimental analysis is high quality and Understandable speech is 85% and Perceptible Quality speech is 15% for MATLAB based Marathi talking calculator.

- The MOS score for Android based Marathi talking calculator is 95% for high quality and understandable speech and Perceptible Quality speech is 4.5%

- High Quality audiovisual contents

- Ability to test applications on most computing platforms, including Windows and Linux.

5.1 Salient Features

- This research work attempts to work for speech synthesis for Marathi in which still efforts are needed.

- The study reveals the details regarding the Marathi speech synthesis using Festival which will help the others researchers to contribute in the same domain.

- The database created can be utilized for developing applications for different purposes.

- The designed Android based Marathi speech calculator will assist the local mass of Maharashtra to utilize it in daily routine. The application is available in Google Play store.

- As computer systems are now becoming language friendly the computer based application will be useful in government offices for the calculation purposes in Marathi language.

5.2 Salient Features

- The work is limited for specific application of calculation for ten digits only.

- It works only for basic arithmetic operations like addition, Subtraction, Division and Multiplication.

5.3 Future Works

It is quite clear that there is still very long way to go before text-to-speech synthesis, is fully acceptable. However, the development is going forward steadily and in the long run the technology seems to make progress faster than we can imagine. The Multi language applications can be designed. Many useful assertive devices can be developed in regional languages. Even interpretation from a language to another may became feasible which will help to learn the language easily.

6 List of Publication

1. **Sangramsing Kayte**, Kavita Waghmare, Dr. Bharti Gawali Marathi Speech Synthesis: A review International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169 Volume: 3 Issue: 6 3708 3711 (Impact Factor 5.83

2. **Sangramsing Kayte,** Bharti Gawali "A Text-To-Speech Synthesis for Marathi Language using Festival and Festvox" International Journal of Computer Applications (0975 8887) Volume 132 No.3, December 2015

3. **Sangramsing Kayte**, Kavita Waghmare and Bharti Gawali "Analysis of Pitch and Duration in Speech Synthesis using PSOLA. Communications on Applied Electronics 4(4):10-18, February 2016. Published by Foundation of Computer Science (FCS), NY, and USA.

4. **Sangramsing N Kayte**, Charansing N Kayte and Bharti W Gawali. Article: Marathi Synthetic Voice using Synthesizer Modules of Festival Speech and HTS Straight Processing. Communications on Applied Electronics 3(7):9-12, December 2015. Published by Foundation of Computer Science (FCS), NY, USA

5. **Sangramsing Kayte** Marathi Speech Recognition System Using Hidden Markov Model Toolkit International OPEN ACCESS Journal Of Modern Engineering Research (IJMER), ISSN: 22496645, Vol. 5, Iss. 12, December 2015.

6. **Sangramsing Nathusing Kayte** Festival and Festvox Framework Tools for Marathi Text-to-Speech Synthesis International Journal of Computer Applications 132(4):38-43, December 2015. Published by Foundation of Computer Science (FCS), NY, USA

7. **Sangramsing Nathusing Kayte** "Text To Speech for Marathi Language using Transcriptions Theory. International Journal of Computer Applications 131(6):39-41, December 2015. Published by Foundation of Computer Science (FCS), NY, USA

8. **Sangramsing Kayte**, Bharti Gawali "A Text-To-Speech Synthesis for Marathi Language using Festival and Festvox" International Journal of Computer Applications (09758887)Volume-132, No.3, December2015

9. **Sangramsing N. Kayte**, Dr.Bharti Gawali "The Marathi Text-To-Speech Synthesizer Based On Artificial Neural Networks " International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 02 Issue: 08, Nov-2015-ISSN: 2395-0072

6.1 CHAPTERS IN THE BOOK

1. **Sangramsing N. Kayte**, Monica Mundada, Santosh Gaikwad and Bharti Gawali "Performance Evaluation of Speech Synthesis Techniques for English Language" Springer Science Business Media Singapore 2016 S.C. Satapathy et al. (eds.), Proceedings of the International Congress on Information and Communication Technology, Advances in Intelligent Systems and Computing 439, DOI 10.1007/978-981-10-0755-2_27

References

[1] S. Kayte and B. Gawali, "Marathi speech synthesis: A review," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 3, no. 6, pp. 3708–3711, 2015.

[2] D. Sasirekha and E. Chandra, "Text to speech: a simple tutorial," *International Journal of Soft Computing and Engineering (IJSCE)*, vol. 2, no. 1, pp. 275–278, 2012.

[3] D. H. Klatt, "Review of text-to-speech conversion for english," *The Journal of the Acoustical Society of America*, vol. 82, no. 3, pp. 737–793, 1987.

