Development of Punjabi Text To Speech Application for Mobile Phone

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

Master of Technology

in

Computer Science and Application

Submitted By

Ankita Goel

(Roll No. 601003002)

Under the supervision of:

Khushneet Jindal

System Analyst, SMCA

SCHOOL OF MATHEMATICS AND COMPUTER APPLICATIONS

THAPAR UNIVERSITY

PATIALA – 147004

June 2012

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Development of Punjabi Text To Speech Application for Mobile Phone*", in partial fulfillment of the requirements for the award of degree of Master of Technology in *Computer Science and Applications* submitted in School of Mathematics and Computer Applications, Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Mr. *Khushneet Jindal* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Ankita Goel

(Ankita Goel)

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

(Khushneet Jindal)
System Analyst
SMCA

Countersigned by

(Dr. S.S Bhatia)
Head, School of Mathematics and Computer Applications
Thapar University
Patiala

(Dr. S.K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to **Mr. Khushneet Jindal** System Analyst**,** School of Mathematics and Computer Applications, Thapar University, Patiala for his patient guidance and support throughout this report. I am truly very fortunate to have the opportunity to work with him.

I am also thankful to **Dr. S.S. Bhatia, Head,** School of Mathematics and Computer Applications**,** as well as **PG Coordinator Mr. Singara Singh, Assistant Professor,** School of Mathematics and Computer Applications, entire faculty and staff of School of Mathematics and Computer Applications and then friends who devoted their valuable time and helped me in all possible ways towards successful completion of this work. I thank all those who have contributed directly or indirectly to this work.

Lastly, I would also like to thank my parents for their years of unyielding love and encourage. They have always wanted the best for me and I admire their determination and sacrifice.

Ankita Goel

(601003002)

ABSTRACT

In recent years speech synthesis and recognition has achieved lot of success in the field of Information Technology. Speech is means of communication between different people. This thesis is highly motivated to help visually impaired people, partially sighted people and to promote Punjabi language.

Text to Speech Synthesis broadly works in two parts front end and back end. The front end takes input in the form of raw text (*i.e* input sting) and produce output a symbolic linguistic representation. Further, the back end takes the symbolic linguistic representation as input and outputs the synthesized waveform. This report discusses about the different phases in the text analysis and various techniques used for determining the symbolic linguistic representation. The accuracy for determining the linguistic representation plays important role in text to speech synthesis. Speech Synthesizers are used for synthesis of the speech. There are different types of techniques available to synthesise a speech signal. Concatenative synthesis basically selects the units (phoneme, syllable and words) and synthesize into the waveform. Formant Synthesis uses a set of parameters like frequency, amplitude and pitch to synthesize a speech signal. Articulatory synthesis uses set of articulators (like tongue, jaw, teeth) and generate speech. But concatenative synthesis synthesizes a natural speech.

This thesis work is concerned with the development of a mobile text to speech synthesis application of Punjabi language. The goal of this project is to utter Punjabi speech when input is provided in English language. The research work is carried out with the aim to maximize the system intelligibility and naturalness. To achieve the desired aim some techniques and customized rules are followed to resolve the ambiguities present in input text *i.e* resolution of issues like Initials, numerals and titles *etc*. During mapping of English text to Punjabi Phonemes many issues arose and maximum of them are resolved to some extent. An efficiency of 82% is achieved while resolving the mapping. Finally, for generation of output speech, concatenative technique is used. It produces the desired output speech with some limitations.

TABLE OF CONTENTS

	CONTENTS	Page No.
	CERTIFICATE	i
	ACKNOWLEDGEMENT	ii
	ABSTRACT	iii
	CONTENTS	iv
	List of Tables	vi
	List of Figures	vii
1	**Introduction**	
	1.1 Speech Processing	1
	1.2 TTS Synthesiser	1
	1.2.1 Text and Linguistic Analysis	2
	1.2.2 Prosody and Speech Generation	2
	1.3 History of Text to Speech Synthesis	3
	1.3.1 From Mechanical to Electrical	3
	1.3.2 Electrical Synthesiser	4
	1.3.3 Computer Synthesiser	5
	1.4 Quality of TTS Synthesiser	5
	1.5 Application of TTS System	5
	1.6 Thesis Outline	7
2	**Literature Review**	
	2.1 Architecture of TTS System	8
	2.1.1 Text Analysis	9
	2.1.2 Automatic Phonetization (L to S)	10
	2.1.3 Prosody	13
	2.1.4 DSP Component	14
	2.2 Units of Speech	19
	2.3 TTS for Indian Languages	21
	2.4 Different Researches	22

2.5 Punjabi/English Phonology
2.5.1 Punjabi Vowels 23
2.5.2 Punjabi Consonants 24
2.5.3 English Phonology 25

3 Problem and Proposed Solution

3.1 Problem Statement 26

3.2 Proposed Solutions 28

3.3 Pre-requisites 28

3.4 Speech Unit/Data Preparation
3.4.1 Speech Unit Selection 29
3.4.2 Syllable/Phoneme Recording 30

3.5 Database Design 30

3.6 Pre-Processing 36

3.7 Grapheme to Phoneme Mapping 40

3.8 Speech Synthesis 45

4 Test and Results

4.1 Test Cases 47

4.2 Result 52

5 Conclusion and Future Scope

5.1 Conclusion 55

5.2 Future Scope 55

5.3 Limitations 56

6 Limitations 57

7 References 58

LIST OF TABLES

Table No.	Title	Page No.
Table 2.1	Classification of Vowels based on Position of Tongue	23
Table 2.2	Classification of Vowels based on height of tongue tip	24
Table 2.3	Classification of Vowels based on shape of lips	24
Table 3.1	Titles	30
Table 3.2	Initials	31
Table 3.3	Phonemes	31
Table 3.4	Syllables	32
Table 3.5	Unknown String	32
Table 3.6	English Tokens	33
Table 3.7	Special Tokens	33
Table 3.8	Punjabi Tokens	34
Table 3.9	HTK Tokens	34
Table 3.10	Numerals	35
Table 3.11	Mobile_Code	35
Table 3.12	List of Vowels with Mapping	41
Table 3.13	List of Consonants with Mapping	41
Table 3.14	Letter to Sound Rules	42
Table 3.15	Pattern for Letter ”a”	43
Table 4.1	Multi Mapping Problem	54
Table 4.2	Similar Spelling Words but Different Pronunciations	54

LIST OF FIGURES

Figure No.	Title	Page No.
Figure 1.1	Text to Speech synthesis	2
Figure 1.2	Kratzenstein's resonators	3
Figure 1.3	Wheatstone's reconstruction of von Kempelen's speaking machine	4
Figure 1.4	Voder Speech Synthesisers	4
Figure 2.1	Basic Components of TTS Synthesiser	8
Figure 2.2	Prosodic Dependencies	14
Figure 2.3	Source Filter Model	15
Figure 2.4	Cascade Formant Synthesiser	16
Figure 2.5	Parallel Formant Synthesiser	16
Figure 2.6	Concatenative Synthesis	17
Figure 2.7	HMM based Speech Synthesis	19
Figure 2.8	Basic Punjabi Consonants	25
Figure 2.9	English Vowels	25
Figure 2.10	English Consonants	25
Figure 3.1	Categorization of Names Stored in Phone Book	27
Figure 3.2	Caller's Id Names in Mobile Phone Directory	27
Figure 3.3	Text to Speech Synthesiser	29
Figure 3.4	Speech Unit	30
Figure 3.5	Grapheme to Phoneme Mapping	40
Figure 4.1	Correct and Incorrect Mapped Names	53
Figure 4.2	Accuracy of Grapheme to Phoneme Mapping	53

CHAPTER 1

INTRODUCTION

Speech is most important means of communication in day to day life. TTS (Text to Speech Synthesis) is becoming popular over the years since it is useful for visually impaired and illiterate people who can understand their native language. So, this thesis is focussed on the development of application for Mobile Phone Device for Punjabi language using text to Speech Synthesis. In this chapter, brief description of speech processing has been discussed, which includes speech synthesis and speech recognition. Further sections will give an outline of the field of speech synthesis. Then, applications of text to speech synthesis are discussed.

1.1 SPEECH PROCESSING

Speech processing is a technique in which speech signals are interpreted, understood, and acted upon. It specifically refers to the processing of human speech by computerized systems. Speech processing is further divided into two categories:

- Speech Synthesis- Speech Synthesis, in simple words, is translation of input text to spoken words.
- Speech Recognition- Speech Recognition is translation of spoken words into text.

1.2 TTS SYNTHESIZER

It takes raw text as input then formats that text according to some set of rules and procedures and thereafter, at the end produces synthesized speech using some media. A text to speech (or "engine") is composed of two parts:

- Text and Linguistic Analysis
- Prosody and Speech Generation

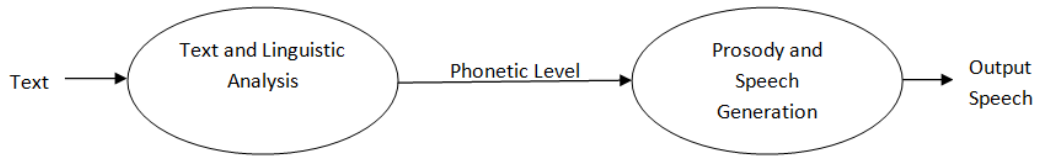


Figure 1.1 Text to Speech Syntheses

1.2.1 TEXT AND LINGUISTIC ANAYSIS

Text and Linguistic analysis has two major tasks. First, it converts text containing symbols like numbers and abbreviations into the written-out words. This process is often called text normalization/ pre-processing/ tokenization.

For example:

- "Mr." will be converted into "Mister".
- "1776" will be converted into "one thousand seven hundred seventy six".
- "Henry IV" will be converted into "Henry the Fourth".
- "987654321" will be converted into "nine eight seven six five four three two and one"

After the completion of pre-process phase phonetic transcriptions to each word will be assigned. This process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion.

For example:

- "Vineet" will be converted into "ਵ ਇ ਨ ਈ ਤ".
- "Shukla" will be converted into "ਸ਼ ਉ ਕ ਲ ਆ".

In this phase, the front end will get the phonetic transcription. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end.

1.2.2 PROSODY AND SPEECH GENERATION

In the back end, the synthesizer converts the symbolic linguistic representation into sound. There are different ways to map the symbolic linguistic representation into sound.

- Synthesis by rule- Synthesis by rule helps in production of speech with the help of these parameters (like frequency, amplitude and duration). Mainly, formant synthesis and articulatory synthesis come in the synthesis by rule.
- Synthesis by concatenation- In it pre-recorded units of speech is concatenated to form speech sound.

1.3 HISTORY OF TEXT TO SPEECH SYNTHESIS

This section discuss mainly about evolution of text to speech synthesiser

1.3.1 FROM MECHANICAL TO ELECTRICAL

In St. Petersburg 1779 Russian Professor Christian Kratzenstein (He studied the Natural Sciences at the University of Halle starting in 1742, earning a doctorate in 1746) explained physiological differences between five long vowels (/a/, /e/, /i/, /o/, and /u/) and made apparatus to produce them artificially. He constructed acoustic resonators similar to the human vocal tract and activated the resonators with vibrating reeds like in music instruments [11].

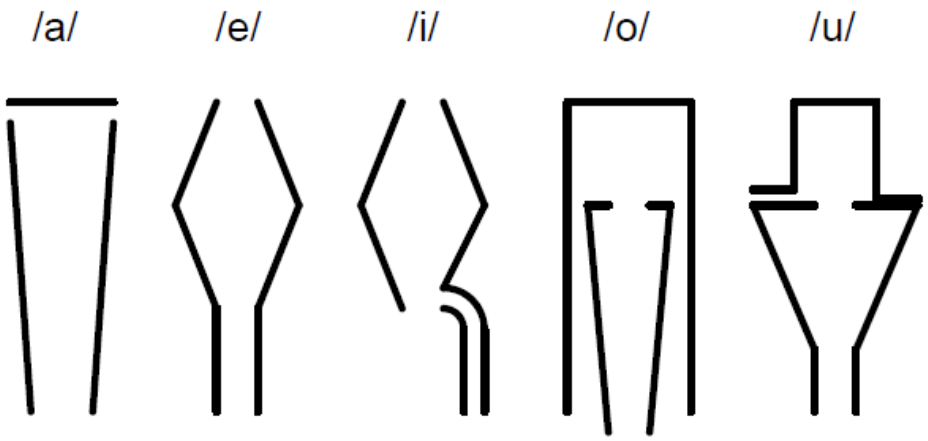


Figure 1.2 Kratzenstein's resonators *[10]*

In 1791, Kempelen achieved success in building a successful model consisting of bellows where the air being forced through whistle and adjustable leather 'vocal tract' which resulted in synthesised speech. The essential parts of the machine were a pressure chamber for the lungs, a vibrating reed to act as vocal cords, and a leather tube for the vocal tract action. In about mid 1800's Charles Wheatstone (6 February 1802 – 19 October 1875) 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.

