1

INTRODUCTION TO RECOGNITION OF TEXT PATTERN FOR TEXT TO SPEECH CONVERSION

* Introduction
* Statement of the problem
* Objective of the study
* Rational of the study
* Scope of the study
* Limitations of the study
* Literature review for the study
* Some generic TTS Frameworks
* Speech synthesis mark-up languages: An overview
* Short summary of some speech Engine
* Linguistic studies in India
* Salient features of the present study
* Glossary
* Organization of the Thesis
* References
CHAPTER – 1
INTRODUCTION TO RECOGNITION OF TEXT PATTERN FOR TEXT TO SPEECH CONVERSION

1.1 Introduction

Human being started writing for the purpose of storing information, communicate and exchange their thoughts. Writing will last for long while speech will not. In starting period of human evaluation, they used figures to describe their feelings, fears and facts. Egyptians used papyrus to write down their ideas and the Chinese, in 11th Century, went beyond handwriting using typography with moveable type to create multiple copies of their documents. Through centuries typography has evolved and nowadays is one of the major manners to exchange ideas and information.

Using computers, typography and the way we write has changed radically. Entering and storing text become easy (using an editor like NOTEPAD). Text can be formatted in view to font and paragraph (using software like Open Office). Utilizing these tools, computer users have the ability not only to write a plain text, but also to format the text and arrange it in the page. In newspapers, for example, the page designer distinguishes the title from the body of text at the top of the page or the article, with larger font size. Also, when an editor wishes to emphasize a specific word or phrase he uses bold or italics typesetting. A writer can convey a message, a felling or an idea not only by the meaning of the content but also by the way the text is visually presented to the reader. The use of the WWW and the web page creation and design, introduced a new perception for the meaning of documents and publishing. In web pages, the text and background combinations have impact on the readability and aesthetics and a well designed graphical web document can be reader friendly. So, storing, formatting, editing, printing and communicating with text become easy due to evolution in technology use of computers.

But on other hand these types of utilities have no much more importance for the peoples with various types of disabilities. There are many types of disabilities like blindness and other reading difficulties as well as by pre-literate children. They will be benefited if they can get information in form of speech instead of text. Speech synthesis has long been a vital assistive technology tool and its application in this area is significant and widespread. The longest application has been in the use of screen
readers for people with visual impairment, but text-to-speech systems are now commonly used by people with dyslexia also. Text to Speech is also finding new applications outside the disability market. For example, speech synthesis, combined with speech recognition, allows for interaction with mobile devices via natural language processing interfaces. Speech synthesis is also used for facilitating the creation of online presentations.

1.2 Statement of the problem

**Title of the present study was:** “Recognition of Text Pattern for Text To Speech Conversion”

In present study speech pattern recognition is pertaining to Gujarati language.

Gujarati language consists different 34+10+12 alphabets including ‘k (iants)’ to ‘gna (dris)’, ‘1(७)’ to ‘9(९)’ and ‘०’ to ‘२०’. The prime theme of the study is to identify or recognize the patterns of the text stored into Gujarati font and convert it into speech.

When user enters Gujarati text its alphabetic pattern is identifies and it will be compared with the corresponding alphabetic pattern stored in the standard phoneme databases, corresponding series of phoneme is used to concatenate and produce in form of speech.

1.3 Objective of the study

The principle objective of the present study is

Develop a module for recognition of text pattern for text to speech conversion.

To full fill this objectives some sub objectives were formed which are as following:

1.3.1 Developing software for recognition of text pattern for text to speech conversion.

1.3.2 Identification of a new Phoneme based speech unit suitable for concatenative speech synthesis.

1.3.3 Designing and development of an interactive user-friendly text editor, which allows the user to enter the text, manipulate text, select text.

1.3.4 Designing and development of Gujarati fonts.

1.3.5 Creation of Gujarati phoneme database.

1.3.6 Creation of wave files for speech dictation.
1.3.7 Development of a model that will compare the text data with phoneme database and speak it using speakers.
1.3.8 Exploring promising new ideas in text recognition.
1.3.9 Incorporating emotions in output speech.
1.3.10 Input text can be entered text, a document or e-mail.
1.3.11 Output can be obtained in different audio format like wav and mp3.
1.3.12 Developing advanced technology incorporating these ideas.
1.3.13 Performance testing of the application developed in new technology.

1.4 Rational of the study

Importance of the study can be identified as:
1.4.1 Text-to-speech helps those who cannot read and understand its use and benefits for a variety of reasons.
1.4.2 Individuals with severe learning disabilities such as dyslexia, who have difficulty decoding and understanding text may comprehend printed content much better when it is read out loud.
1.4.3 Individuals with physical disabilities who cannot hold a book or turn its pages may be benefited text-to-speech output. The appropriate assistive technologies will provide these students with the accommodations, including text-to-speech, as needed to read independently.
1.4.4 A student who has difficulty decoding multi syllabic words, loses his or her place on the page, or has difficulty comprehending printed text, may benefit from text-to-speech output.
1.4.5 Speech can add a new dimension to computer-based learning with speech output added to an on-screen representation of the word.
1.4.6 Synthesized voices are wonderful for checking your written work. It’s much easier to hear the mistakes than see them. As well as spelling mistakes, speech output also helps spot the right word in the wrong place, missing or duplicate words.
1.4.7 Text to speech can improve the end-users
   - Word recognition skills and vocabulary.
   - Reading comprehension, fluency, accuracy, and concentration.
   - Information recall and learning/memory enhancement.
1.4.8 There are an awful lot of situations where text to speech facility is useful.

- People with Gujarati as a second language
- People with poorer literacy
- People with dyslexia
- People with poor vision
- Students who are struggling with reading

1.4.9 Other benefits of Text to Speech facility are helping with proofreading and reducing eyestrain.

1.4.10 Using this facility one can read anywhere; he or she may be in traveling or in garden or somewhere else.

1.4.11 User can up or down reading speed while using text to speech facility.

1.4.12 Text-to-speech systems are useful for the visually impaired, and for situations where users are not able to view the computer screen at all times, such as when they are driving.

1.4.13 Voice handicaps originate in mental or motor/sensation disorders. Machines can be an invaluable support in the latter case: with the help of an especially designed keyboard and a fast sentence assembling program, synthetic speech can be produced in a few seconds to remedy these impediments. Astrophysicist Stephen Hawking gives all his lectures in this way.

1.4.14 Used in Interactive voice Response (IVR) systems, e.g. banking applications.

1.5 Scope of the study

“Text recognition”, the word leads the readers mind in a lot many different paths, but here the study is concerned only with text pattern recognition, for conversion of the inputted Gujarati text into equivalent speech. The study is distributed among the following different software components, which are developed using Microsoft Visual Studio 2010 (C #), SQL Database, NVidezor, Free Audio Editor 2015 etc.