[4] P. Chaudhury, M. Rao, and K. Kumar, "Symbol based concatenation approach for text to speech system for hindi using vowel classification technique," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*. IEEE, 2009, pp. 1082–1087.

[5] P. Taylor, *Text-to-speech synthesis*. Cambridge university press, 2009.

[6] J. Devaney and C. Goodyear, "A comparison of acoustic and magnetic resonance imaging techniques in the estimation of vocal tract area functions," in *Proceedings of ICSIPNN'94. International Conference on Speech, Image Processing and Neural Networks*. IEEE, 1994, pp. 575–578.

[7] A. Greenwood and C. Goodyear, "Articulatory speech synthesis using a parametric model and a polynomial mapping technique," in *Proceedings of ICSIPNN'94. International Conference on Speech, Image Processing and Neural Networks*. IEEE, 1994, pp. 595–598.

[8] S. Parthasarathy and C. Coker, "Phoneme-level parameterization of speech using an articulatory model," in *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1990, pp. 337–340.

[9] P. Vary, U. Heute, and W. Hess, "Digitale sprachsignalverar beitung (teubner, stuttgart)," 1998.

[10] J. Beskow, "Talking heads-communication, articulation and animation," in *proceedings of Fonetik*, vol. 96, 1996, pp. 29–31.

[11] N. S. Krishna, H. A. Murthy, and T. A. Gonsalves, "Text-to-speech (tts) in indian languages," in *Int. conf. natural language processing*, 2002, pp. 317–326.

[12] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1. IEEE, 1996, pp. 373–376.

[13] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *the Journal of the Acoustical Society of America*, vol. 67, no. 3, pp. 971–995, 1980.

[14] D. H. Klatt and L. C. Klatt, "Analysis, synthesis, and perception of voice quality variations among female and male talkers," *the Journal of the Acoustical Society of America*, vol. 87, no. 2, pp. 820–857, 1990.

[15] I. Dunđer, S. Seljan, and M. Arambašić, "Domain-specific evaluation of croatian speech synthesis in call," *Recent Advances in Information Science-Recent Advances in Computer Engineering Series, WSEAS*, vol. 1, p. 142, 2013.

[16] D. Ravi and S. Patilkulkarni, "A novel approach to develop speech database for kannada text-to-speech system," *Int. J. on Recent Trends in Engineering & Technology*, vol. 5, no. 01, 2011.

[17] H. Hon, A. Acero, X. Huang, J. Liu, and M. Plumpe, "Automatic generation of synthesis units for trainable text-to-speech systems," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, vol. 1. IEEE, 1998, pp. 293–296.

[18] F.-C. Chou, C.-Y. Tseng, and L.-S. Lee, "A set of corpus-based text-to-speech synthesis technologies for mandarin chinese," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 7, pp. 481–494, 2002.

[19] M. Hasegawa-Johnson, "Lecture notes in speech production, speech coding, and speech recognition," *class notes, University of Illinois at Urbana-Champaign, Fall*, 2000.

[20] T. Dutoit, *An introduction to text-to-speech synthesis*. Springer Science & Business Media, 1997, vol. 3.

[21] R. E. Donovan, "A new distance measure for costing spectral discontinuities in concatenative speech synthesizers," in *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*, 2001.

[22] R. E. Donovan and E. M. Eide, "The ibm trainable speech synthesis system," in *Fifth International Conference on Spoken Language Processing*, 1998.

[23] D. Pennell and Y. Liu, "Toward text message normalization: Modeling abbreviation generation," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5364–5367.

[24] H. A. Murthy, "(a) personal," 1985.

[25] A. Katsamanis, M. Black, P. G. Georgiou, L. Goldstein, and S. Narayanan, "Sailalign: Robust long speech-text alignment," in *Proc. of Workshop on New Tools and Methods for Very-Large Scale Phonetics Research*, 2011.

[26] M. Macon, A. Cronk, J. Wouters, and A. Kain, "Ogireslpc: Diphone synthesizer using residual-excited linear prediction," 1997.

[27] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *speech communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[28] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The hmm-based speech synthesis system (hts) version 2.0." in *SSW*. Citeseer, 2007, pp. 294–299.

[29] R. Sproat, A. W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards, "Normalization of non-standard words," *Computer speech & language*, vol. 15, no. 3, pp. 287–333, 2001.

[30] S. Hailemariam and K. Prahallad, "Extraction of linguistic information with the aid of acoustic data to build speech systems," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4. IEEE, 2007, pp. IV–717.

[31] P. Lavanya, P. Kishore, and G. T. Madhavi, "A simple approach for building transliteration editors for indian languages," *Journal of Zhejiang University-SCIENCE A*, vol. 6, no. 11, pp. 1354–1361, 2005.