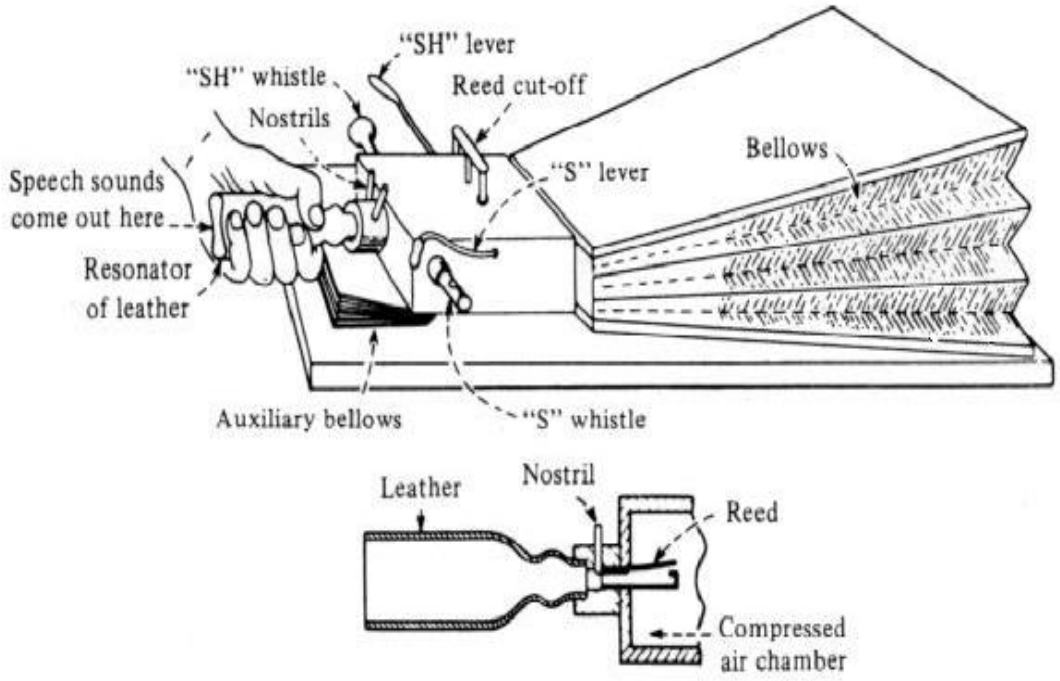


Figure 1.3 Wheatstone's reconstruction of von Kempelen's speaking machine [10]

1.3.2 ELECTRICAL SYNTHESIZER

In the 20th century, VODER (Voice Operating Demonstrator) was designed. Voder consist of wrist bar for selecting the voicing or noise bar and a foot pedal to control the fundamental frequency. VODER was inspired by VOCODER (Voice Coder) developed at Bell Laboratories in the mid-thirties. The original VOCODER was a device formalizing speech into slowly varying acoustic parameters that could then drive a synthesizer to reconstruct the approximation of the original speech signal [11].

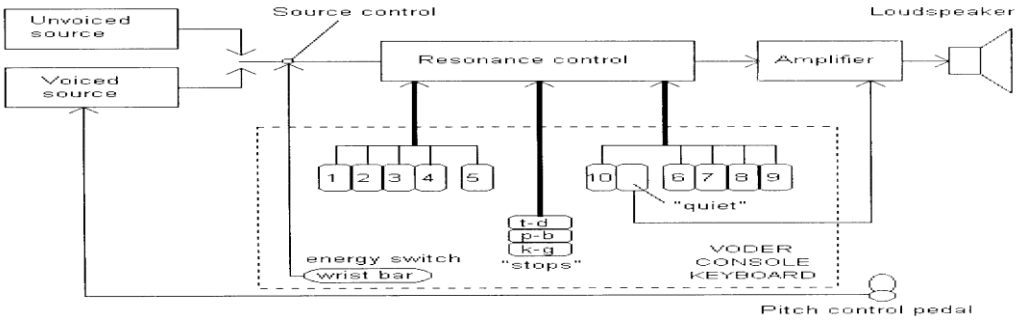


Figure 1.4 Voder Speech Synthesisers [10]

There were many more electrical synthesiser developed & experiments done during this time period.

1.3.3 COMPUTER SYNTHESISER

Thereafter, the first computer synthesiser was developed in the 1979 by Allen, Hanicutt and Klatt. They developed a MITTalk laboratory text to speech synthesis developed in MITTalk [10].After that many computer based synthesisers are developed.

1.4 QUALITY OF TTS SYNTHESIZER

- **INTELLIGIBILITY**

  It refers to how easily the output can be understood. It can be measured by taking into account several kinds of units (phonemes, syllables, words, phrases, *etc*) .For Example-

  - "Vineet" will be changed into "ਵ ਇ ਨ ਈ ਤ".
  - "8654321"will be changed into "eight six five four three two and one".

- **NATURALNESS**

  It means that up to what extent the sound generated using synthesizer is close to the human speech *i.e* whether the speech sound is like artificial or human.

1.5 APPLICATIONS OF TTS SYSTEM

Synthetic speech can be used in a number of applications. It can be used in various applications in everyday life. These are as follows

1.5.1 APPLICATIONS FOR THE VISUALLY HANDICAPPED PEOPLE

The most important and useful application in speech synthesis is reading the text that is provided directly or through external device and speaking that text majorly for visually impaired and partially sighted people. It reduces barriers to accessibility, simplifying interaction with technology for old and visually impaired people. There are many applications being developed for blind people. Screen Reader is a type of application specially developed for unsighted people or partially sighted people. It is a

kind of application that interprets and read aloud the text written in the screen. For example Browse Aloud is a type of web based screen reader that reads news updates, weather science daily report on the web and reports it.

1.5.2. APPLICATIONS IN TELECOMMUNICATION

The newest applications in speech synthesis are in the area of multimedia. Texts range from simple small messages to big messages and every text message cannot be stored on machines. Through TTS it helps to retrieve the information and produce the desired result. With the help of text to Speech Synthesis, email messages can be listened just like we are listening a telephone call. Synthesized speech may also be used to speak out short text messages (*s m s*) in mobile phones.

1.5.3 EDUCATIONAL APPLICATIONS

Synthesized speech can also be used in the field of education. In remote areas, it can be used by the language teachers for teaching their tutorial. TTS synthesis coupled with a Computer Aided Learning system can provide a helpful tool to learn a new language. Synthesized speech can also be used with interactive educational applications. It is also almost impossible to learn writing and reading without spoken help. So TTS helps to learn writing and reading.

1.5.4 TALKING TOYS AND BOOKS

With the help of TTS system the market of taking toys and books are getting a lot of success. With the help of talking toys, small children can learn alphabets, numbers easily. By just giving a command children can play with toys. Many talking dictionaries are available in the market which can help to learn vocabulary.

1.5.5. INTERACTIVE BROWSER

In the present scenario, TTS system is being incorporated with the interactive web browsers in different languages, which will be helpful to person, who can understand a language but cannot read it. It can be helpful to illiterate masses. Persons who cannot read text but listen it. There are many interactive web browsers available in the market.

1.5.6 APPLICATIONS IN EVERYDAY LIFE

Text to Speech Synthesis system is highly helpful in everyday life. It is the tool which helps to avoid accidents. It minimizes user distraction while driving, during usage of any machinery or providing any emergency service. It helps in maximum utilisation of time. By just giving a voice command, system can get to know about the instruction and acts accordingly.

1.6 THESIS OUTLINE

This thesis is divided into six chapters, including this introduction.

Chapter 1	Introduction
Chapter 2	Examines various research papers that are related to Text to Speech Synthesis. It mainly discuss about text to Speech Synthesiser. It discusses about various techniques involved in designing the Text to Speech Synthesiser. In this there is also discussion about the different Indian Text to Speech Synthesiser.
Chapter 3	Presents the problem definition and discusses about the design and development of application of Text to Speech Synthesis.
Chapter 4	It presents the results from the work done.
Chapter 5	It draws the thesis to a close by formalizing the results presented and discussed in previous chapter. The discussion is followed by a section on what further work can be done within this field.

CHAPTER 2

LITERATURE REVIEW

This section discusses about different techniques and development in the field of text to speech synthesis. Firstly the basic architecture of text to speech synthesis will be discussed. After that there will be detailed description of every component and techniques available to synthesise the waveforms. In the end there will be discussion about different Indian synthesiser available in market.

2.1 ARCHITECTURE OF TTS SYSTEM

Text to Speech Synthesis is basically divided into two components NLP and DSP [7]. NLP component is known as Natural Language Processing which helps to process the raw text. Inference engine tries to derive the answer from knowledge base and logical inference helps to derive the answer. For example, NLP decides "987654321" should be "nine eight seven six five four three two one" or "ninety eight seventy six fifty four thirty two and one".

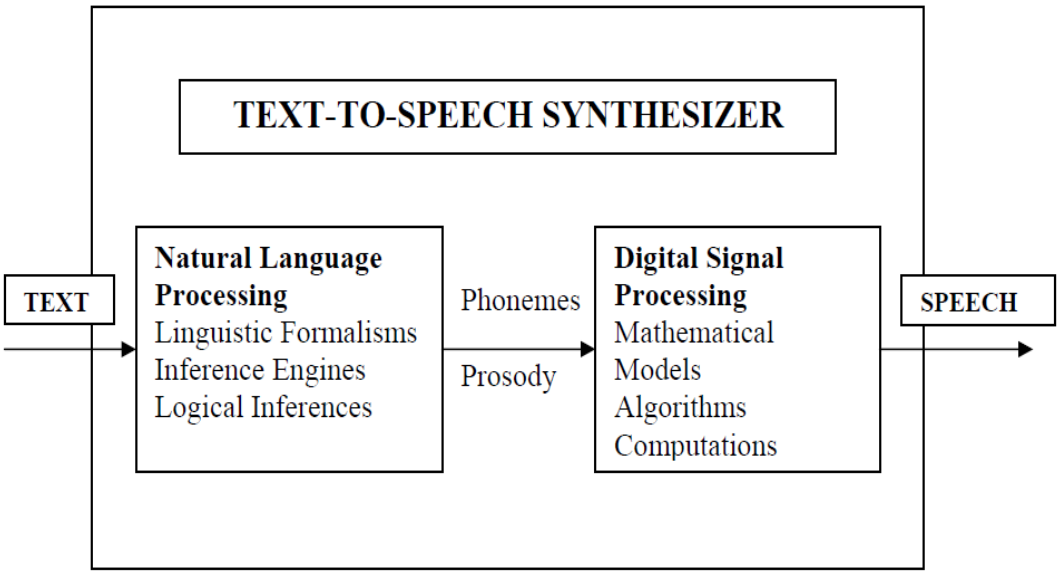


Figure 2.1 Basic Components of TTS Synthesiser [7]

It is further classified into text analysis, automatic phonetization, and prosody and dsp component.

2.1.1 TEXT ANALYSIS

Text Analysis phase can be further divided into three modules.

2.1.1.1 PRE-PROCESSING

It identifies numbers, abbreviations and acronyms, then if required performs transformation into the full text. Acronyms can be identified as a sequence of capital letters.

- "OOPS" will be changed into "Object Oriented Programming Structure".
- "9876543210" will be changed into ninety eight seventy six fifty four thirty two and ten

2.1.1.2 MORPHOLOGICAL ANALYSIS

The task of Morphological Analysis module is to identifies, analyze and understand the structure of word [14]. It assigns part of speech category to each individual word .It consist of three phases that is inflection, derivation and compounding.

Inflection consists of systematic modification of root forms by adding prefixes and suffixes. For example,'s' in dog, it will be changed into dogs that is plural form. Derivation means change in syntactic which results in change in meaning. For example, there is the difference in teach and teacher syntactically as well as in the meaning. Compounding is the creation of new word from the old word. For example, boathouse is the compound word which is created by two individual morphemes 'boat' and 'house'.

2.1.1.3 CONTEXTUAL ANALYSIS

The contextual analysis module allows reducing the possible part of speech category to a very restricted number if it is given that the corresponding possible parts of speech of neighboring words [14].

- **TECHNIQUE TO DETERMINE CONTEXTUAL FACTORS**

  N-Gram predicts the next word or syllable or phoneme [14]. It is a stochastic problem, contextual factors uses classification of the previous words, (the history of previous word) to predict the next word. It is a type of probalisitic language model for predicting the next item in the form of (n-1) th markov

process. N-gram of size 1 is called as unigram, n-gram of size 2 is called as bi-gram and size of 3 is called as tri-gram. *E.g.* "the dog smelled like a skunk", the trigram has would be: "# the dog", "the dog smelled", "dog smelled like", "and smelled like a "," like a skunk "and" a skunk #". A description of contextual information about syllables, words, phrases, and utterances [12]. These are

- Phoneme
  - Preceding, current, succeeding phonemes
  - Position of current phoneme in current syllable.
- Syllable
  - Whether or not preceding, current, succeeding syllables are stressed
  - Number of phonemes in preceding, current, succeeding syllables
  - Position of current syllable in current word
  - Number of stressed syllables in current phrase before, after current syllable
  - Number of syllables, counting from previous stressed to, current syllable in the utterance
  - Number of syllables, counting from current to next Stressed syllable in the utterance
- Word
  - Part-of-speech category of preceding, current, succeeding words
  - Number of syllables in preceding, current, succeeding words
  - Position of current word in current phrase
  - Number of content words in current phrase before and after current word
  - Number of words counting from previous content word
  - To current word in the utterance number of words counting from current to next content
  - Word in the utterance
- Phrase
  - Number of words in preceding, current, succeeding phrases
  - Position of current phrase in current utterance
- Utterance

- Number of syllables, words, phrases in the utterance

2.1.2 AUTOMATIC PHONETIZATION (L to S)

Automatic Phonetization module is used for automatic determination of the phonetic transcription of the orthographic representation of text [7]. It is also called as grapheme to phoneme conversion.

There are two ways to store the phonemes

2.1.2.1 DICTIONARY-BASED SOLUTIONS

It stores phonemes in the database

For example-k-/k/, a-/a/

There are various issues while aligning letter to phoneme. It is not easy to store each word or phonemes in the lexicon. Therefore, another approach should be followed for assigning letter to phonemes. With the help of rule based approach a letter is assigned to phonemes which are not available in the lexicon.

2.1.2.2 RULE-BASED TRANSCRIPTION SYSTEMS

It transfer most of the phonological rules of dictionaries into `a set of letter-to-sound (or grapheme-to - phoneme) rules. Words that are pronounced in a different way that constitute a rule on their own are stored in an exceptions dictionary. These rules are in the form of context-sensitive rewrite rules of the form A/X/B->y, meaning that letter X is rewritten as phoneme y, when X occurs in the context of letters A and B. Another example to illustrate this is “rough” will be pronounced as /r u f/. Rule based transcription system is the most reliable method. Rule based approach is a trainable method for generating letter to sound rules, for producing the pronunciation of out-of-vocabulary words [4]. Several approaches have been adopted over the years for grapheme to-phoneme conversion, such as hand-seeded rules, finite state transducers, neural networks, HMMs etc Their approach is based on a semi-automatically lexicon, from which we derived rules for automatic transcription.

2.1.2.3 ALGORITHMS USED FOR LETTER TO SOUND (L TO S)

- **DECISION TREE**

  It is top-down approach. It is the technique where the rule is derived after observing the database [3]. One must use a set of possible yes-no questions .It

is also called as classification and regression tree. Each interior node corresponds to the input variables [24]. There are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf. A tree can be "learned" by splitting the source set into subsets based on an attribute value test.