The software mainly contains two parts:
1. Phoneme and Sound Data creator
2. Text to Speech Converter
Phoneme and Sound data Creator part contains Microphone button to take a phoneme. The aim of this part is to create a person’s sound database. It will display next Gujarati phoneme to speak, person will speech the phoneme, and it can be correct by removing silence, finalize the sound and store it in the database.

Text to Speech Converter part contains a text box to enter Gujarati text. The base theme of study is to enter Gujarati text in the text box and it will be speak out through speaker.

The software is specially designed for Gujarati type, so using the font creator software, the new \textit{ttf} Gujarati font is developed which uses the standard Gujarati keying of keyboard. The editor also supports the other related Gujarati font like, TeraFont which are installed in the computer system.

For the character comparison a phoneme database with different characteristic is created as a standard sample.

The user can listen directly the converted speech or record the converted text into .wav or .mp3 format file.

1.6 Limitation of the study

1.6.1 Only C (Consonant), V (Vowel), CV and VC type phoneme is taken. CVV or VCV type phonemes are not considered.
1.6.2 Output speech is found only in wav format. Conversion in mp3 format is not obtained.
1.6.3 Diaphone and triphone not considered.
1.6.4 Only one person’s voice is recorded. So, selection in voice is not incorporated.
1.6.5 Emotions are not included.
1.6.6 Only text entered in prescribed textbox is converted into speech.

1.7 Literature review for the study

In the XXI century widespread use of computers opened a new stage in information interchange between the user and the computer. Among other things, an opportunity to input the information to computer through speech and to reproduce in voice text information stored in computer has been made possible. The main objective of the present study was: “To develop software for text to speech conversion”. The researcher had studied a lot of literature to identify the different works done in the
related area. Study had given a more clear vision in the task. In the present chapter, the brief information about related research works for the present study is given.

1.7.1 A glance over related literature

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware. A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diaphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely synthetic voice output.

Different organization and researchers define the term “speech synthesis”, and it can be summarized as follow:

- Speech synthesis is the computer-generated simulation of human speech. It is used to translate written information into aural information.
- Speech synthesis is the process of generating spoken language by machine on the basis of written input [1].
- Speech synthesis is Computer technology that 'constructs' human speech from electronic circuits to replace pre-recorded human voice [2]. Its major applications are in assistive technology for helping blind hear the written word, and in telephone answering devices such as automated attendants.
- Speech synthesis is the generation of a sound waveform of human speech from textual or phonetic description [3].
- Speech synthesis is the artificial production of human speech and a Text-to-speech (TTS) system converts written text (language) into speech. A computer system used for this purpose is called a speech synthesizer, and can be implemented in software or hardware products.
1.7.2 Some empirical studies

Language is a fundamental part of everyday life. Whether we are using speech, sign language, or perhaps a coding system that conveys meaning through touch, we use language to express our thoughts, intentions, reactions, and experiences -- often scarcely considering that we are even speaking, but rather feeling as if we were engaging in direct manipulation of the notions with our conversation partner.

The complexity of the underlying systems involved in speech rarely enters our minds as we go about our business, and it seems as if this fosters an unspoken attitude: "It's easy to talk; it should be easy to make a machine that talk.". Human started efforts and experiments in direction to produce synthesis speech which involves lots of components of the process, the relative effort and expertise involved in each component, and where the trade-offs lie in speech synthesis today.

The creation of synthetic speech covers a whole range of processes, and though often they are all lumped under the general term text-to-speech, a good deal of work has gone into generating speech from sequences of speech sounds; this would be a speech-sound (phoneme) to audio waveform synthesis, rather than going all the way from text to phonemes (speech sounds), and then to sound.

Let’s have some remarkable landmarks in development of speech synthesis:

The earliest efforts to produce synthetic speech were made over two hundred years ago (Flanagan 1972, Flanagan et al. 1973, Schroeder 1993) [4]. In St. Petersburg 1779 Russian Professor Christian Kratzenstein explained physiological differences between five long vowels (/a/, /e/, /i/, /o/, and /u/) and made apparatus to produce them artificially [5]. He constructed acoustic resonators similar to the human vocal tract and activated the resonators with vibrating reeds like in music instruments. The basic structure of resonators is shown in Figure 1.1. The sound /i/ is produced by blowing into the lower pipe without a reed causing the flute-like sound.

![Figure 1.1: Kratzenstein's resonators (Schroeder 1993).](image)
A few years later, in Vienna 1791, Wolfgang von Kempelen introduced his "Acoustic-Mechanical Speech Machine", which was able to produce single sounds and some sound combinations (Klatt 1987, Schroeder 1993).

Figure 1.2: Von Kempelen's talking machine (1791)

In about mid 1800's Charles Wheatstone constructed his famous version of von Kempelen's speaking machine. It was a bit more complicated and was capable to produce vowels and most of the consonant sounds. Some sound combinations and even full words were also possible to produce.

In late 1800's Alexander Graham Bell with his father, inspired by Wheatstone's speaking machine, constructed same kind of speaking machine. Bell made also some questionable experiments with his terrier. He put his dog between his legs and made it growl, then he modified vocal tract by hands to produce speech-like sounds (Flanagan 1972, Schroeder 1993) [6][7].

Also around that time, Homer Dudley developed a mechanical device at Bell Laboratories that operated through the movement of pedals, and mechanical keys, like an organ. With a trained operator, it could be made to create sounds that, if given a good set-up, almost sounded like speech. Called the VODER (Voice Operating Demonstrator),
Figure 1.3: Wheatstone's construction of von Kempelen's speaking machine.

Figure 1.4: Architectural blueprint for the VODER

The VOCODER (Voice Coder), also developed by Homer Dudley, is one such example. Much of the work in synthesis in the 40s and 50s was primarily concerned with constructing replicas of the signal itself rather than generating the phones from an abstract form like text.

In 1951, Franklin Cooper and his associates developed a Pattern Playback synthesizer at the Haskins Laboratories (Klatt 1987, Flanagan et al. 1973). It reconverted recorded spectrogram patterns into sounds, either in original or modified form. The spectrogram patterns were recorded optically on the transparent belt.
The first formant synthesizer, PAT (Parametric Artificial Talker), was introduced by Walter Lawrence in 1953 (Klatt 1987). PAT consisted of three electronic formant resonators connected in parallel. The input signal was either a buzz or noise. A moving glass slide was used to convert painted patterns into six time functions to control the three formant frequencies, voicing amplitude, fundamental frequency, and noise amplitude.