[32] H. Garg, "Overcoming the font and script barriers among indian languages," Ph.D. dissertation, MS dissertation, International Institute of Information Technology , 2004.

[33] G. Madhavi, B. Mini, N. Balakrishnan, and R. Raj, "Om: One tool for many (indian) languages," *Journal of Zhejiang University-SCIENCE A*, vol. 6, no. 11, pp. 1348–1353, 2005.

[34] M. Choudhury, "Rule-based grapheme to phoneme mapping for hindi speech synthesis," in *90th Indian Science Congress of the International Speech Communication Association (ISCA), Bangalore, India*, 2003.

[35] E. P. P. Soe and A. Thida, "Diphone-concatenation speech synthesis for myanmar language," *International Journal of Science, Engineering and Technology Research (IJSETR)*, vol. 2, no. 5, 2013.

[36] J. Bloch, *Formation of the Marathi Language*. Motilal Banarsidass Publ., 1970.

[37] M. Berntsen and J. Nimbkar, "A marathi reference grammar." 1975.

[38] R. V. Dhongde and K. Wali, *Marathi*. John Benjamins Publishing, 2009, vol. 13.

[39] N. Narendra, K. S. Rao, K. Ghosh, R. R. Vempada, and S. Maity, "Development of syllable-based text to speech synthesis system in bengali," *International journal of speech technology*, vol. 14, no. 3, p. 167, 2011.

[40] K. Bali, P. P. Talukdar, N. S. Krishna, and A. Ramakrishnan, "Tools for the development of a hindi speech synthesis system," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[41] H. A. Murthy, A. Bellur, V. Viswanath, B. Narayanan, A. Susan, G. Kasthuri, and K. Prahallad, "Building unit selection speech synthesis in indian languages: An initiative by an indian consortium," *Proceedings of COCOSDA, Kathmandu, Nepal*, 2010.

[42] S. Kishore, A. W. Black, R. Kumar, and R. Sangal, "Experiments with unit selection speech databases for indian languages," *Carnegie Mellon University*, 2003.

[43] S. P. Kishore and A. W. Black, "Unit size in unit selection speech synthesis," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[44] T. Shreekanth, V. Udayashankara, and C. A. Kumar, "An unit selection based hindi text to speech synthesis system using syllable as a basic unit," *IOSR Journal Of VLSI And Signal Processing (IOSR-JVSP)*, vol. 4, no. 4, pp. 49–57, 2014.

[45] M. Vinodh, A. Bellur, K. B. Narayan, D. M. Thakare, A. Susan, N. Suthakar, and H. A. Murthy, "Using polysyllabic units for text to speech synthesis in indian languages," in *2010 National Conference on Communications (NCC)*. IEEE, 2010, pp. 1–5.

[46] B. Sudhakar and R. Bensraj, "Development of concatenative syllable based text to speech synthesis system for tamil," *International Journal of Computer Applications*, vol. 91, no. 5, 2014.

[47] S. Jothilakshmi, S. Sindhuja, and V. Ramilingam, "Dravidian-tamil tts for interactive voice response system," *International Journal of Innovative Research and Development (ISSN 2278–0211)*, vol. 2, no. 4, pp. 725–736, 2013.

[48] P. Deshpande and J. Chitode, "Transformation coding for emotion speech translation: a review," *International Journal of Electrical and Electronics Engineering Research (IJEEER)*, vol. 6, no. 1, pp. 1–12, 2016.

[49] J. Kothari and C. Kumbharana, "A phonetic study for constructing a database of gujarati characters for speech synthesis of gujarati text," *International Journal of Computer Applications*, vol. 117, no. 19, 2015.

[50] S. Kishore, R. Sangal, and M. Srinivas, "Building hindi and telugu voices using festvox," *Proceedings of ICON*, 2002.

[51] S. Kishore, R. Kumar, and R. Sangal, "A data driven synthesis approach for indian languages using syllable as basic unit," in *Proceedings of Intl. Conf. on NLP (ICON)*, 2002, pp. 311–316.

[52] V. Peddinti and K. Prahallad, "Significance of vowel epenthesis in telugu text-to-speech synthesis," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5348–5351.

[53] M. S. Kumar, P. Prabhu, M. S. Reddy, and P. Kumar, "Text to speech system for telugu language," *International Journal of Engineering Research and Applications*, vol. 4, no. 3, pp. 913–917, 2014.

[54] G. K. Anumanchipalli, K. Prahallad, and A. W. Black, "Festvox: Tools for creation and analyses of large speech corpora," in *Workshop on Very Large Scale Phonetics Research, UPenn, Philadelphia*, 2011, p. 70.

[55] B. Mahananda, C. Raju, R. R. Patil, N. Jha, S. Varakhedi, and P. Kishore, "Building a prototype text to speech for sanskrit," in *International Sanskrit Computational Linguistics Symposium*. Springer, 2010, pp. 39–47.