- **HMM (Hidden Markov Model)**

  It consists of finite set of states each of which is associated with the probability distribution. Only the outcome is visible to an external observer and therefore the states are 'hidden' to outsiders. There are three model parameters for HMM. There are finite numbers, say N, of states in HMM [13]. At each time t, a new state is entered based on the transition probability distribution which depends on previous state. After each transition, an observation of output symbol depends on the current state. If the most probable sequence of phonemes is given *i.e* the input grapheme sequence, then output phoneme sequence can be determined through HMM[24] .Using $G =< g_1; g_2; :::; g_N >$ represents the sequence of graphemes, and $S =< s_1; s_2; :::; s_M >$ represents the hidden sequence of phonemes, the problem in the standard HMM could be formulated as $S = \text{argmax}_S P(G|S)P(S)$ where P(S) is the prior probability of a sequence of phonemes occurring and P(G|S) is the likelihood of grapheme sequence G given phoneme sequence S[18].HMM can be divided into two phases. First phase is the training phase, in this phase, we have given the observation sequence $O=O_1, O_2,.....,OT$ how we choose a state sequence $l=i_1,i_2,...,iT$ in some meaningful way. So to solve it, we have viterbi algorithm. In viterbi algorithm, there are different steps, these are

  Initialisation

  $\delta_i(t)= \pi_i b_j(O_1)$

  Recursion

  For $2<=t<=T \quad 1<=j<=N$

  $\delta_t(j)=\max(\delta_{t-1}(i)\, a_{ij})b_j(O_t)$

  $\Psi_t(j)=\text{argmax}[\delta_{t-1}(i)a_{ij}]$

  The state sequence which maximizes the probability of seeing the observation to time t-1, landing in state j,and seeing the observation

Termination

$P^*=\max[\delta_t(j)]$ P* gives the state-optimised probability

$I_t^*=\text{argmax}[\delta_t(i)]$

Path (State sequence) Backtracking

For t=T-1,T-2,…1

$I_t^*=\Psi_{t+1}(i_{t+1}^*)$

The second problem is that how do we adjust the model parameters that are A, B and π. Through Baum-Welch Algorithm, the model parameters can be determined for each word model during training period.

$\hat{\pi} = \gamma_1(i)$ The expected frequency of state *i* at time *t=1*

$\hat{a}_{ij} = \frac{\sum \xi_t(i,j)}{\sum \gamma_t(i)}$ Ratio of expected no. of transitions from state *i* to *j* over expected no. of transitions from state *i*

$\hat{b}_j(k) = \frac{\sum_{t,o_t=k} \gamma_t(j)}{\sum \gamma_t(j)}$ Ratio of expected no. of times in state *j* observing symbol *k* over expected no. of times in state *j*

The next problem is phoneme matching. If observation sequence $O=O_1, O_2,\ldots O_T$ and the model parameters A,B and π. Then how to recognise the each word based on the given observation sequence.

Through Forward and Backward algorithm, it can be achieved .

$$\alpha_t(i) = P(o_1,\ldots,o_t \mid q_t = i,\lambda)$$

Where $\alpha_t(i)$ -- probability of observing a partial sequence of observables $o_1,\ldots o_t$ such that at time t, state $q_t=i$. The HMMs emit context-sensitive discrete observations and are used with a grapheme-to-phoneme conversion system [15].

2.1.3 PROSODY

Prosody helps to determine gender, age, emotions and other features and with the help of it a very natural sound can be produced.

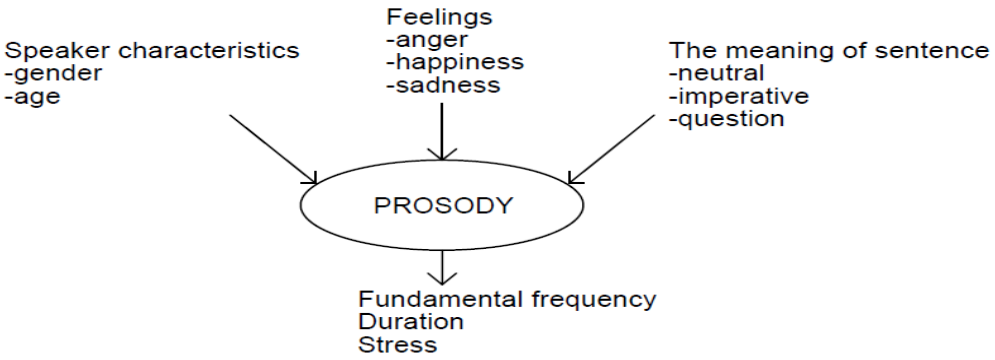


Figure 2.2 Prosodic Dependencies [8]

- **Pitch**

  Pitch or fundamental frequency is highly influenced by gender, age and mood of speaker. Pitch level also changes the meaning of sentence [8].For example, "I will take lunch in afternoon" and "will you take lunch in afternoon?" both sentences have different pitch levels. Pitch also changes while starting and end of sentence or word. For example, "Shyam" and "Keshav" in both the words, pitch level of "sh" are different.

- **Duration**

  Duration or time characteristics can be investigated at several levels from phoneme level to sentence level timings, speaking rate and rhythm. With the help of duration which word should be spoken fast or slowly. Every syllable or phoneme is dependent on previous syllable or phoneme. Duration helps to decide the previous phoneme should be spoken slowly.

- **Intensity**

  Intensity can also be defined as loudness within speech. The intensity of a voiced sound goes up in proportion to fundamental frequency [9]. At syllable level, vowels are usually more intense than consonants and at a phrase level, **s**yllables at the end of an utterance can become weaker in intensity [8].

2.1.4 THE DSP COMPONENT

It is also called synthesizer component. It works in the back-end. It can be also divided into three basically rule based formant synthesis, articulatory synthesis and formant synthesis.

2.1.4.1 SOURCE FILTER THEORY

In the source filter theory, the vocal tract can be modelled as a linear filter that varies over time [22]. The filter (i.e. a set of resonators) is excited by a source, which can be either a simulation of vocal cord vibration for voicing, or a noise that simulates a constriction somewhere in the vocal tract. The sound wave is created in the vocal tract, then radiates trough the lips.

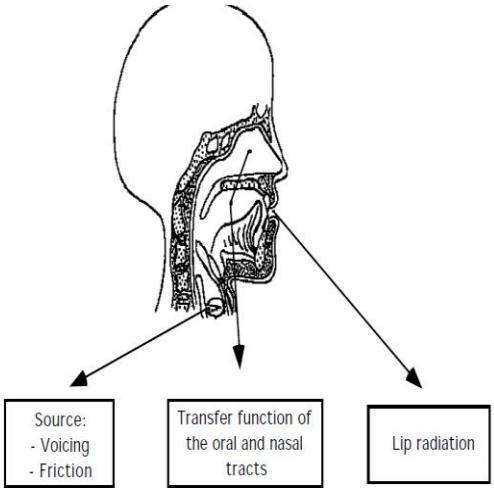


Figure 2.3 Source Filter Model *[10]*

This model is divided into three separate parts, namely the source (voicing and friction), the filter (implementing the transfer function of oral and nasal tract), and lip radiation

2.1.4.2 FORMANT SYNTHESIS

Formants are the acoustic resonance in the human vocal tract. It can be said as the vibrations while production of voice. It models the real part of the frequency or transfer function based on the source filter theory. The rule based formant synthesis uses set of parameter necessary to synthesize a desired utterance using a formant synthesizer [10].

Formant Synthesizer are classified into two

- **CASCADE FORMANT SYNTHESIS**

  The output of one resonator is applied to the input of another formant**.** Cascade Formant Synthesiser works serially. It does not work in parallel. It consists of band pass resonators connected in series and the output of each resonator is applied to the input of another resonator. It is found better for

non-nasal voiced sounds and it needs less information. Therefore it is easy to implement.

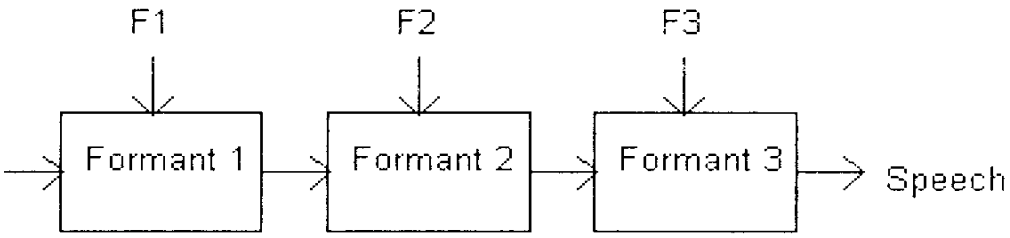


Figure 2.4 This figure illustrates the cascade formant synthesis. Here F1, F2, F3 are the frequencies. [10]

- **PARALLEL FORMANT SYNTHESIS**

  In Parallel Formant Synthesis which excitation is applied to all the formants in parallel and there outputs are summed.

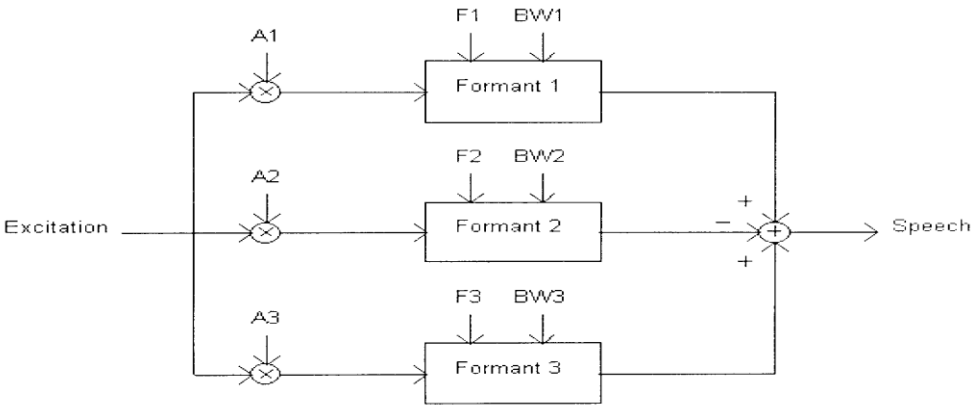


Figure 2.5 illustrates the parallel formant synthesis. Here F_1, F_2, F_3 are the frequencies and BW_1, BW_2, BW_3 are the bandwidth [10]

Formant-synthesized speech can be reliably intelligible. It does not have any database of speech samples. These are advantages of formant synthesis. But the speech is artificial and robotic.

2.1.4.3 ARTICULATORY SYNTHESIS

Articulatory speech synthesis models the natural speech production system as accurately as possible [16]. This is accomplished by creating a synthetic model of human vocal tract and making it speak. Human vocal organs are potentially the most

satisfying model to model the high quality synthetic speech [10]. It models the human articulators (like tongue, lip or jaw etc) and vocal cords. The articulators are usually modeled with the set of area functions between glottis and mouth. The shape of the vocal tract can be controlled in a number of ways which usually involves modifying the position of the speech articulators. Speech is created by digitally simulating the flow of air through the representation of the vocal tract

In articulatory synthesis, the vocal tract model allows accurate modeling of transients due to abrupt area changes. But the data is always 2D whereas the real vocal tract is 3D.

2.1.4.4 CONCATENATIVE SYNTHESIS

Concatenative synthesis consists of pre-recorded samples of real speech which are smoothly combined to create an arbitrary synthetic utterance [17]. Concatenation techniques take small units of speech i.e. waveform data, and concatenate sequences of these small units together to produce waveforms. Concatenation systems are concerned with the selection of appropriate units and the algorithms that join those units together. TTS system designers need to make decisions about the size of the concatenative units. The discontinuities in the concatenation can cause distortion in the signal. It is less flexible and also it can imitate the specific speaker with only one voice quality. Another constraint is that the very vast storage of the pre-recorded units. While the speech sounds are more natural in concatenative synthesis.

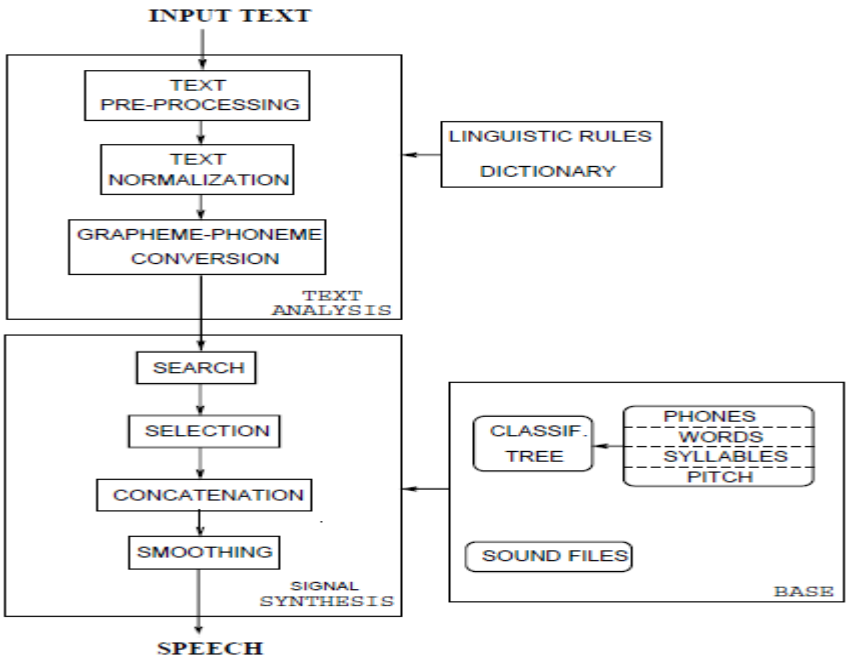


Figure 2.6 Concatenative Syntheses [17]

Prudon *et al.* [2001] developed the text to speech synthesis using concatenative approach for French Language. They have divided text to speech synthesis in three parts. In the first part, database design and annotation methods are presented. In a second part, the system architecture is described. A comparison is described in the third part. Diphone based concatenation is done and synthesis by rules of the prosody. The most significant improvements brought by the new system seem to be for voice pleasantness and overall impression.

Concatenative Synthesis can be further divided into three parts

- **Unit selection synthesis**:

  This type of synthesis uses large speech databases consisting of various units of speech. The division into segments is done using a specially modified speech editor where the speech signal is presented in the form of waveform and spectrogram. In this signal excluding noise is determined. Then these signals are concatenated to produce the desired word.