![Figure 1.5: (a) The First Formant Synthesizer CAT and (b) The First Cascade Formant Synthesizer OVE I.](image)

At about the same time Gunnar Fant introduced the first cascade formant synthesizer OVE I (Orator Verbis Electris) which consisted of formant resonators connected in cascade. Ten years later, in 1962, Fant and Martony introduced an improved OVE II synthesizer, which consisted of separate parts to model the transfer function of the vocal tract for vowels, nasals, and obstruent consonants. Possible excitations were voicing, aspiration noise, and frication noise. The OVE projects were followed by OVE III and GLOVE at the Kungliga Tekniska Högskolan (KTH), Sweden, and the present commercial Infovox system is originally descended from these (Carlson et al. 1981, Barber et al. 1989, Karlsson et al. 1993)[8].

First articulatory synthesizer was introduced in 1958 by George Rosen at the Massachusetts Institute of Technology, M.I.T. (Klatt 1987). The DAVO (Dynamic Analog of the VOcal tract) was controlled by tape recording of control signals created by hand. In mid 1960s, first experiments with Linear Predictive Coding (LPC) were made (Schroeder 1993).[7] Linear prediction was first used in low-cost systems, such as TI Speak'n'Spell in 1980, and its quality was quite poor compared to present
systems. However, with some modifications to basic model the method has been found very useful and it is used in many present systems.

The first full text-to-speech system for English was developed in the Electrotechnical Laboratory, Japan 1968 by Noriko Umeda and his companions (Klatt 1987). It was based on an articulatory model and included a syntactic analysis module with sophisticated heuristics. The speech was quite intelligible but monotonous and far away from the quality of present systems.

With the rise of digital representations of speech, digital signal processing, and the proliferation of cheap, general-purpose computer hardware, more work was done in concatenation of natural recorded speech. Diphones appeared; that is, two adjacent half-phones (context-dependent phoneme realizations), cut in the middle, joined into one unit. The justification was that phone boundaries are much more dynamic than stable, interior parts of phones, and therefore mid-phone is a better place to concatenate units, as the stable points have, by definition, little rapid change, whereas there are rapid changes at the boundaries that depend upon the previous or next unit.

The rise of concatenative synthesis began in the 70s, and has largely become practical as large-scale electronic storage has become cheap and robust [9]. When a megabyte of memory was a significant part of researcher’s salary, less resource-intensive techniques were worth their... weight in saved cycles in gold, to use an odd metaphor. Of course formant, synthesis can still require significant computational power, even if it requires less storage; the 80s speech synthesis relied on specialized hardware to deal with the constraints of the time.

In 1972, the standard Unix manual (3rd edition) included commands to process text to speech, form text analysis, prosodic prediction, phoneme generation, and waveform synthesis through a specialized piece of hardware. Of course Unix had only about 16 installations at the time and most, perhaps even all, were located in Bell Labs at Murray Hill.

Techniques were developed to compress (code) speech in a way that it could be more easily used in applications. In late 1970's and early 1980's, considerably amount of commercial text-to-speech and speech synthesis products were introduced (Klatt 1987). 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 (Klatt 1987). The quality
was poor, by modern standards, but for the time it was very impressive. Speech was
basically encoded using LPC and mostly used isolated words and letters though there
were also a few phrases formed by concatenation. Simple text-to-speech (TTS)
engines based on specialized chips became popular on home computers such as the
BBC Micro in the UK and the Apple.

Dennis Klatt’s MITalk synthesizer [allen87] in many senses defined the
perception of automatic speech synthesis to the world at large. Later developed into
the product DECTalk, it produces somewhat robotic, but very understandable, speech.
It is a formant synthesizer, reflecting the state of the art at the time.

Before 1980, research in speech synthesis was limited to the large laboratories
that could afford to invest the time and money for hardware. By the mid-80s, more
labs and universities started to join in as the cost of the hardware dropped. By the late
eighties, purely software synthesizers became feasible; the speech quality was still
decidedly inhuman (and largely still is), but it could be generated in near real-time.

Of course, with faster machines and large disk space, people began to look to
improving synthesis by using larger, and more varied inventories for concatenative
speech. Yoshinori Sagisaka at Advanced Telecommunications Research (ATR) in
Japan developed nuu-talk [nuutalk92] in the late 80s and early 90s. It introduced a
much larger inventory of concatenative units; thus, instead of one example of each
diphone unit, there could be many, and an automatic, acoustically based distance
function was used to find the best selection of sub-word units from a fairly broad
database of general speech. This work was done in Japanese, which has a much
simpler phonetic structure than English, making it possible to get high quality with
relatively small databases. Even up through 1994, the time needed to generate of the
parameter files for a new voice in nuu-talk (503 sentences) was on the order of several
days of CPU time, and synthesis was not generally possible in real time.
With the demonstration of general unit selection synthesis in English in Rob Donovan's PhD work [donovan95], and ATR's CHATR system ([campbell96] and [hunt96]), by the end of the 90's, unit selection had become a hot topic in speech synthesis research. However, despite examples of it working excellently, generalized unit selection is known for producing very bad quality synthesis from time to time. As the optimal search and selection algorithms used are not 100% reliable, both high and low quality synthesis is produced -- and many difficulties still exists in turning general corpora into high-quality synthesizers as of this writing.

Modern speech synthesis technologies involve quite complicated and sophisticated methods and algorithms. One of the methods applied recently in speech synthesis is Hidden Markov Models (HMM). HMMs have been applied to speech recognition from late 1970's. For speech synthesis systems it has been used for about two decades. A hidden Markov model is a collection of states connected by transitions with two sets of probabilities in each: a transition probability which provides the probability for taking this transition, and an output probability density function (PDF) which defines the conditional probability of emitting each output symbol from a finite alphabet, given that that the transition is taken (Lee 1989) [10].

![Figure: 1.6: Some milestones in speech synthesis](image)

The availability of free and semi-free synthesis systems, such as the Festival Speech Synthesis System and the MBROLA project, makes the cost of entering the field of speech synthesis much lower, and many more groups have now joined in the development.

Neural networks have been applied in speech synthesis for about ten years and the latest results have been quite promising. However, the potential of using neural
networks have not been sufficiently explored. Like Hidden Markov Models, neural networks are also used successfully with speech recognition (Schroeder 1993).

However, although we are now at the stage were talking computers are with us, there is still a great deal of work to be done. We can now build synthesizers of (probably) any language that can produce recognizable speech [11], with a sufficient amount of work; but if we are to use speech to receive information as easily when we're talking with computers as we do in everyday conversation, synthesized speech must be natural, controllable and efficient (both in rendering and in the building of new voices).

1.8 SOME GENERIC TTS FRAMEWORKS

This section gives some of the generic frameworks available in public domain for the development of a TTS synthesizer. Some of these act as back-end engines and others are full-featured commercial TTS frameworks.

1.8.1 MBROLA SYNTHESIZER

MBROLA is a high-quality, diphone-based speech synthesizer that is available in public domain. It is provided by the TCTS Lab of the Faculte Polytechnique de Mons (Belgium) which aims to obtain a set of speech synthesizers for as many languages as possible. The MBROLA speech synthesizer is free of charge for non-commercial, non-military applications. Anyone can send in his or her own speech recordings and an MBROLA database for synthesis is prepared. There are presently diphone databases existing for several languages: American English, Brazilian Portuguese, Breton, British English, Dutch, French, German, Greek, Romanian, Spanish and Swedish.