[56] P. Singh and G. S. Lehal, "Syllables selection for the development of speech database for punjabi tts system," *International Journal of Computer Science Issues (IJCSI)*, vol. 7, no. 6, p. 164, 2010.

[57] S. Mukherjee and S. K. D. Mandal, "A bengali speech synthesizer on android os," in *Proceedings of the 1st Workshop on Speech and Multimodal Interaction in Assistive Environments*. Association for Computational Linguistics, 2012, pp. 43–46.

[58] M. M. Alam, M. N. Uddin, and M. R. I. Chy, "Text to speech system for bengali language using characters based transliterations," *International Journal of Computer Applications*, vol. 108, no. 3, 2014.

[59] F. Alam, P. K. Nath, and M. Khan, "Text to speech for bangla language using festival," 2007.

[60] F. Alam, S. M. Habib, and M. Khan, "Bangla text to speech using festival," in *Conference on Human Language Technology for Development*, 2011, pp. 154–161.

[61] A. A. Shah, A. W. Ansari, and L. Das, "Bi-lingual text to speech synthesis system for urdu and sindhi," in *National Conf. on Emerging Technologies*, 2004, pp. 20 126–130.

[62] A. Soman, S. S. Kumar, V. Hemanth, M. S. Manikandan, and K. Soman, "Corpus driven malayalam text-to-speech synthesis for interactive voice response system," in *International Journal of Computer Applications*. Citeseer, 2011, vol. 29, no. 4, pp. 0975–8887.

[63] S. S. Nair, C. Rechitha, and C. S. Kumar, "Rule-based grapheme to phoneme converter for malayalam," *International Journal of Computational Linguistics and Natural Language Processing*, vol. 2, no. 7, pp. 417–420, 2013.

[64] S. Shirbahadurkar and D. Bormane, "Speech synthesizer using concatenative synthesis strategy for marathi language (spoken in maharashtra, india)," *International journal of recent trends in engineering*, vol. 2, no. 4, p. 80, 2009.

[65] E. V. Raghavendra, S. Desai, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Global syllable set for building speech synthesis in indian languages," in *2008 IEEE Spoken Language Technology Workshop*. IEEE, 2008, pp. 49–52.

[66] R. Kumar and S. P. Kishore, "Automatic pruning of unit selection speech databases for synthesis without loss of naturalness," in *Eighth International Conference on Spoken Language Processing*, 2004.

[67] E. V. Raghavendra, P. Vijayaditya, and K. Prahallad, "Speech synthesis using artificial neural networks," in *2010 National Conference On Communications (NCC)*. IEEE, 2010, pp. 1–5.

[68] K. Prahallad, E. N. Kumar, V. Keri, S. Rajendran, and A. W. Black, "The iiit-h indic speech databases," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[69] K. Prahallad and A. W. Black, "Handling large audio files in audio books for building synthetic voices," in *Seventh ISCA Workshop on Speech Synthesis*, 2010.

[70] S. H. Mariam, S. P. Kishore, A. W. Black, R. Kumar, and R. Sangal, "Unit selection voice for amharic using festvox," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[71] S. Kayte, M. Mundada, and D. C. Kayte, "Screen readers for linux and windows–concatenation methods and unit selection based marathi text to speech system," *International Journal of Computer Applications*, vol. 975, p. 8887, 2015.

[72] M. Kaszczuk and L. Osowski, "The ivo software blizzard challenge 2009 entry: Improving ivona text-to-speech," in *Blizzard Challenge Workshop*. Citeseer, 2009.

[73] A. Black and K. Lenzo, "Building voices in the festival speech synthesis system," 2000.

[74] A. Black, P. Taylor, R. Caley, and R. Clark, "The festival speech synthesis system," 1998.

[75] M. Karjalainen, "Review of speech synthesis technology," *Helsinki University of Technology, Department of Electrical and Communications Engineering*, 1999.

[76] A. W. Black and P. A. Taylor, "Automatically clustering similar units for unit selection in speech synthesis." 1997.

[77] S. Kayte, M. Mundada, and D. C. Kayte, "Speech synthesis system for marathi accent using festvox," *International Journal of Computer Applications*, vol. 130, no. 6, pp. 38–42, 2015.

[78] K. S. R. Murty, B. Yegnanarayana, and M. A. Joseph, "Characterization of glottal activity from speech signals," *IEEE signal processing letters*, vol. 16, no. 6, pp. 469–472, 2009.

[79] M. Maier, N. Ghazisaidi, and M. Reisslein, "The audacity of fiber-wireless (fiwi) networks," in *International Conference on Access Networks*. Springer, 2008, pp. 16–35.