- **Diphone synthesis**

  It uses a minimal speech database containing all the diphones (is the last half of one phone followed by the first half of the next one) occurring in a given language. Diphone synthesis suffers from the articulation effect of concatenative synthesis and robotic sound of formant synthesis, and but diphone synthesis consist of small database.

- **Domain-specific synthesis**

  It concatenates pre-recorded words and phrases to create sound. It is used in applications which are limited to a particular domain, like trains schedule announcements or weather reports. This technology is very simple to implement, and has been in commercial use for a long time. This is the technology used by gadgets like talking clocks and calculators.

2.1.4.5 HMM BASED SPEECH SYNTHESIS (HTS)

HTS is the statistical model which can be used for modelling the speech parameters extracted from a speech database [23], and then generating the parameters according to text input for creating the speech waveform. HMM-based speech synthesis systems

are able to produce speech in different speaking styles with different emotions. It has better adaptability and clearly small memory requirement.

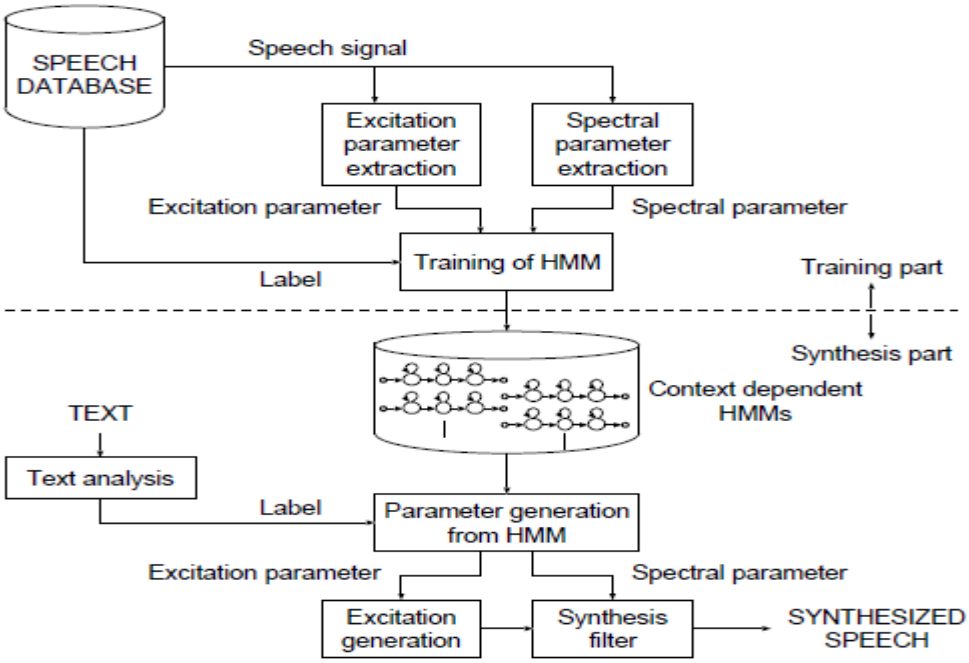


Figure 2.7 HMM Based Speech Synthesis [23]

However, the HMM-based TTS systems often suffer from degraded naturalness in quality compared to concatenative based speech synthesizers. In this spectral frequency and fundamental frequency are modelled and used to generate the speech waveform.

2.2 UNITS OF SPEECH

- **Word**

  It can be said as the sequence of the characters. Each letter is related to the phone. Thus, words is spoken as a sequence of phones. ”cut” will be spoken as /k/ /a/ /t/.The advantage of using words is that concatenating the words in relatively easy then concatenating the sub words. But it needs a large storage in the database and all the words cannot be stored since it is unlimited.

- **Syllable**

  A syllable is made of syllable nucleus (which is vowel) with optional initial and final margins (typically consonants).The consonant on the left side is

called as onset while on the right side is called as coda. Syllable can be further divided into

- Monosyllable is composed of only one unit. E.g.-dog(/dog/),
- Demisyllable is composed of two units. E.g.-monkey(mon key)(/mon//key/)
- Trisyllable is composed of three units. E.g.-indigent(in di gent)(/en//de//gent/)

Duan *et al* [2010] discussed that syllables provide better results than phonemes. Syllables are required because syllables are longer than phoneme. Therefore there will be less unit co-articulation effects (Unit co-articulation is the coupling effect when we join /k/ and /a/, then it will produce /ka/).Syllables are smaller than words, it take less memory.

- **Phonemes**

  Phonemes are the basic unit of speech. It is most commonly used units in speech synthesis. Phonemes can be further divided into diphones and triphones.

- **Di-phones**

  It is the last half of one phone followed by the first half of the next one. For example, #-t, t-e, e-s, s-t and t-#. Wolfer [1997] discussed diphone based speech synthesis. He basically divided speech synthesis into two parts. In the first part, he determined phonetic transcription for each orthographic transcription. Secondly, he concatenated the recorded speech units that are diphones. In this thesis, he told orthographic transcription to phonetic transcription on the small amount of data. He concluded that when memory is less than decision trees are used. Decision trees can be trained easily and do not require an extensive research to determine the phonemes.

- **Triphones**

  It consists of one phoneme in between the half phoneme. For example-(half phoneme-one phoneme-half phoneme).

In the next section there will be discussion about different Indian Synthesiser so far developed.

2.3 TTS SYNTHESISER FOR INDIAN LANGUAGES

In this there will be discussion on some text to speech synthesiser developed for Indian Languages.

- **VANI FRAMEWORK**

  Vani is a TTS synthesiser developed primarily for Indian Languages. This is the type of synthesiser [14]. It basically works on the concatenation technique with phonemes as the basic unit. The aim of vani is to express every emotion. Vani is built using java to enable platform independence and uses Java sound API (JSAP).

- **DHVANI FRAMEWORK**

  Dhvani is the project started by Dr. Ramesh Hariharan at IISC, Bangalore. It is language independent design which is based on phoneme concatenation technology. It is developed in C/Linux based system. It is developed for various Indian languages like Hindi, Kannada and Malyalam. It starts as a simputer project.

- **SHRUTI TTS**

  SHRUTI synthesizer works for two Indian languages namely Hindi and Bengali and produces natural audio output. The synthesizer works for the Microsoft Pocket PC and other handheld devices. Two versions of the synthesizer had been built, one which resides on the system memory and another which runs on a storage card.

- **MATRUBHASHA API**

  Matrubhasha is a project carried out at C-DAC Bangalore. It was made because this application can speak and listen to the masses in any Indian language. 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.

- **C-DAC, NOIDA**

  C-DAC Noida developed a Text to Speech Synthesiser for Hindi Language. It is based on the concatenative approach . The input in Unicode is processed by the Text processing unit and the speech-processing unit generates synthesized speech. It is developed for the handheld devices and other PDA's. It uses phonemes as the basic unit for synthesis.

2.4 DIFFERENT RESEARCHES

Black *et al* [2006] discussed about their handheld two-way speech translation system for English and Iraqi. The computation and memory limitations on the handheld impose constraints on the ASR and TTS components. In this paper they discussed their approaches to optimize these components for the handheld device and present the results from the evaluation that was a part of the project.

Balyan *et al* [2011] discussed the development of unit selection for Hindi language. Phoneme has been chosen as the basic unit as larger domain since with syllables cannot be created. They discussed various issues of speech segmentation using Hidden Markov Model (HMM) based technique. They report the comparison of automatically segmented labels (speech units) using base line model of HMM with the manually segmented labels in the context of Hindi Speech.

Shirbahadurkar *et al* [2009] discussed in the paper about concatenative text-to speech system and discuss the issues relevant to the development of a Marathi speech synthesizer using different choice of units like words, phonemes. 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. They synthesize the Marathi text and perform the subjective evaluations of the synthesized speech.

2.5. PUNJABI AND ENGLISH PHONLOGY

Phonemes is the representative of the class of sounds, which speaker accepts as a single unit, regardless of the position variants .Phonemes can be identified by the minimal pair test technique. A minimal pair test is quick and direct way of establishing that two sounds belong to separate phonemes in the language [20]. Phoneme is the functional unit and is represented by two slanted lines (//)

Phonemes can be further divided into segmental and super segmental phonemes. Punjabi consonants and vowels are considered as segmental phonemes. Super segmental phonemes do not have independent existence. Super Segmental Phonemes are further divided into stress, intonation, juncture, nasality and tone. Phonemes can be further divided into segmental and super segmental phonemes. Punjabi consonants and vowels are considered as segmental phonemes. Super segmental phonemes do not have independent existence. Super Segmental Phonemes are further divided into stress, intonation, juncture, nasality and tone. It consist of 20 punjabi vowels, 10 non-nasalised vowels and five are nasalised and remaining 33 are non-nasalised in consonants.

2.5.1 PUNJABI VOWELS

In Punjabi language there are total twenty vowels out of which ten are non-nasalised (ਆ, ਓ, ਐ, ਇ, ਈ, ਏ, ਐ, ਅ, ਉ, ਊ) and ten are nasalized vowels which are as follows

(ਓਂ, ਉਂ, ਊਂ, ਐਂ, ਆਂ, ਅਂ, ਓਂ, ਈਂ, ਏਂ)[20].

The vowels are classified on the basis of position of the physical organs. The detailed discussion is as follows

- **Position of tongue**

  When the front part of tongue is lifted towards hard palate, unobstructed air stream produces sound called the front vowels and when it is lifted towards the soft palate, back vowels are produced [10]. Third type of vowels is called as the central vowels when the central part of tongue is lifted towards the hard palate. For example

Front vowels	**Back Vowels**	**Central Vowels**
ਓ	ਊ	ਇ.

Table 2.1 Classification of Vowels based on Position of Tongue

- **Height of Tongue Tip**

  During utterance, height of tongue tip measures the degree obstruction in the path of air stream and produces the distinguishing sound [10]. These are classified as

the high, mid-high, middle, mid-low, low vowels. Classification of vowels based on height of tongue tip is provided in the next page.

High Vowel	Middle High Vowel	Middle Vowels	Middle Low Vowels	Low Vowels
ਊ, ਈ	ਇ, ਉ	ਏ, ਅ, ਓ	ਅੈ, ਅੈ	ਆ

Table 2.2 Classification of Vowels based on height of tongue tip

- **Shape of lips**

  This category can be classified according to the shape of the lips. These are called as rounded vowels and un-rounded vowels.

Rounded Vowels	Un-rounded Vowels
ਉ, ਊ, ਓ, ਅੌ	ਇ

Table 2.3 Classification of Vowels based on shape of lips

2.5.2 PUNJABI CONSONANTS

Consonants are the sounds produced by an obstruction of blocking or some other restriction for the free passage of the air, exhaled from the lungs, through the oral cavity. Out of the 38 consonants in Punjabi [20] five are nasalized (ਢ, ਨ, ਮ, ਙ, ਣ) and remaining 33 are non- nasalized.

In the next page, there is discussion about English phonology. English Phonology consists of 21 consonants and 5 vowels.

ਸ Sussa *Sa*	ਹ Haha *Ha*	ਕ Kukka *Ka*	ਖ Khukha *Kha*	ਗ Gugga *Ga*	ਘ Ghugga *Gha*
ਙ Ungga *Nga*	ਚ Chucha *Ca*	ਛ Chhuchha *Cha*	ਜ Jujja *Ja*	ਝ Jhujja *Jha*	ਞ Yanza *Nya*
ਟ Tainka *Tta*	ਠ Thutha *Ttha*	ਡ Dudda *Dda*	ਢ Dhudda *Ddha*	ਣ Nahnha *Nna*	ਤ Tutta *Ta*
ਥ Thutha *Tha*	ਦ Duda *Da*	ਧ Dhuda *Dha*	ਨ Nunna *Na*	ਪ Puppa *Pa*	ਫ Phupha *Pha*
ਬ Bubba *Ba*	ਭ Bhubba *Bha*	ਮ Mumma *Ma*	ਯ Yaiyya *Ya*	ਰ Rara *Ra*	ਲ Lulla *La*
ਵ Vava *Va*	ੜ Rahrha *Rra*				

Figure 2.8 Basic Punjabi Consonants [5]

2.5.3 ENGLISH PHONOLOGY

English Phonology consists of 26 alphabets having 5 vowels and 21 consonants.

- List of English Vowels

A	E	I	O	U

Figure 2.9 Basic English Vowels [5]

- List of English Consonants

B	C	D	F	G	H	J	K	L	M	N
P	Q	R	S	T	V	W	X	Y	Z	

Figure 2.10 Basic English Consonants [5]

CHAPTER 3

PROBLEM AND PROPOSED SOLUTION

3.1 PROBLEM STATEMENT

Speech is important means of communication between people and now a day's researchers are working hard to make a reliable communication between man and machine with the use of one of the technique known as TTS synthesis. This thesis is focussed on development of text to speech synthesis for a mobile device application in Punjabi language. The application aims for uttering name of caller which is in English into Punjabi language after mapping. Major benefit of this application is preferably for those people who are visually impaired, can understand their regional language (i.e. Punjabi) and for those who understand spoken Punjabi but are illiterate. Moreover, integration of this application with mobile devices can be helpful in everyday life. A person can listen the text instead of reading caller id name or messages *etc.*

To start with, a survey was conducted on 50 mobile phones to study how users store names in phone book. One can store caller name in different ways. A person can write a caller name as Mr. Sharma, Thapar Mr.Arti, Nidhi Sharma, Nidhi, Gaurav (Mithu), Meenu Thapar1, Sheenam c.s.e, Chetan kotwal, avinash@com, X31jan, 100, 101, himani2, Honey2 Banglore, Lucky Friend, RahulMittal0987654321, M1, M2, Papa .