TCTS also provides speech database labeling software: MBROLIGN, a fast MBROLA-based TTS aligner. MBROLIGN can also be used to produce input files for the MBROLA v2.05 speech synthesizer. More information and demos of the different voices and languages and also comparisons between MBROLA and other synthesis methods can be found on the MBROLA project home page [MBROLA, 1998].
1.8.2 FESTIVAL

The Festival TTS synthesizer was developed in CSTR at the University of Edinburgh by Alan Black and Paul Taylor and in co-operation with CHATR, Japan [Black et al., 2001]. It is a freely available complete diphone concatenation and unit selection TTS synthesizer. Festival is the most complete freeware synthesis system and it includes a comprehensive manual. Festival offers a general framework for building speech synthesis systems as well as including examples of various modules. As a whole, it offers full TTS synthesizer through a number of APIs. Festival is multi-lingual (currently English, Spanish and Welsh). The English version is most advanced and the developments for this version are very fast.

The synthesizer is written in C++ and uses the Edinburgh Speech Tools for low-level architecture and has a Scheme (SIOD)-based command interpreter for control. The latest details and a full software distribution of the Festival Speech Synthesis System are available through the Festival home page [Black et al., 2001].

1.8.3 FLITE

Flite (Festival-lite) is a smaller, faster alternative version of Festival designed for embedded systems and high volume servers. Flite is designed as an alternative synthesis engine to Festival for voices built using the FestVox suite of voice building tools.

1.9 Speech Synthesis Markup Languages: An Overview

Plain text alone does not contain enough information for a speech engine to know how to accurately synthesize it with speech. There are many ambiguous and implicit features of text that cannot accurately be inferred from text alone. Examples of such ambiguities include word pronunciations and paragraph delineation. Similarly, the prosody and intonation of synthesized speech cannot be inferred with complete accuracy without metadata accompanying the text. Other speaker directives, such as speaker/voice characteristics, are also absent in plain text.

A number of attempts have been made to create open standards for speech markup, and ultimately to develop a common markup control language for text-to-speech synthesis. It should be noted, however, that not all these languages have the same goals.
Many of the speech markup languages explored in this document aim to provide the ability for experts and non-experts to markup input text to guide speech synthesis.

Following are such markup languages:

1.9.1 Spoken Text Markup Language (STML)

Building on the work done for SSML, Paul Taylor and Amy Isard (the original SSML authors) from Edinburgh University and Richard Sproat and Michael Tanenblatt from Bell Labs worked collaboratively to create a specification for the Spoken Text Markup Language (STML). The aim for STML was again to make a markup language that worked with a variety of synthesis engines.

1.9.2 Java Speech API Markup Language (JSML)

Around 1997, Sun Microsystems finished development of its own speech markup language specification, the Java Speech API Markup Language (JSML), which it released as part of its Java Speech API (JSAPI). JSAPI defines an abstract Application Programming Interface (API) in the form of a set of abstract classes and interfaces that represent a Java programmer's view of a speech engine. JSAPI also defined JSML and the Java Speech API Grammar Format (JSGF) as companion specifications [12].

1.9.3 SABLE

After their work on STML, Edinburgh and Bell Labs continued to collaborate and work towards developing a single speech markup standard. First published in 1998, the SABLE specification was co-authored by researchers from the University of Edinburgh and Bell Labs, along with representatives from Sun Microsystems, Boston University, CMU and BT Labs.

SABLE was designed to provide a single standard for speech synthesis markup. The principal aim was to address the problems created by the existence of a large number of incompatible and proprietary speech synthesis control languages. To address these issues, the authors based the Sable standard on both STML and JSML.
1.9.4 W3C Speech Synthesis Markup Language (SSML)

The W3C’s Speech Synthesis Markup Language (SSML) [13] is based largely on the Java Speech API Markup Language (JSML), but also takes inspiration and concepts from the SABLE specification. SSML also adds new elements, not present in either of the JSML or SABLE standards.

The aim of SSML is to provide a feature rich, XML-based markup language for controlling and guiding the generation of synthesized speech in Web and other applications. As with SABLE, JSML etc., the main role of SSML is to provide a standard way to control aspects of speech such as pronunciation, volume, pitch, and rate across multiple heterogeneous speech synthesis platforms.

1.9.5 Apple Speech Synthesis Manager

The Speech Synthesis Manager (formerly known as the Apple Speech Manager) is the part of the Mac OS that provides a standard method for Macintosh applications to generate synthesized speech.

This Speech Synthesis interface is designed with very different goals to the speech markup languages, in that it is squarely targeted at dedicated application developers, who must access the specific functions in the Speech Synthesis API to synthesize speech and control speech parameters.

1.9.6 Microsoft Speech API (SAPI)

The leading vendors are beginning to support Microsoft’s Speech API, or SAPI, which is based on the COM specification and is being adopted as the industry standard. The motive of SAPI is to eventually allow interoperability between the speech engines. The Microsoft Speech API provides applications with the ability to incorporate speech recognition (command & control dictation) or TTS, using either C/C++ or Visual Basic. SAPI follows the OLE Component Object Model (COM) architecture. It is supported by many major speech technology vendors. The major interfaces are:

- **Voice Commands**: high-level speech recognition API for command and control. Voice Text: simple high-level TTS API.
The Voice Text: object is available in two forms: a standard COM interface IVoiceText and companion interfaces, and also an ActiveX COM object, VtxtAuto.dll

Multimedia Audio Objects: audio I/O for microphones, headphones, speakers, telephone lines, files etc. With the Microsoft Speech SDK, and in particular, the TTS VtxtAuto ActiveX COM object, any developer can create a TTS-enabled application using a few simple commands, such as register and speak.

1.9.7 Microsoft Speech Application Software Development Kit (SASDK)

Microsoft seems to be phasing SAPI as a standard, instead choosing to focus on its Speech Application SDK (SASDK), which is a set of development tools supporting SALT (Speech Application Language Tags) for speech related technology. The SASDK is particularly focused on incorporating speech functionality into web applications [17].

The SASDK supports two types of speech output – TTS synthesis (using SSML – based on the W3C draft from April 2002) and Recorded prompts. The recorded prompts are clearly aimed at telephony style speech applications, where speech output is fairly limited in scope. The recorded prompts require human speech to be recorded and stored in a database, which is then used to provide concatenated speech output in preference to TTS synthesized speech. TTS is mostly used as the fallback option for situations where pre-recorded speech is not available.

1.9.8 VoiceXML (VXML)

VoiceXML is designed for creating audio dialogs that feature synthesized speech, digitized audio, recognition of both speech input and telephony based Dual Tone Multi-Frequency (DTMF) key input, recording of speech input, telephony, and mixed initiative conversations. Its major goal is to bring the advantages of web-based development and content delivery to interactive voice response applications.

1.9.9 SML in VHML

VHML (Virtual Human Markup Language) is an attempt to combine existing markup-languages developed for the various aspects of human-computer interaction
(e.g. facial expression, body animation, emotional representation) into a unified specification language. The sub-part of VHML concerned with the markup for speech synthesis is called Speech Markup Language (SML) and is – according to the current “VHML Working Draft v0.3” from 21.Oct.2001 – based on W3C’s SSML. Comparing SML to the current version of SSML points out, that roughly speaking SML currently is a slightly downsized variant of SSML. The most important difference to SSML and SAPI is, that it foresees the labelling of the speaker’s emotion via VHML’s Emotional Markup Language (EML). Emotion-tags specified in EML are inherited by SML and are thus visible to the speech synthesis.

1.10 Sort summary of some speech Engine

A text-to-speech system is sometimes called speech engine or text recognition software. There are several text recognition software available in the market, which converts text to voice form. Some of these software, engines are:

1. **Annuncify**

Annuncify (formerly known as SayMyName) reads out loud the caller's name. Annuncify is saves you from annoying calls and unnecessary interruptions [16]. Additionally, Annuncify offers a broad range of plugins - from all kinds of messaging, like Facebook, Twitter, SMS and Gtalk - to crazy things like reading out loud a book for you and an API allowing other developers to integrate their app easily. Other features includes:

- reads SMS, MMS, Gmail, Gtalk, K-9 Mail, Plume (formerly Touiteur) and Twidroyd too
- mute gesture
- very customizable
- Text-To-Speech, TTS
- Open-Source

2. **Balabolka**

Balabolka is a Text-To-Speech (TTS) program [14]. All computer voices installed on your system are available to Balabolka. The on-screen text can be saved as a WAV, MP3, MP4, OGG or WMA file. The program can read the clipboard content, view the text from AZW, AZW3, CHM, DjVu, DOC, EPUB, FB2, HTML,
LIT, MOBI, ODT, PRC, PDF and RTF files, customize font and background colour, control reading from the system tray or by the global hotkeys.

The program uses various versions of Microsoft Speech API (SAPI); it allows to alter a voice's parameters, including rate and pitch. The user can apply a special substitution list to improve the quality of the voice's articulation. This feature is useful when you want to change the spelling of words. The rules for the pronunciation correction use the syntax of regular expressions.

Balabolka can save the synchronized text in external LRC files or in MP3 tags inside the audio files. It is Freeware and supports following languages: English, Arabic, Bulgarian, Chinese (Simplified), Chinese (Traditional), Croatian, Czech, Dutch, Finnish, French, German, Greek, Hungarian, Italian, Japanese, Korean, Polish, Portuguese (Brazil), Portuguese (Portugal), Romanian, Russian, Serbian, Spanish, Turkish, Ukrainian, Vietnamese.

3. Chrome Speak

Chrome Speak (App Version), select the text and right-click to speak with offline TTS (text to speech) engine [18]. Chrome speak provides native support for speech on Windows (using SAPI 5), Mac OS X, and Chrome OS, using speech synthesis capabilities provided by the operating system. On all platforms, the user can install extensions that register themselves as alternative speech engines.

4. DSpeech

DSpeech is a TTS (Text To Speech) program with functionality of ASR (Automatic Speech Recognition) integrated. It is able to to read aloud the written text and choose the sentences to be pronounced based upon the vocal answers of the user. It is specifically designed to quickly and directly provide the functions and improved practical usefulness that are requested by this kind of program. In the meantime, the invasiveness and resource consumption is minimal. Some notable features of DSpeech are:

- Allows you to save the output as a .WAV, .MP3, AAC, WMA or OGG file.
- Allows you to quickly select different voices, even combine them, or juxtapose them in order to create dialogues between different voices.
- DSpeech integrates a vocal recognition system that, through a simple script language, allows you to create interactive dialogues with the user.
• Allows you to configure the voices in an independent way.
• Thanks to apposite TAGs, it allows you to dynamically change the features of the voices during the playback (speed, volume and frequency), to insert pauses, emphasize specific words, or even to spell them out.
• Allows you to capture and reproduce the content of the Clipboard.
• DSpeech is compatible with all vocal engines (SAPI 4-5 compliant).
• AI dialog system. Not really useful, but amusing. It does not work in every language.

5. Free Natural Reader

NaturalReader is a Text to Speech software with natural sounding voices [15]. This easy to use software can convert any written text such as MS Word, Webpage, PDF files, and Emails into spoken words. NaturalReader can also convert any written text into audio files such as MP3 or WAV for your CD player or iPod.

NaturalReader has many other functions, such as OCR. OCR function works with scanner to convert printed characters into digital text and it is up to 99% accurate. This allows to listen printed file or edit it in a word-processing program.

6. PowerTalk

PowerTalk is a free program that automatically speaks any presentation or slide show running in Microsoft PowerPoint for Windows. You just download and install PowerTalk and while you open and run the presentation as usual it speaks the text on your slides. The advantage over other generic 'Text To Speech' programs is that PowerTalk is able to speak text as it appears and can also speak hidden text attached to images.

Speech is provided by the synthesised computer voices that are provided with Windows 7, Vista and XP, and other voices are available. PowerTalk uses PowerPoint supplied with Microsoft Office to show the presentation.

7. Select and Speak
Select and Speak uses iSpeech's human sounding text to speech (TTS) to let you select text from almost any website and make it talk.

8. SpeakIt!

Select text you want to read and listen to it. SpeakIt converts text into speech so you no longer need to read. SpeakIt reads selected text using Text-to-Speech technology with language auto-detection. It can read text in more than 50 languages.

9. SpokenText

SpokenText lets you easily convert text into speech. Record (English, French, Spanish or German) PDF, Word, plain text, PowerPoint files, and web pages, and convert them to speech automatically. Download your recordings as .mp3 or .m4b (Audio Book) files (in English, French, Spanish and German) of any text content on your computer or mobile phone.

10. Text to Voice

TTS gives Firefox the power of speech. Select text, click the button on the bottom right of Firefox window and this add-on speaks the selected text for you. Isn't it brilliant? Audio is downloadable.

11. text2speech

Free online text to speech converter. Just enter your text, select one of the voices and download the resulting mp3 file to your computer. This service is free and you are allowed to use the speech files for any purpose, including commercial uses.

12. Voki

Voki is a FREE service that lets you create customized avatars, add voice to your Voki avatars, post your Voki to any blog, website, or profile, and take advantage of Voki's learning resources.

13. vozMe
Vozme is an online text to speech program that lets you type-in any English or Spanish text and then play it as an audio stream.