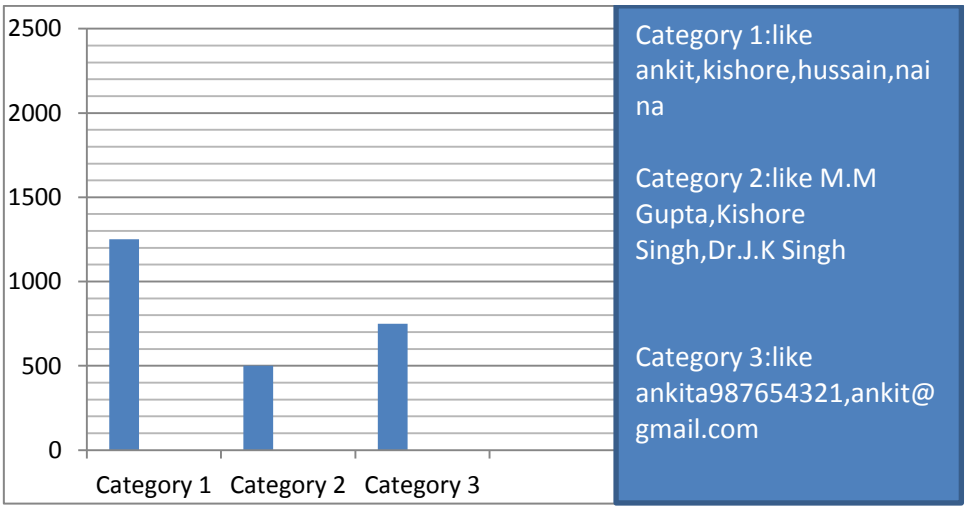


Figure 3.1 Categorization of Names Stored in Phone Book

Out of these 70% of names falls in generic category (*For e.g.*, Ankit, Mr.M.K.Jain, M.M Suri) and rest lie in the non-standard or non-generic category (For *e.g.*, Ankita31jan, amita@gmail.com). After that analysis, most common and generic patterns are chosen and are used for the implementation.

Since the application tries to map English tokens to Punjabi phonemes, there are some problems that arose during this token mapping process. In order to resolve the issues, a technique/ algorithm need to be devised along with the pre-processing phase.

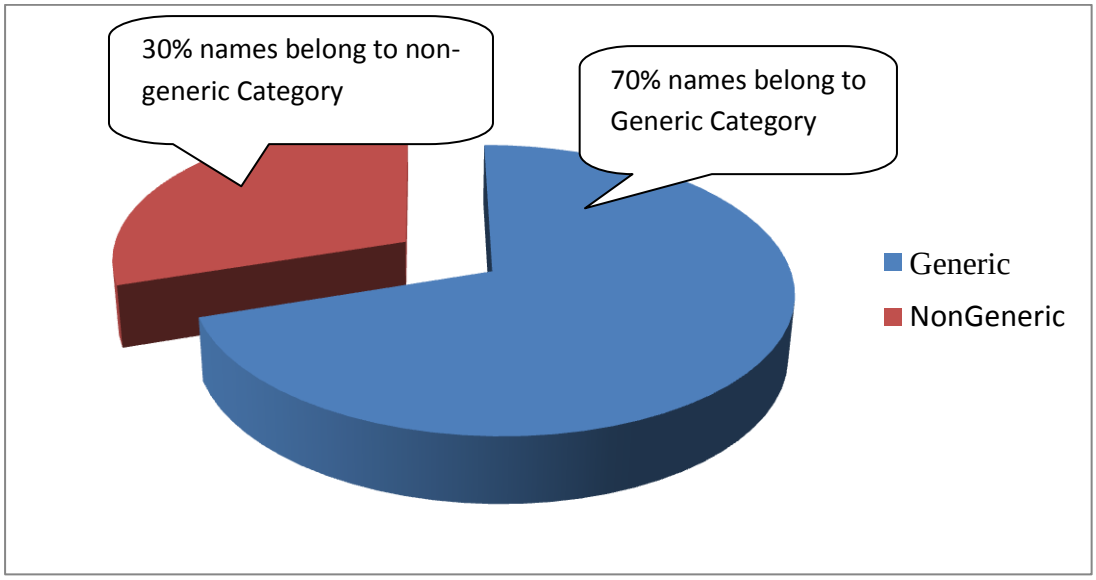


Figure 3.2 Caller's Id Names in Mobile Phone Directory

Since the application tries to map English tokens to Punjabi phonemes, there are some problems that arose during this token mapping process. In order to resolve the issues, a technique/ algorithm need to be devised along with the pre-processing phase

From literature review numerous approaches have been discussed for speech synthesis of various applications. Every technique has its own implementation methodology with some advantages and disadvantages. Articulatory synthesis model requires human vocal tract like design in physical and produce speech with articulators (*i.e* tongue, tip and jaws etc). But data is always 2D whereas the real vocal tract is 3D. So it is difficult to model this kind of synthesis. Formant Synthesis is rule based synthesis (it uses certain parameters to generate waveforms) but voice is highly robotic *i.e* not natural. HTS (HMM-based Text to Speech synthesis) is a statistical model which can

be used for modelling the speech but the speech that is generated is not natural. Concatenative Synthesis approach selects the units (phoneme, syllable and words) and joins them to synthesize into the waveform. It majorly depends upon a large database to generate the waveforms. After studying all the techniques in detail it is proposed to use two techniques in mixed mode. First is HMM-based TTS and second is Concatenative synthesis.

3.2 PROPOSED SOLUTION

Text to Speech Synthesis includes a number of stages, as discussed in chapter 1. Based on the stages the proposed work has been discussed in three sections as pre processing, mapping of English text to Punjabi phonemes and Concatenative Synthesis which will be discussed in later sections. The pre processing section deals with determination of titles and initials within the name. The section pertaining to English text to Punjabi phonemes needs knowledge of Punjabi keywords and their pronunciation while mapping English string to Punjabi phonemes. In the Concatenative Synthesis section the phonemes are concatenated based on their selection to produce the speech sound.

3.3 PREREQUISITES

- This application is developed on Visual Studio 2008 with C# as front end and Sql Server 2008 standard as the back end. Microsoft Visual Studio is an integrated development environment (IDE) (it is a software application that provides facilities to computer programmer for software development. It consist of code editor, a build automation tools and a debugger) from Microsoft.
- Power Sound Editor is used for recording sound. It visually edits the audio files. It supports various format for sound files like .wav,.mp3.Recording the wave files is done with the help of mike, speakers or headphones
- Punjabi Raavi Unicode Font.
- Personal computer with atleast windows XP, 80 GB Hard Disk, 512 MB Ram and 2 GHz processor.

  Mobile Device
- Android/ Windows / Iphone Development environment with necessary hardware / software.

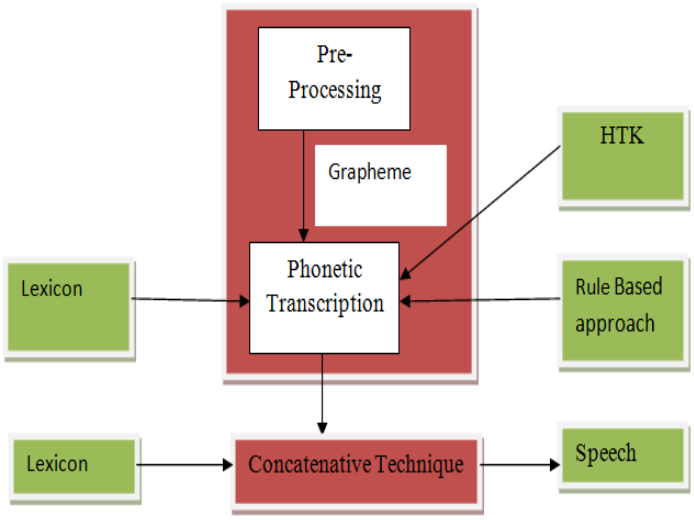


Figure 3.3Text to Speech Synthesiser

3.4 SPEECH UNIT SELECTION AND DATA PREPARATION

This section discuss mainly about the preparation of Speech database. There will be discussion about the selection of unit for Punjabi database and recording of syllables / phonemes.

3.4.1 SPEECH UNIT SELECTION

For the development of TTS System, Punjabi syllables and phonemes were selected for concatenation. Phonemes are the basic unit for synthesis. Syllables are a unit of speech made up of one or more phonemes and in turn one or more syllables make up a word. A syllable can be made of more than one character. On the basis of survey, syllable is considered to be using various rules C, V and CV where C is consonant, V is vowel and CV is consonant and vowel.

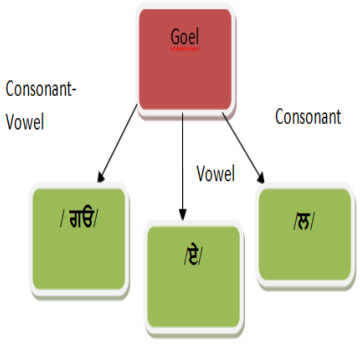


Figure 3.4 Speech Unit

Another Example is /divya/ where / di /is Consonant-Vowel combination and /v/ is C and /ya/ is the Consonant-Vowel Combination.

3.4.2 SYLLABLE/PHONEMES RECORDING

Recording of syllables/phonemes are done individually with the following characteristics

- Sampling Rate 8000KHz
- Bit Depth 16bits
- Channels Mono
- File Extension Wav

These standard characteristics are used in order to minimize the file size and to make the file compatible with most of the system.

3.5 DATABASE DESIGN

After detailed study it was observed that the database of Text to Speech Synthesiser for Mobile application must consist of at least eleven tables. The description of Tables is as follows:-

I. Table Name: TITLES

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
Key_code	Varchar	3	Not Null	It acts as a primary key
Englishtitles	Varchar	10	Not Null	This field contains englishtitles (Mr.,Ms.,*etc*)
Pathfortitles	Varchar	50	Not Null	This field contains path for wave files.
Fulltitle	Varchar	20	Not Null	This field contains full titles(*e.g.*,Mister *etc)*

Table 3.1 Titles

II) Table Name: INITIALS

Attribute with properties are given below

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
U_id	Varchar	2	Not Null	It acts as a primary key
EnglishTokens	Varchar	2	Not Null	This field will contain English chars with periods (“.”)
PathforInitials	Varchar	50	Not Null	This field contains path for wave files.

Table 3.2 Initials

III) Table Name: PHONEMES

Attribute with properties are provided here

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
U_id	Varchar	2	Not Null	It acts as a primary key
EnglishTokens	Varchar	4	Not Null	This field will contain English tokens like (a ,b, c, d, *etc*)
PunjabiTokens	Varchar	2	Not Null	This field will contain Punjabi tokens

Table 3.3 Phonemes

IV) Table Name: SYLLABLE

Description of attribute and properties are given below

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
U_id	Varchar	3	Not Null	It acts as a primary key
PunjabiSyllables	Varchar	3	Not Null	This field will contain Punjabi Syllables
PathforWaveFiles	Varchar	50	Not Null	This field contains physical path for wave file

Table 3.4 Syllables

V) Table Name: UNKNOWN STRING

Description of attribute and properties are given below

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
U_id	Varchar	1	Not Null	It acts as a primary key
UnknownString	Varchar	1	Not Null	This field contains string like(@,#,$ *etc*)

Table 3.5 Unknown String

VI) Table Name: ENGLISHTOKENS

Description of attribute and properties are given below

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
U_id	Varchar	2	Not Null	It acts as a primary key
EnglishTokens	Varchar	2	Not Null	This field will contain English tokens
Consonant/Vowel	Varchar	1	Not Null	Contains value" 2" for consonant and "1" for vowel
Nasalisation	Varchar	1	Not Null	Contains value "yes" for nasalised and "no" for non-nasalised

Table 3.6 English Tokens

VII) Table Name: SPECIAL TOKEN

Attribute with properties are given below

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
U_id	Varchar	1	Not Null	It acts as a primary key
SpecialToken	Varchar	1	Not Null	This field will contain tokens like(" ",".")
OtherCharacter	Varchar	1	Not Null	Value "3" is given to char

Table 3.7 Special Tokens

VIII) Table Name: PUNJABITOKENS

Description of attribute and properties are given below

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
U_id	Varchar	2	Not Null	It acts as a primary key
PunjabiTokens	Varchar	2	Not Null	This field will contain Punjabitokens
Consonant/Vowel	Varchar	1	Not Null	Contains value" 2" for consonant and "1" for vowel
Nasalisation	Varchar	1	Not Null	Contains value "yes" for nasalised and "no" for non-nasalised

Table 3.8 PunjabiTokens

IX) Table Name: HTKTOKENS

Description of attribute and properties are given below

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
U_id	Varchar	3	Not Null	It acts as a primary key
HTKTokens	Varchar	1	Not Null	This field will contain HTK Tokens
PunjabiTokens	Varchar	1	Not Null	This field will contain punjabitokens

Table 3.9 HTKTokens

X) Table Name: NUMERALS

Description of attribute and properties are given below

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
Key_code	Varchar	2	Not Null	It acts as a primary key
Integer	Varchar	1	Not Null	This field contains integers
Path	Varchar	50	Not Null	This field will contain path for wave file
Punjabi Numerals	Varchar	1	Not Null	This field contains punjabi integers

Table 3.10 Numerals

XI) Table Name: MOBILECODE

Description of attribute and properties are given below

Field Name	**Data Type**	**Size**	**Null/Not Null**	**Description**
Key_code	Varchar	05	Not Null	It acts as a primary key
CountryCode	Numeric	10	Not Null	This field contains code for region
CityCode	Numeric	10	Not Null	This field contains code for city
Path	Varchar	50	Not Null	This field will contain path for wave file

Table 3.11 MobileCode

3.6 PRE-PROCESSING

In Pre-Processing, abbreviations, acronyms, titles and numerals are determined.

In this application, Pre- processing module is divided into five parts.

- When a caller calls some person, caller's name is sent to the synthesiser as input string. That string will be checked for English alphabets i.e whether it consist of non alphabet characters like (@, #, $, %, *, *etc*). If the string contains non alphabets then the synthesiser will consider the string as non-generic and discard them will not produce any sound.
- In the next module application will check the input string for mobile number *i.e* weather the string contains phone number 9876543210 (this case will be true when the name is not stored against that number i.e unknown contacts). In this case the system will produce the sound of those numerals one by one.
- It will check for titles in the string (like "Shri.", "Mr.", "Ms.", etc). If title is present in the database then it is stored in new substring and rest of the string is stored in newstring otherwise oldstring will be stored in newstring. This is done by function tiles (oldstring).
- In the next part, it is checked whether a character is consonant or vowel. After getting the newstring, it is parsed into a list of characters. Then, that character is checked whether that character is consonant or vowel or other character (like " ","."") and assign string "1" for vowel,"2" for consonant and "3" for other character. After that , characters are concatenated like "chh", "bh", "kh", "ch", "ai", "oo", "au", "aa", "ee", "gh", "dh", "sh", "th", "jh", "ou" and stored in a list called englishcharacterslist. This is done by function Englishcharacterset (newstring)
- Finally in the third part it is checked whether character is initial or not. For Example, B.K.Saini, Y A Sharma, Y Singh. This is done by initials (Englishcharacterset)

Phase 1: Pre-processing

Pre-processing has following input and output parameters and below is the list of functions defined in pre-processing.