14. WordTalk

A free Windows text-to-speech plugin for Microsoft Word. It will speak the text of the document and will highlight it as it goes. It contains a talking dictionary and a text-to-mp3 converter. WordTalk is a free text-to-speech plugin developed for use with all versions of Microsoft Word (from Word 97 up to Word 2013). Siting neatly in your Microsoft Word toolbar it is highly configurable. It will speak the text of the document and will highlight it as it goes. WordTalk was conceived and developed by Rod Macaulay of TASSC in Aberdeen.

1.11 Linguistic studies in India

India is a country with more than 15 official languages and each language is spoken in different forms across different places. According to experts, Speech recognition and Speech synthesis technology can be very useful in the Indian context as it provides an easy interface for interacting with computers. Using such a convenient means of rendering information to or from the machine would mean that the end-user need not be computer literate and still can use the power of the IT industry.

There are several language problems in India which IT can solve. In this regard Government of India has funded several projects taken up by various institutions all over the country.

From the point of view of TTS development, the most helpful aspect of Indian scripts is that they are basically phonetic in nature, and there is one-to-one correspondence between the written and spoken forms of most of the Indian languages. This makes the task of automatic phonetization simpler [Krishna, 2002].

This chapter is a survey of the work done in the field of TTS for Indian languages. The goal of this survey is to study the techniques used for Indian language TTS development and identify suitable techniques for development of Gujarati TTS synthesizer. Even among Indian Languages, focus is given more to the work done for Hindi TTS synthesizer, keeping in mind that Hindi text and speech resembles with
that of Gujarati text and speech. The different works have been studied from the point of view of following aspects:

- Is the code available?
- Can software be run for demo?
- Is it a framework?

Several institutions in India are working in the field of TTS for Indian languages. The work done in the following institutions is discussed in detail.

- C-DAC Bangalore - Matrubhasha API [Raman, 2004]
- IIT Mumbai - Vani Framework
- HP Labs - Hindi TTS
- IIT Kharagpur - SHRUTI TTS
- Simputer Trust - Dhvani TTS [Dhvani, 2001], [Hariharan, 2002]

Some other institutions where the TTS development is going on, include – IIT Madras, IIIT Hyderabad, HCU, IISc Bangalore, Utkal University, TIFR Mumbai, CDAC Noida and College of Engineering, Guindy.

1.11.1 C-DAC BANGALORE – MATRUBHASHA API

C-DAC Bangalore (formerly National Centre for Software Technology (NCST)) is an autonomous society, involved in Research & Development, under the administrative purview of Department of Information Technology, Ministry of Communications & Information Technology, and Government of India.

Matrubhasha is a project carried out at C-DAC Bangalore, as a part of the digital divide bridging activities [Raman, 2004]. It was undertaken with the intention of making end user applications speak and listen to the masses in any Indian language that they are comfortable to communicate in.

Matrubhasha is a Unicode and MBROLA based software solution for TTS synthesis and, CMU Sphinx based Speech Recognizer for Indian languages. It is visualized with the objective of building a framework, which can be used by any software developer to incorporate speech capabilities (in Indian languages) into his/her software thus increasing its usability across different sections of society.

The Matrubhasha project is an activity of the ICT Research and Training Centre (India) of Development Gateway Foundation. The Government of India is a member of the Development Gateway Foundation, a World Bank initiative. An
Information and Communication Technologies (ICT) - Research & Training (R&T) Centre of the Foundation has been set up in Bangalore, India with Centre for Development of Advanced Computing (C-DAC) as the Project Implementing Agency and the Indian Institute of Technology (IIT), Bombay as the first collaborating institution.

**Matrubhasha provides:**

- One single engine and different language bases. By just using different language rule files, the same engine speaks different languages, and thus speaks multiple languages from the same document.
- Tools with a linguist friendly user interface which help the linguists to create language bases for any language and any dialect and create rules in the language base so as to bring out correct pronunciation by imposing those rules during conversion of written text to spoken phonemic form. These tools include Anuvaachak, Uchharak and Bhavna.
- An application programming interface (API) that speech enables any application. This API can also be extended to create plug-ins to general purpose utilities like office products and internet browsers.

### 1.11.2 IIT MUMBAI– VANI FRAMEWORK

Vani is a TTS synthesizer proposed by IIT Mumbai. It is primarily developed for Hindi, but with minor modification could directly be modified for any language which is phonetic in nature. The approach is similar to concatenation synthesis, with phonemes as the basic unit. However, the phonemes are not selected from a database, but are generated from a more basic unit, which they call fract-phoneme in the case of vowels. The basic observation is that vowels look like a continuous repetition of these very small segment phonemes called fract-phonemes. Since fract-phonemes are very small in size, they are a good choice for acting as a basic unit.

The aim of Vani was to allow complete specification of speech, i.e., one can get the software to speak exactly what they want to. This means typically that the software can also sing if so desired. Also emotions can be expressed. In order to give complete freedom of expression to the user Vani represents the speech by a new encoding scheme called vTrans which is an extension of iTrans encoding scheme. Vani generates speech from a given vTrans file.
For Schwa deletion algorithms for Hindi, Vani uses a rule-based algorithm [Basu, 2002]. Vani is built using java to enable platform independence and uses Java Sound API (JSAPI).

1.11.3 HP LABS – HINDI TTS

HP Labs, India have developed a Hindi TTS synthesizer based on Festival framework. This effort is a part of the Local Language Speech Technology Initiative (LLSTI), which facilitates collaboration between motivated groups around the world, by enabling sharing of tools, expertise, support and training for TTS development in local languages. It aims to develop a TTS framework around Festival that will allow for rapid development of TTS synthesizers in any language.

Since Festival does not provide complete language processing support specific to various languages, it needs to be augmented to facilitate the development of TTS synthesizers in certain new languages. Because of this, a generic G2P converter has been developed at HP Labs India as part of the LLSTI initiative. The system consists of a rule processing engine which is language independent. Language specific information is fed into the system in the form of lexicon, rules and mapping.

1.11.4 IIT KHARAGPUR- SHRUTI TTS

An Indian language TTS synthesizer (named SHRUTI) that accepts text inputs in two Indian languages namely Hindi and Bengali and produces near natural audio output has been developed at IIT Kharagpur [Mukhopadhyay, 2006]. The synthesizer runs on a Compaq iPaq PDA built around the Intel Strong Arm-1110 processor running Microsoft Pocket PC, a customized version of Microsoft’s operating system WinCE for mobiles and other handheld devices. The synthesizer has also been ported to a Casio Cassiopeia built around a MIPS 124 processor running Pocket PC. Two versions of the synthesizer have been built, one which resides on the system memory and another which runs from a storage card.

1.11.5 SIMPUTER TRUST – DHVANI TTS

The Simputer Trust is a registered (1999) non-profit charitable trust with the broad goal of harnessing the potential of Information Technology for the benefit of the weaker sections of society. It has brought together people from two entities: Computer Science and Automation Department of the Indian Institute of Science and Encore Software (formerly Ncore technologies). Simputer is a low-cost multilingual, mass access handheld device that uses Indian Language User Interfaces.
to provide Information technology based services to the multilingual population of India [Hariharan, 2002]. It is a low cost portable alternative to PCs, by which the benefits of IT can reach the common man. The Amida Simputer was launched in Bangalore on 26th March 2004 by Picopeta Simputers Pvt. Ltd. It is a retail product.

DHVANI is the TTS effort of the Simputer Trust. The aim of this effort is to ensure that literacy and knowledge of English are not essential for using the Simputer. Using images in conjunction with voice output in local languages makes the Simputer accessible to a larger fraction of the Indian population.

Dhvani has a C/Linux implementation. Currently, it has a phonetic-to-speech engine which is capable of generating intelligible speech from a suitable phonetic description in many Indian languages. In addition, it is capable of converting UTF-8 text in Hindi and Kannada to the phonetic description, and then speaking it out using the phoneticto-speech engine.

**1.10.6 OTHER INSTITUTIONS**

IIT Madras, IIIT Hyderabad, HCU, IISc Bangalore, Utkal University, TIFR Mumbai, C-DAC Noida and College of Engineering, Guindy: These institutes are also working on TTS synthesizers for various Indian Languages. Following is a brief description of the techniques used by these institutions for developing their TTS synthesizers.

**1.11.6.1 IIT MADRAS**

Systems Development Laboratory, IIT Madras has come up with a speech enabled multilingual editor (http://acharya.iitm.ac.in) which supports text processing and speech generation for several Indian languages including Telugu. The speech is produced using the MBROLA speech engine. Since the databases required for Indian languages are not yet available for use with MBROLA, the IITM team had experimented with other available data bases where the phonemes are close to the phonemes of Indian languages. In this regard, Swedish database was used for generating the speech for Indian languages as the phonemes of Swedish are well suited to produce speech output in Indian Languages.

The Telecommunication and Computer Networking (TeNeT) Group, a coalition of 14 faculty members from the Electrical Engineering and Computer Science & Engineering Departments of IIT-Madras, has taken initiative for
developing local language speech interface system. In this regard, they have worked with Festival to perform speech synthesis. Their system has following features:

- Common phoneset for Hindi and Telugu, also usable for other Indian language.
- Diphone unit selection for synthesis.
- Data-driven prosody modeling using Classification and Regression Tree (CART).
- Concatenative synthesis technique is used to produce natural sounding speech.

Also, a separate group in Speech and Vision Lab, Department of Computer Science and Engineering, IIT Madras, is working with an objective to address different issues in acquisition and incorporation of duration and intonation knowledge in the context of TTS synthesis for Indian languages.

1.11.6.2 IIIT HYDERABAD

Language Technologies Research Centre (LTRC), IIIT Hyderabad has as its goal the development of technologies dealing with language. It includes technologies pertaining to translation and other NLP areas, speech processing, optical character recognition, etc. Their research and development focus is to develop speech interfaces for Indian Languages. Currently, they are working on TTS synthesizers for Indian languages. They already have some restricted domain TTS synthesizers developed for Telugu and Hindi working in a number of domains.

LTRC has used data-driven (Corpus-based / Example-based) approach using festvox for the development of TTS synthesizers. They have used Unicode as the input to the front end for the text processing. They experimented with different choices of units: syllable, diphone, phone and half phone, for unit selection speech synthesis for Hindi. The observation they have come up with is that the syllable unit performs better than diphone, phone and half phone, and seems to be a better representation for languages such as Hindi. Also, the half phone synthesizer performs better than diphone and phone synthesizers. They have also proposed a data-driven synthesis method for Indian languages using syllables as basic units for concatenation.

Some applications developed by LTRC based on TTS technology are:

- SAMACHAR VANI – a framework for automated spoken news service. It is TTS based newsreader software.
- TALKING TOURIST AID – an application which allows a non native Hindi/Telugu person to express his queries/concern related to city, travel,
accommodation etc., in the language of the native speaker. This application is ported to Simputer.

- SCREEN READER -for visually handicapped. A preliminary version of it is developed using Hindi voices. They are also working on a Hindi speech synthesizer that fits in 1.45 MB flat. Such a small speech synthesizer can be put on small systems like PDAs and embedded devices including mobiles. The only other speech synthesizer available at comparable sizes for Indian languages is the Dhvani speech synthesizer which is more than 2 MB in size. LTRC has also come up with an implementation of an API for the TTS synthesizer in Windows for robust application development in Windows environments.

1.11.6.3 HYDERABAD CENTRAL UNIVERSITY (HCU) - VAANI

Language Engineering Research Centre (LERC) at Resource Centre for Indian Language Technology Solutions (RCILTS), HCU has developed several products related to speech technologies which include an experimental TTS synthesizer for Telugu -VAANI. Vaani uses MBROLA as a back end. They have used the diphone segmentation approach. The Telugu voice used by them is developed by Jahnavi Ayachitam and Kalpana Reddy KVK of ISS College of Info. Tech. and Engineering for Women, Hyderabad, AP, INDIA. Presently, this is the only Telugu diphone database available for MBROLA synthesizer.

1.11.6.4 IISC BANGALORE -THIRUKKURAL & VAACHAKA

IISc Bangalore has developed a complete Tamil and Kannada TTS synthesizer named Thirukkural and Vaachaka respectively. Similar techniques are applied in the development of both the TTS synthesizers. Syllables of different lengths have been selected as units. Automatic segmentation algorithm has been devised for segmenting syllables into consonant and vowel. Thirukkural is designed in VC++ and runs on windows 95/98/NT.

1.11.6.5 UTKAL UNIVERSITY, ORISSA

Speech Tech Group (STG), RC-ILTS-ORIYA Utkal University, Orissa has developed an ORIYA TTS synthesizer. This TTS synthesizer is designed by the Character Based Concatenation technique. In this method, the words are parsed into characters and a character is segregated into pure consonant and vowel. For example, ‘ka’ is obtained from the word ‘kataka’ and further this is separated as pure consonant
‘k’ and ‘a’. Likewise, all the pure consonants and vowels are stored in the ‘.wav’ format. As the vowels are dominant in the utterance, they are stored for different durations as they occur in the word. Using the techniques of Artificial Intelligence the synthesis of sound is obtained. Further the transition between two characters is stored by taking the help of Paninian philology to give a natural shape to the output. These rules help to slot in different levels of pitch for incorporating prosody in the output. The classifications on the silence region and the unvoiced regions have also been studied to place them in proper places. Intonation modeling is also incorporated in the synthesis system.