Input Parameters

English string (caller name), a string of size 20 is considered.

Output Parameters

English character_set without titles, initials and numerals

List of functions used in pre-processing modules

Call numerals (oldstring)

//this function checks whether string is telephone number and store it in another list otherwise string will remain same.

Call identify_titles (oldstring)

//this function returns a newstring without titles and if title is present then it is stored in a newsubstring

Call consonantandvowel (newstring)

//this function returns a list of characters of English with various combinations.

Call initials (Englishcharacters)

//this function returns a list of characters in English without titles

Function1: Identify numerals in a list

INPUT Telephone Number

OUTPUT TelephonenumberList

STEPS

1. It will identify numerals in the string.
2. If it is present then store it in another list named telephone_number
3. Utter the number.

Function2: Identify English titles in a list

INPUT English Name

OUTPUT Name without titles

VARIABLES

1. pos1 consists of Position of period i.e "."
2. pos2 consists of Position of space i.e " "
3. NewString consisting of String without title
4. NewSubstring consisting of title if it is found in database
5. OldString consisting of original word.

Steps for titles are shown below

1. First input to the program will be caller name (stored in English).
2. Determine the position of "." And " " which are pos1 (position of ".") and pos2 (position of " ") .

3. If pos1>=0 and (pos1<pos2 OR pos2== -1)

3.1 Connected to database tts1 having table titles

3.2 Determine substring

3.3 If substring matches with English title present in the table titles

3.3.1 Determine the substring after position pos1+1 and store it in New_Word

3.3.2 New_Substring=substring

3.4 Else New_Word= Original_Word

3.5 End if

4. Else pos2>=0 and (pos2<pos1 OR pos2== -1)

4.1 Connected to database tts1 having table titles

4.2 Determine substring

4.3 If substring matches with English title present in the table titles

4.3.1 Determine the substring after position pos2+1 and store it in New_Word

4.3.2 New_Substring=substring

4.4 Else New_Word= Original_Word

4.5 End if

4. Else New_Word= Original_Word

5. End if

It will return a newstring without titles or if there are no titles then oldstring will be stored in newstring. If there are titles then newstring will from oldstring.

Function 3: Assigning consonant or vowel to English Characters

This function will determine whether a character is consonant or vowel and return a list of English characters.

Input: Newstring after pre-processing

Output: EnglishcharactersList

//Consonant_Vowel (newstring)

Steps

1. First it will take input as pre-processed string.

2. After reading string, the synthesiser will chop each character and determine whether it is a consonant, vowel or other character (that is "." Or "").

2.1 If it is a consonant, then it will assign "2" to it

2.2 If it is vowel, then it will assign "1" to it.

2.3 Otherwise "3" will be assigned

3.Then make combinations like “chh”, ”jh”, ”ch”, ”kh”, ”gh” ,”ai”, ”au” ,”ee”, ”oo”, ”aa” ,”bh”, ”dh”, ”sh”, ”th”, ”ou” *etc* and store in englishcharacters list.

Return list of englishcharacters and whether a character is consonant or vowel or other character.

Function 4: Determine initials in the string

This function will determine whether a character is initial or not. For Example, Mr. M.K Sharma then ‘M’ is the character and ‘K’ is the character, M.K, Y, Nitish, Naina, M Sharma.

Input: List of englishcharacters with initials

Output: List of englishcharacters without initials

VARIABLES

1. Englishcharacterlist contains list of characters where englishcharacterlist can have initials or cannot have initials.
2. Initiallist contains list of initials if it is found in a string
3. Newcharacterlist contains list of characters without initials
4. Character variable stores the character at the current location

Steps for initials are as follows

1. If character is either consonant or vowel
   1.1 If pos==0 AND character at i+1 position is “.” OR”\0” OR null
      1.1.1 Put that character in initial list
   1.2 If (character==”.” OR character==”\0”) AND character at i+1 position is “.” OR “\0” OR null
      1.2.1 Put that character in initial list
   1.3 Otherwise put that character in new_characterlist
   1.4 End if
2. Otherwise put that character in new_characterlist
3. character stores the character at current location
4. Repeat the steps 1 to 3 until list is empty.
5. End if

Return a character list without initials

3.7 GRAPHEME TO PHONEME MAPPING

This phase needs deep knowledge of Punjabi Phonology. This unit acts as a main unit since it provides phonetic transcription of input orthographic text.

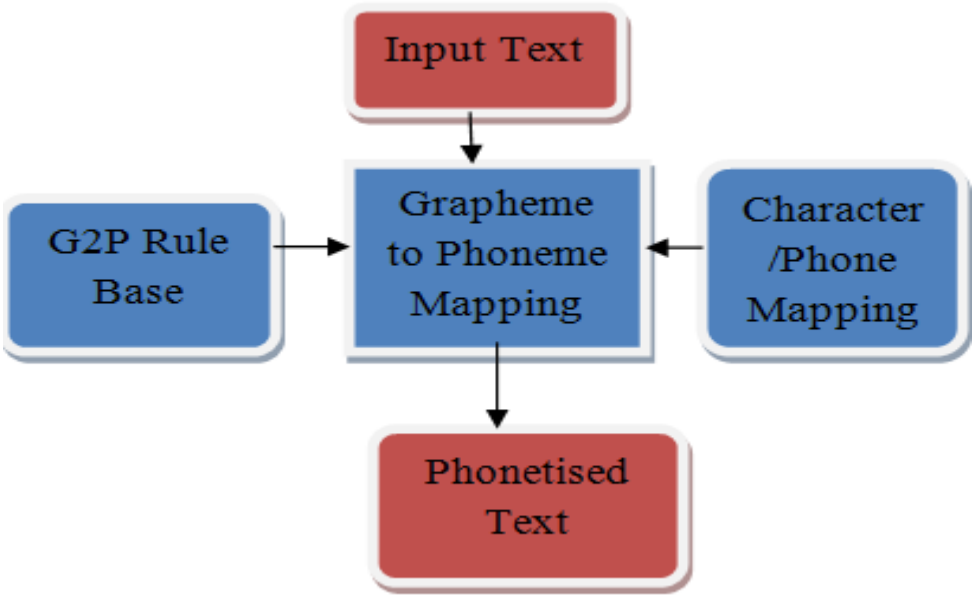


Figure 3.5Grapheme to Phoneme Mapping

Phase 2: Grapheme to Phoneme Conversion

1. Englishcharacterlist is retrieved from earlier stage.
2. Then one to one mapping is done between English tokens and Punjabi phonemes. Simultaneously, tokens are resolved through HTK and Rule Based Approach.
3. Finally, list of Punjabi phonemes are retrieved.
4. After that syllables are formed.

Below are given set of rules for mapping English character to Punjabi Phoneme.

INPUT English list of tokens

OUPUT Punjabi _Phonemes_list

VARIABLES

1. Newcharacterlist is a string type 3*3 list consisting of English phonemes, whether that phoneme is consonant or vowel and position of phoneme in a string.

2. PunjabiPhoneme is a string type 3 *3 list consisting of Punjabi phonemes, whether that phoneme is consonant or vowel and position of phoneme in a string.

Rules

Some customized rules are formulated for assigning Punjabi phonetic transcription to map English orthographic text. Initially list of 150 words was considered and some rules are formulated out of it. For Example: English letter ‘a’ is mapped with /ਅ/ in Punjabi.

English Letter	A	aa	I	Ee	U	oo	E	ai	Au	o	ou	An	aan
Punjabi Phoneme	ਅ	ਆ	ਇ	ਈ	ਉ	ਊ	ਏ	ਐ	ਔ	ਓ	ਔ	ਅੰ	ਆਂ

Table 3.12 List of Vowels with mapping

English Letter	**Punjabi Phoneme**
K	ਕ
Kh	ਖ
G	ਗ
Gh	ਘ
Ch	ਚ
Chh	ਛ
J	ਜ
Jh	ਝ
P	ਪ
F / ph	ਫ
B	ਬ
Bh	ਭ
M	ਮ
Y	ਯ, ਈ
R	ਰ,ੜ
L	ਲ
V,w	ਵ
S	ਸ
H	ਹ
Sh	ਸ਼
T	ਤ,ਟ
N	ਨ
Th	ਥ,ਠ
D	ਦ
Dh	ਧ, ਢ

Table 3.13 List of Consonants with mapping

Below are given some of the generic rules that work on every name.

Letter	Rule	Position of Letter	Sound	Examples
A	(*)a	Last Position	ਆ	Kamla
Y	(*)(a/e)y	End Position	ਏ	Nishay, parlay, Shreya
E	E(* - k)	First Position	ਇ	Esha , Eshita
R	(*)(C)R(*)	Any Position	ਰ੍	Prerna , Harpreet
H	(*)(C)H(*)	Any Position	ਹ੍	Siddhant , Ashis
	(*)(C)(C)(*)	Any Position	Adhak	Vallabh ,Siddhant
N	(*)An(*)	Any Position	Tippi	Ankita, Nishant

Table 3.14 Letter to Sound Rules where * denotes anything C denotes consonant

Different problems arised during this conversion these are:

- Problem of ਅ and ਆ -The problem is where to take phonetic transcription of letter as ਅ and ਆ in between word. For example, bharat and bharati.
- Problem of ਤ and ਟ-The problem is that whether to take phonetic transcription of letter t as ਤ or ਟ in word. For example , Tony and Tanvi.
- Problem of ਧ and ਢ-The problem is that whether to take phonetic transcription of letter dh as ਧ and ਢ . For example, dhanush and dholkiya.

- Problem of ਥ and ਠ-The problem is that whether to take phonetic transcription of letter th as ਥ and ਠ.For example , akshath and Thumri.

So, the problem of letter "a" and "t" is solved by using rule based approach and HTK. 200 names are examined and after that some rules are formulated. These names are taken from some Indian web sites which suggest baby names. Some common pattern needs to be determined. Some of the rules for determining such patterns are

Letter	**Rule**	**Sound**	**Examples**
a	[](s)a(dh)	ਅ	Sadhna, sadhvi, sadhika, Sadhri
A	(*-(k, dh))anya	ਅ	Kanya, dhanya, manya, Tanya
A	(*-(k, kr, sh, dh))ant	ਅ	Siddhant, vikrant, Nishant
A	(*-(m)) ani	ਅ	Mani, vani, pani, dhani
A	(*)lal	ਅ	Gulzarilal
A	(*)ali	ਅ	Kali,mali,deepali,bali
A	(*)am	ਅ	Kamraj , ram
A	(*)bha	ਅ	Shubhangi,nabhanyu, Vibhanshu
A	(*)amini	ਅ	Jamini,yamini

Table 3.15 Patterns for letter "a"

It will return a 3*3 matrix of characters consisting of Punjabi phonemes, whether that phoneme is consonant or vowel and position of phoneme.

Punjabi Phonemes are retrieved from earlier stage and then it is segmented into syllables and phonemes as per rules C, V and CV. This is done by function syllabification (Punjabi_phonemes_list).

//Call Syllabification (Punjabi_phonemes_list)

Return a list of syllables in Punjabi

Steps are as follows

//Syllabification (Punjabi_phonemes_list)

Steps

1. As we segmented the phonemes into consonant and vowel and assign 1 to vowel, assign 2 for consonant and 3 for other characters (“.” OR “\0”) and put in phoneme_list.
2. Repeat the steps until Punjabi_ phonemes_list is not empty
3. If first, second and third character of Punjabi_phonemes_list is not null
   3.1 If first character is consonant and second character is vowel
      3.1.1 add syllable_list=first character+ second character
      3.1.2 Value of i is incremented by 2
      3.1.3 end if
   3.2 If first character is consonant and (second character is consonant OR second character is other character)
      3.2.1 Add syllable_list=first character
      3.2.2 Value of i is incremented by 1
      3.2.3 End if
   3.3 if first character is vowel and (second character is consonant OR second character is other character OR second character is vowel)
      3.3.1 Add syllable_list=first character
      3.3.2 Value of i is incremented by 1
      3.3.3 End if
   3.4 if character is other character then
      3.3.1 add syllable_list=”silence”
      3.3.2 Value of i is incremented by 1
      3.3.3 End if
4. Else If character is not null and next character is not null and third character is null, repeat the steps 3
5. Else if character is not null
   5.1 if character is consonant and (second character is null or second character is “\0”)

5.1.1 add syllable_list=first character+ second character

5.1.2 Value of i is incremented by 1

5.1.3 end if

5.2 If character is vowel and (second character is null or second character is "\0")

5.2.1 add syllable_list=first character+ second character

5.2.2 Value of I is incremented by 1

5.2.3 End if

6. End if

7. End While

It will return a list of syllables that are CV, C and V of phonemes.

3.8 SPEECH SYNTHESIS

Speech synthesis is process of generation of output speech. Concatenative Synthesis along with rule based approach is used to synthesise the waveform and production of desired natural speech.

Phase 3: Speech Synthesis

Speech Synthesis is done by concatenative technique.

Finally, during synthesis, if substring (it consist of English title) matches with the title present in the database and substring is not empty then pathof_titles are stored in sound_list. If initial (it consist of initials) matches with the initial present in the database and pathof_initials is not empty then pathof_initials are stored in sound_list and it numerals are not empty then pathof_numerals are added. If syllables are present in the database and syllables are not empty then pathof_syllables are stored in sound_list. This is done by function SpeechSynthsis (pathof_titles, pathof_initials , ,pathof_numerals,pathof_syllables).

//Call SpeechSynthsis(pathof_titles,pathof_initials, pathof_syllables,pathof_numerals)

Return a Synthesised Speech

This algorithm will segment the sequences of Punjabi phonemes into syllables.

INPUT Punjabi Phonemes

OUTPUT Punjabi Syllables in syllable_list

Function 1

Sound Player Class is used to play wave files and PlaySync method is used to play wave files one by one.

//SpeechSynthesis (pathof_titles, pathof_initials, pathof_syllables, pathof_numerals)

INPUT Syllables, Initials, Titles and numerals

OUTPUT Path of wave files

Variables

Sound_list consist of path

M consist of position where wave files are stored.

STEPS

1. If titles is not equal to null
   1.1Add the path to sound_list at m position
2. Else if initials are not null
   2.1 Add the path to sound_list at m position
3. Else if Numerals are not null
   3.1 Add the path to sound_list at m position
4. Else If syllables are not null
   4.1 Add the path to sound_list at m position
5. End if

And finally synthesised speech is produced.

CHAPTER 4

TESTING AND RESULTS

In this chapter there are different test cases presented and finally result and analysis is provided for grapheme to phoneme phase.

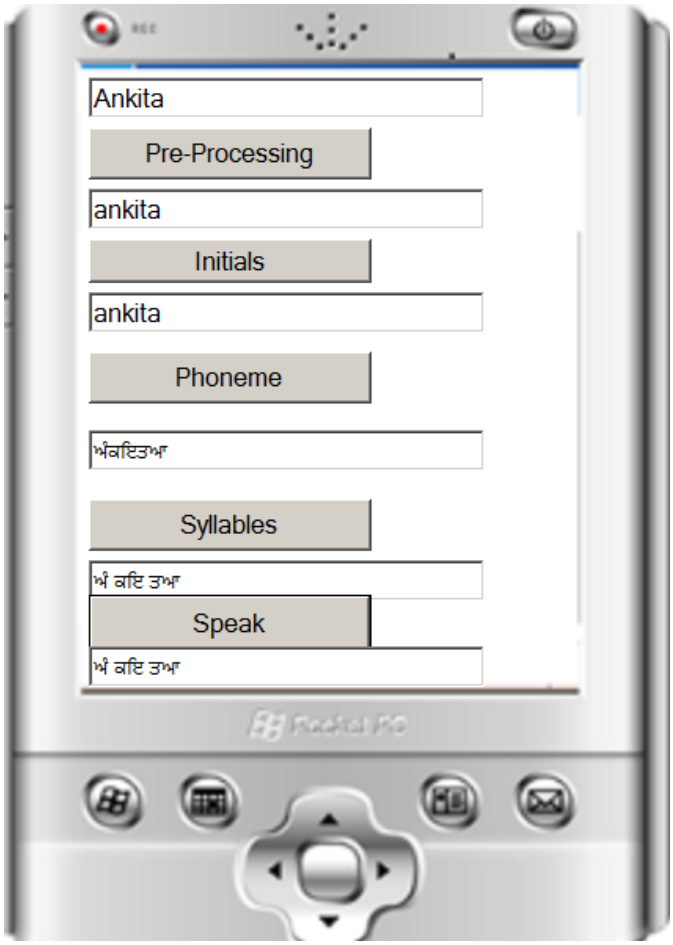
4.1 TEST CASES

The simulation takes input as caller ID (i.e Name stored in phone book which is in English) and map that into desired Punjabi phones after processing.

Case 1

Input String- Ankita

Output Syllables-ਅੰ ਕਇ ਤਆ



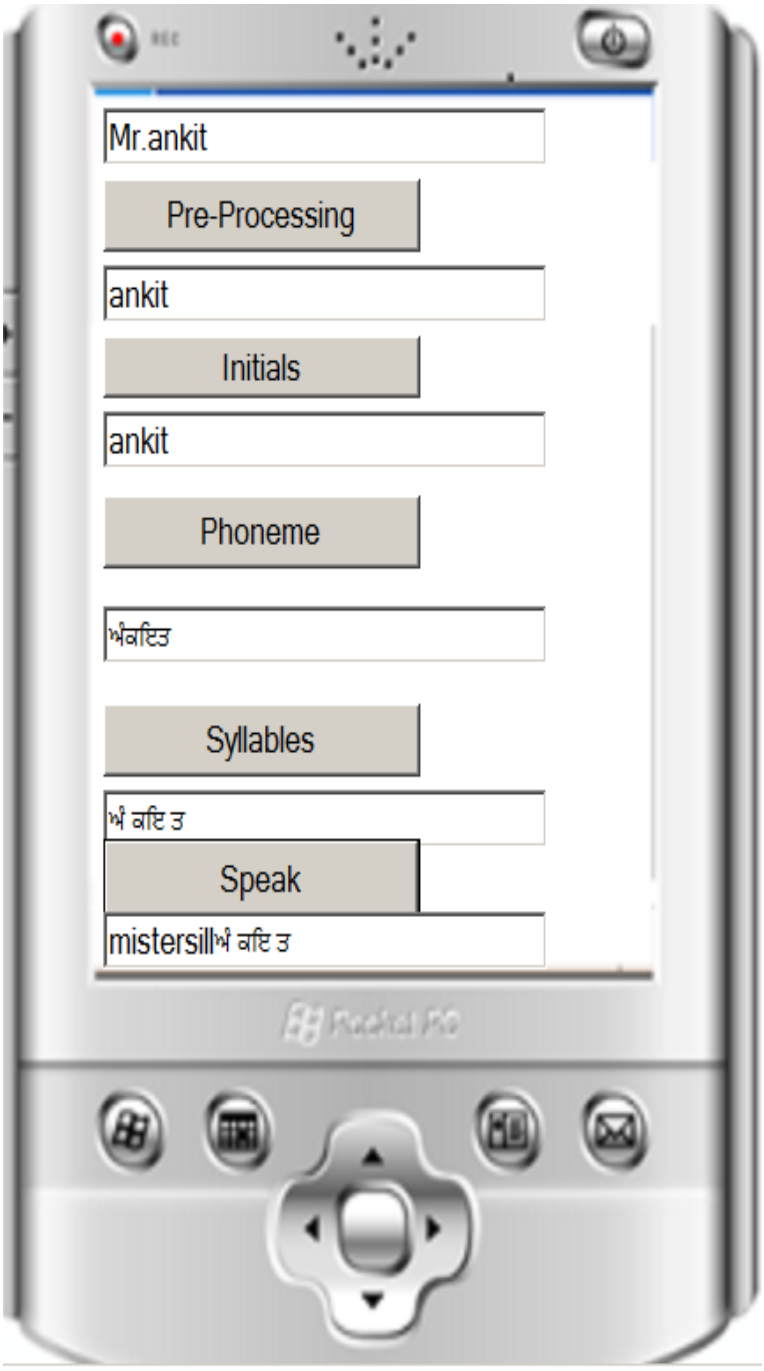
Case 2

Input String: Mr. Ankit

Output String:

Pre-Processing : Mr.