1.11.6.6 TATA INSTITUTE OF FUNDAMENTAL RESEARCH (TIFR), MUMBAI

Work has been done in TIFR, Mumbai for TTS synthesis in Indian English. The important components of the language processor used are the parser to categorize words, an Indian English phonetic dictionary, morphological analyzer, letter-to-sound rules, phonological rules, prosody rules and Indian name detector. The relevant rules are formulated with the aid of a large CMU pronunciation dictionary and a language tool GENEX, developed in TIFR, which can generate a sub-dictionary following a set of specified constraints.

1.11.6.7 C-DAC, NOIDA

The TTS project at C-DAC, Noida, is based on the concatenative approach and aims at developing a TTS synthesizer for Indian languages primarily, Hindi. The input in Unicode is processed by the Text processing unit and the speech-processing unit resulting in synthesized speech. The input text is also normalized and converted to their orthographic form, before breaking into the basic units required for synthesis.

1.11.6.8 COLLEGE OF ENGINEERING, GUINDY, CHENNAI

A preliminary version of TTS synthesizer using concatenative synthesis and diphones is developed for Tamil. They are still working on improving the quality of the speech.

1.12 Salient features of the present study

1.12.1 The present study is tried to develop the text pattern recognition tool, which is font independent.
1.12.2 The present study is mostly the first step in Gujarati language which phoneme is used as basic unit in concatenative speech synthesis method.

1.12.3 The present study is design only for Gujarati alphabets.

1.12.4 The present study generates a Phoneme database generator and a Text to speech module GUI, which is user friendly and easy to operate.

1.12.5 The speech output generated through the developed software is stored in wav format.

1.13 Glossary of terms

**Articulation:** It is the approach or contact of two speech organs, such as the tip of the tongue and the upper teeth.

**Consonant:** A consonant is any articulation that is acoustically produced by forming a constriction by one or more articulators along the vocal tract to the flow of air from the lungs. A long-duration consonant is a syllable closing sound, while a short duration consonant occurs between two vowels.

**Diphone:** A diphone is a sound that starts from the middle of one phoneme and ends in the middle of a neighboring phoneme.

**Frequency:** The precise indication of the number of hertz of a sound wave at any instance.

**Formant:** A formant is the resonating frequency of the air in the vocal tract. A formant is a peak in an acoustic frequency spectrum.

**Grapheme:** Grapheme is a string consisting of one or more letters that together correspond to exactly one phoneme.

**Intonation:** Intonation is the pattern of fundamental frequency changes that occur during a phrase or a sentence.

**Letter:** A letter is an element of the alphabet of a language

**Linguistics:** The scientific study of language, which may be undertaken from many aspects, for ex: - structure of words (morphology), meanings (semantics) etc.

**Morpheme:** A morpheme is a smallest language unit that carries a semantic interpretation. For example, the word ‘unbelievable’ can be divided into the three morphemes un-believe-able.

**Phoneme Modifier:** A symbol defined to adjust the pronunciation of an individual phoneme. Phoneme modifiers are also called prosodic control symbols.

**Prosody:** The rhythm, intonation and lexican stress in speech. It is study of speech- rhythms.

**Phoneme:** The smallest unit of sound in a language. It is an abstract unit that represents sounds and writing in a systematic, unambiguous way.

**Phonetic Transcription (Phonetic Notation):** It is the visual system of symbolization of the sounds occurring in spoken human language. The most common type of phonetic transcription uses an International Phonetic Alphabet. With phonetic
transcriptions, dictionaries tell you about the pronunciation of words. Phonetic transcription is necessary, because the spelling of a word doesn’t tell you how to pronounce it.

**Phonetics:**
Phonetics is the representation of the sounds of a language.

**Phonology:**
It is the description of the systems and patterns of sounds that occur in a language.

**Sound:**
A sound is an acoustic signal. In the context of a TTS system, sounds are generated by the synthesizing component but not by the grapheme-to-phoneme conversion component.

**Speech pitch:**
The middle pitch of a voice, from which the actual pitches of the speech can vary with rising and falling tunes. Pitch is a combination of the average speaking frequency and its variations around that average.

**Speech rate:**
The approximate number of words of text that the synthesizer speaks in a minute.

**Speech Recognition System:**
Software that allows an individual’s voice to be recognized by the computer.

**Speech Synthesis:**
Artificial speech that is generated by computers. For example: a computer can “Read” web page and produce spoken email output.

**Speech Synthesizer:**
An output device that enables a computer to speak.

**Speech volume:**
The average amplitude at which the speech channel generates speech.

**Syllable:**
A syllable is a word or a part of a word that is uttered by a single effort of the voice. A syllable is structured from phonemes that are consonants and vowels. A vowel is usually the syllable nucleus while a consonant usually represents the syllable margin.

**Synthesis unit:**
A synthesis unit is an acoustical entity that represents a complete or partial sound. These units join to form syllables, words, or phrases.

**Vocal Tract:**
The vocal tract is the air passage above the larynx. It consists of the oral and nasal tracts.

**Voice:**
A component containing data and, optionally, executable code that helps to shape the sound of the synthesized speech.

**Vowel:**
A vowel consists of a single vocal sound that is produced by a continuous passage of air from the lungs on an open vocal tract. This definition applies to both short and long vowels.
1.14 Organization of the Thesis

Researcher has distributed entire work into five different chapters. Summary of the remaining chapters i.e. from chapter 2 to chapter 5 is as follow.

**Chapter 2: TEXT TO SPEECH CONVERSION TECHNOLOGY**

This chapter contains literature review regarding text to speech conversion technology. It contains review regarding previous research work carried out in this area as well as various sources which includes journal papers, conference papers, electronic documents, web resources.

**Chapter 3: DESIGNING & DEVELOPMENT OF TEXT TO SPEECH CONVERSION MODEL**

This chapter presents analysis of Gujarati characters (Consonants and vowels), its different CV combinations and phoneme schema to be considered in this study. It explains entire conversion process in form of different models.

**Chapter 4: PROTOTYPE AND COMPONENTS DEVELOPMENT FOR THE TEXT TO SPEECH CONVERSION MODEL**

On base of models discussed in chapter 3, a component development process is discussed in this chapter. It focuses on hardware and software requirement during development process as well as testing the system. It also describes step by step how the concatenative methodology is applied developing “jjktts” text to speech system. Different faces in this chapter include master table creation, sound recording and TTS engine creation.

**Chapter 5: RESULTS, DISCUSSION, CONCLUSION AND FUTURE SCOPE FOR EXTENSION OF THE RESEARCH WORK**

This chapter discusses outcome of the proposed jktts system applied on entered Gujarati text to measure success of proposed research work. Result is analyzed based on various parameters such as success ratio of Listening test and Categorical Rating Test. Further this chapter presents conclusion of proposed research work along with direction for future scope in the present research area.
REFERENCES