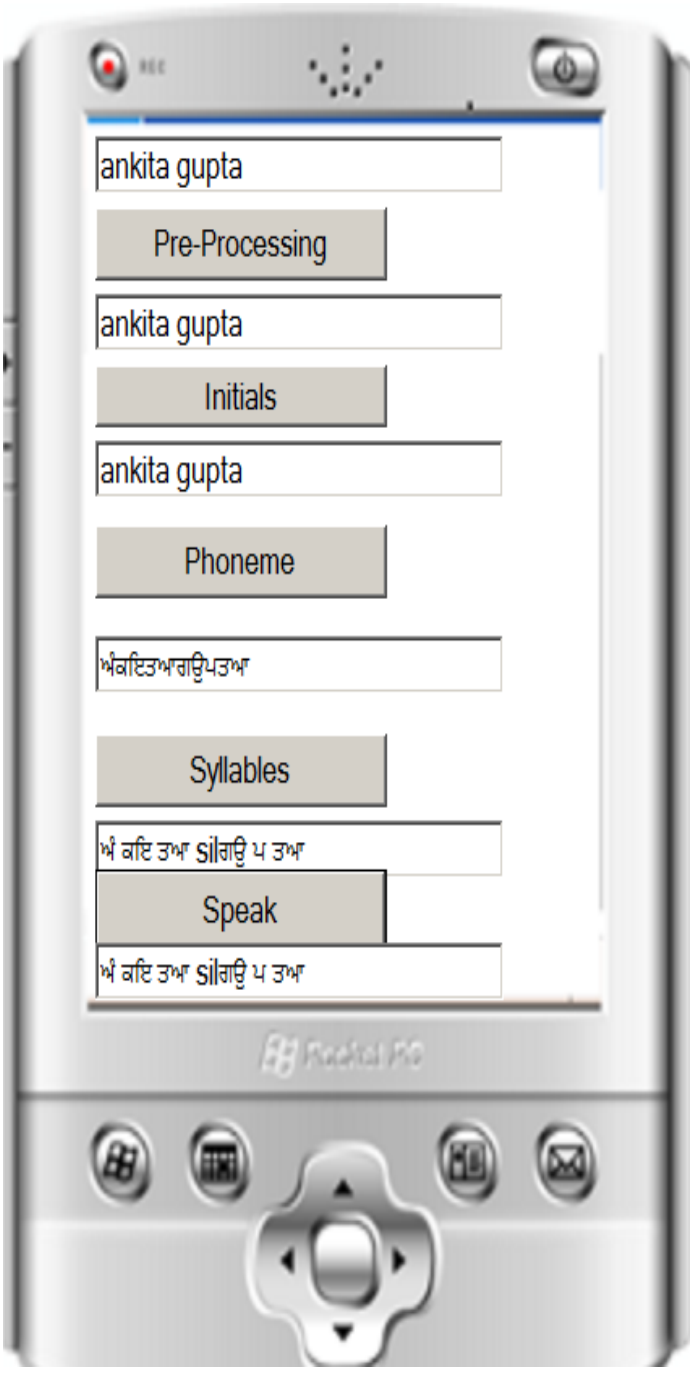
Syllables /ਅੰ/ /ਕਇ/ /ਤ/



Case 3

Input String: Ankita Gupta

Output String: /ਅੰ/ /ਕਇ/ /ਤਆ/ sil/ਗਉ/ /ਪ/ /ਤਆ/



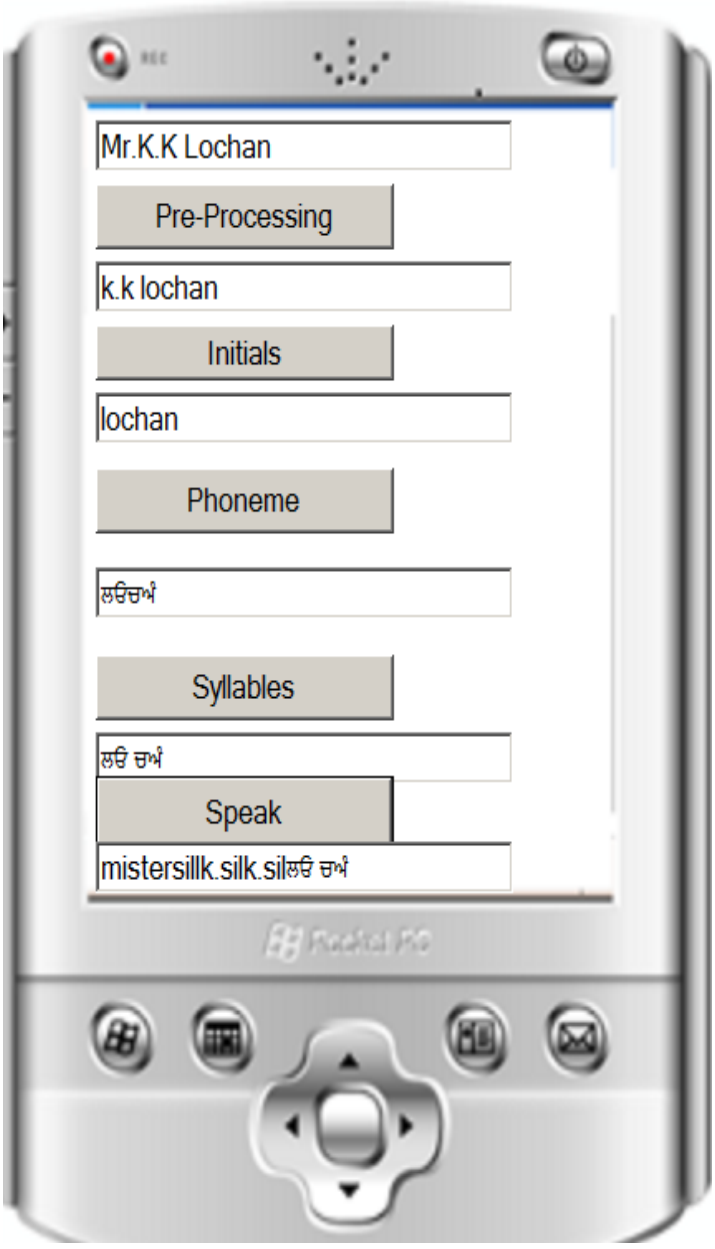
Case 4

Input String: Mr. K.K Lochan

Output String: mistersil k.silk.sil ਲਓ ਚਅੰ

Pre-Processing: Mr., K.K

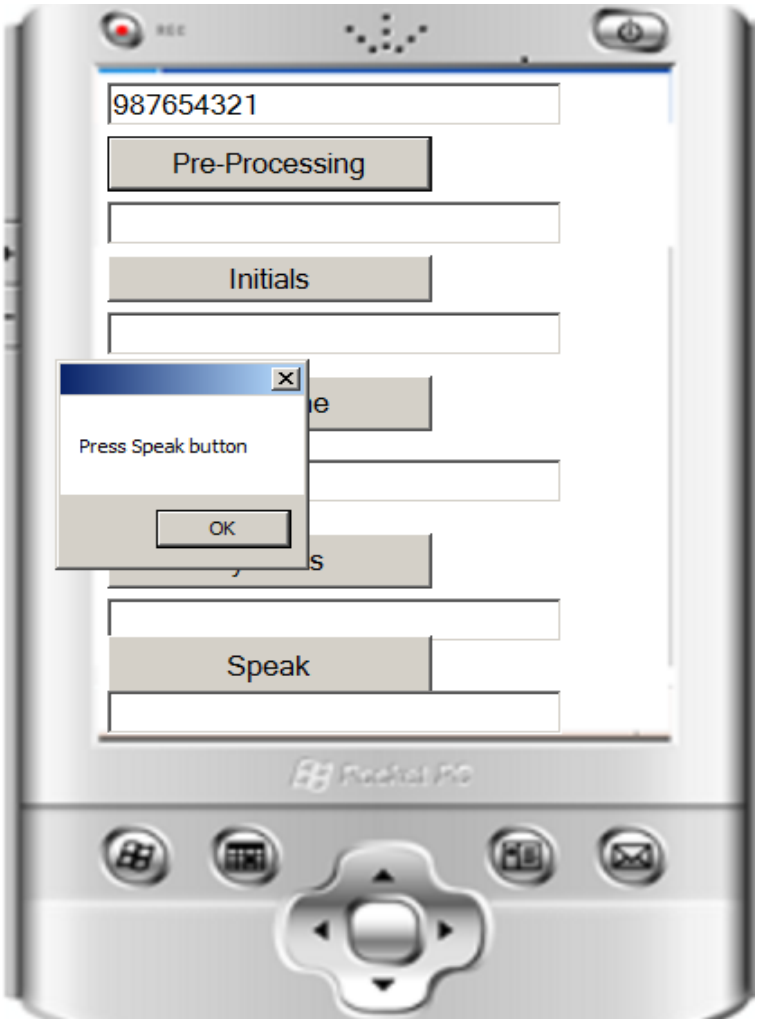
Syllables : ਲਓ ਚਅੰ

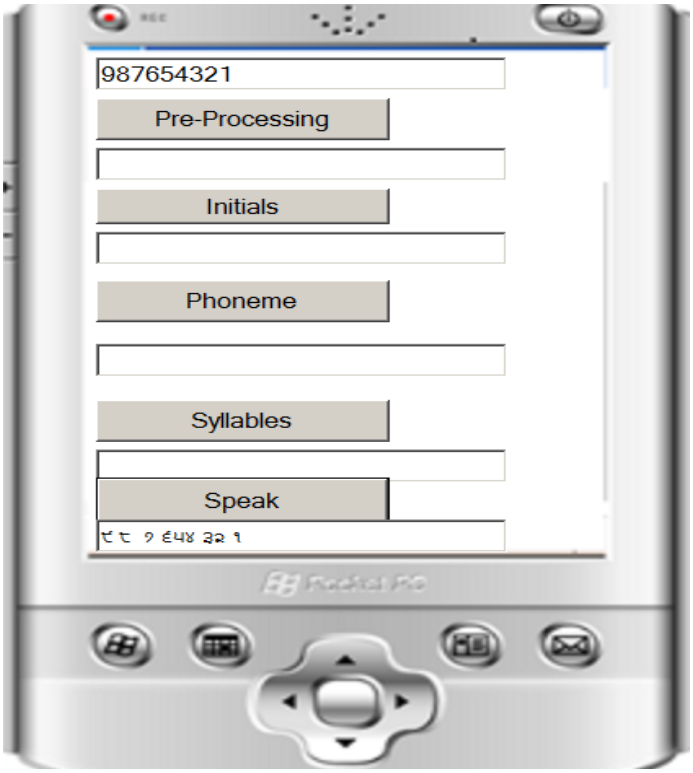


Case 5

Input String: 987654321

Output String: ৫ ৳ ৭ ৫५४ ३२ ৭





4.2 RESULT

GRAPHEME TO PHONEME MAPPING

The main aim is to map English tokens to Punjabi phonemes. To measure the accuracy of mapping, the following is the equation

$$A = (N / T) * 100$$

Where

A = Accuracy

N = Number of correct phonemes

T = Total number of words

For transliteration of English Text to Punjabi phonemes some rules were formulated as discussed in previous section. This system works accurately for most of the Indian names which are taken from common Indian web sites. The system has been tested on approximately 1400 words.

Total No. of Names	Correct Names	Incorrect Names	Accuracy
1400	1150	250	82%

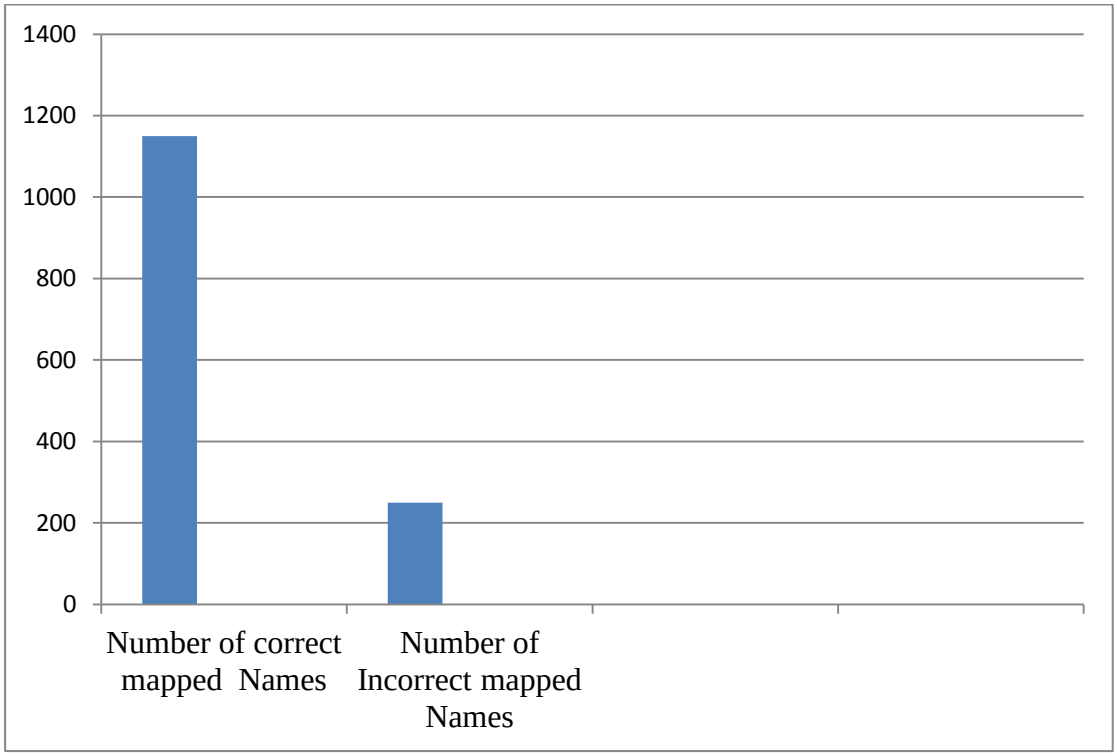


Figure 4.1 Correct and Incorrect Mapped Names

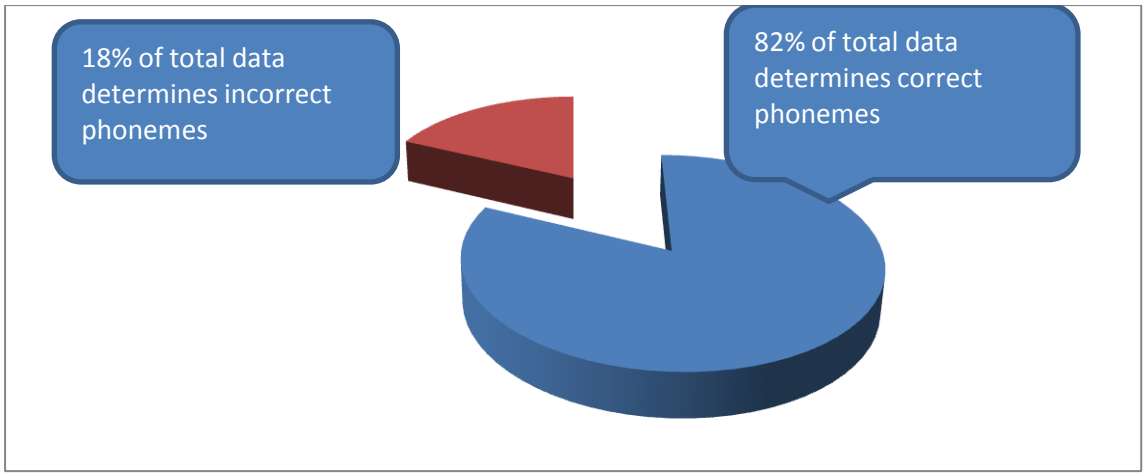


Figure 4.2 Accuracy of Grapheme to Phoneme Mapping

4.3 Error Analysis

- Multi-Mapping Problem

  One letter is mapped to multiple phonemes. For example, letter “a” is mapped to multiple phonemes.

A	**ਅ, ਆ**
Th	**ਥ,ਠ**
T	**ਤ,ਟ**
Dh	**ਧ, ਢ**

Table 4.1 Multi Mapping Problems

- Similar Spelling words

Words having same spelling but different pronunciation.

BalGopal (**ਬਾਲਗੋਪਾਲ**)	BalKumar (**ਬਲਕੁਮਾਰ**)
Kali (**ਕਲਿ)**	Kali **(ਕਾਲੀ)**
Bala (**ਬਾਲਾ)**	Bala **(ਬਲਾ)**

Table 4.2 Similar Spelling Words but different Pronunciation

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

This chapter gives a brief conclusion of thesis. After that future directions and unresolved issues are given.

5.1 CONCLUSION

During the development of mobile TTS application for Punjabi Language where input is in English Language and the output speech is uttered in Punjabi Language. Text to speech synthesiser is judged on the basis of naturalness and intelligibility. A system is highly intelligible if it determines various abbreviations, acronyms and numerals in the input string *i.e* Caller ID. Naturalness can be judged on the basis of output speech. In the pre-processing phase, a survey was conducted and it was observed 70% of names were in category of generic (that are Ms.Kanika, Mr.Ankit, Amit Lochan) it is taken for further processing while rest of it *i.e* in non-generic or non-standard were ignored. Secondly, mapping of English tokens to Punjabi phonemes is one of the major tasks. During that phase some problems occurred and for that certain rules and procedures were deviced. Issues like mapping of letter "a" with ਅ and ਆ. For this phase probability of mapping of letter "a" and "t" are resolved with accuracy rate of of approximately 82%.

On the basis of work done so far it is concluded that with the help of Rule based and Concatenative approach it is possible to achieve the maximum intelligibility and naturalness. The problem of degraded voice quality produced by the simulator could be overcome with the proper-infrastructure and environment. Moreover, if phoneme /syllable is labelled and built from a sound wave file then more efficiency can be achieved.

Thus, the application when actually embedded with some mobile will work well for illiterate masses that understand their native language (*i.e* Punjabi) and for those who are not comfortable with mobile technology *i.e* old age people with vision problems.

5.2 FUTURE SCOPE

This synthesiser works well for most of the Indian names taking into consideration common titles and surnames.

- Prosody and other factors (like pitch, duration and intensity) can be added to the system.
- Accent Variation and multiple voices (male and female voices) can be added in the system.
- Better results can be derived with proper infrastructure and environment.
- This application can be integrated to read short text messages.
- Listen reminders.
- It can be further enhanced for alarm tones.

5.3 LIMITATIONS OF PUNJABI TTS SYNTHESISER FOR MOBILE DEVICE

Punjabi TTS Synthesiser works well for majority of Indian names but still some of the names limitations persist in the system.

- It simply concatenates the speech units at pitch period and plays out but does not make any attempt to do prosody effect on output.
- Pitch modification are not incorporated due to lack of infrastructure.
- Certain problems occur while mapping tokens from English to Punjabi phonemes.

PUBLICATIONS

Accepted

Goel, A., Bansal, D. and Jindal, K. 2012. Grapheme to Phoneme Conversion for Punjabi Language. *International Journal of Science, Technology & Management* .3(1)

Communicated

- Bansal, D., Goel, A., Jindal, K., 2012. Punjabi Speech Synthesis System Using HTK. communicated in *The International Journal of Information Science & techniques*.
- Bansal, D., Goel, A., Jindal, K., 2012. Punjabi Speech Synthesis System Using HTK. communicated in *Speech Communication*.

.

REFERENCES

[1] Balyan, A., Agrawal, S.S. and Dev, A. Development of Hindi Speech synthesiser for Metro Rail Information System. In *International Conference of Electrical and Control Engineering,* (2011).*2166-2170*

[2] Black, A. W, Hsiao, R., Venugopal, A., Kohler, T., Zhang, Y., Charoenpornsawat, P., Zolmann, A., Vogel, S., Schultz, T. and Waibel, A. ,Optimizing Components for Handheld Two-way Speech Translation for an English-Iraqi Arabic System. In *proceedings of INTERSPEECH–ICSLP* (2006)

[3] Breiman, L., Freidman, J.H, Olshen R.A., and Stone, C.J. *Classification and Regressions Trees*. CRC Press .Wadsworth, Belmont CA (1984).

[4] Chalamandaris, A., Raptis, S., Tsiakoulis, P. Rule Based Grapheme to Phoneme method or Greek. In *proceedings of INTERSPEECH* (2005), 2937-2940

[5] Deep, K. and Goyal, V., (2011). Development of English to Punjabi Transliteration System. *International Journal of Computer Science and Communication***,** *2(2),* 521-526

[6] Duan, Q., Kang, S., Wu, Z., Cai, L. Comparison of Syllable/Phone HMM based Mandarin TTS. In *Proceedings of International Conference of Pattern Recognition*(2010),4496-4499

[7] Dutoit, T. *An Introduction to text to Speech Synthesis*. Kluwer Academic Publishers, NetherLands, 1996.

[8] Gera , P. and Sharma , R.K.. Text to Speech Synthesis for Punjabi language. (2007), M.Tech Thesis, Thapar University, Patiala

[9] Klatt, D. (1987). Review of text-to-speech conversion for English. *Journal of the Acoustical Society of America*, 82(3), 737-93

[10] Lehal, G. and Singh, P. Text to Speech Synthesis for Punjabi Language. (2006) , M.Tech Thesis, Punjabi Language, Patiala

[11] Lemmety,S. Review of Speech Synthesis Technology. (1999), M.Sc Thesis, Helinki University of Technology, Finland.

[12] Maia, R. da S , Zen ,H., Tokuda, H., Kitamura, T., Resende Jr, *F.G.V*,(Towards The development of Brazilian Porutguese Text to Speech System Based on HMM. In *Proceeding Of Eurospeec* , (2003),2465-2468.

[13] Manning, C.D. Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge**,** 1999

[14] Mukhopadhyay, A., Soumen, C., Choudhury, M., Lahiri, A., Dey, S., Basu A. (2006) Shrutian embedded text-to-speech system for Indian languages. In *IEEE proceeding software,* 153(2), 75-79

[15] Ogbureke, K.U., Cahill, P., Berndsen, J.C. Hidden Markov Models with Context-Sensitive Observation for Grapheme to Phoneme Conversion. In *INTERSPEECH* ISCA, (2010), 1105-1108

[16] Palo, P. A review of Articulatory Speech Synthesis, (2006), M.Tech thesis, Helsinki University, Helsinki

[17] Prudon, R. and Alessandro, C. A selection/concatenation text to speech synthesis Database development, system design, comparative evaluation. In *4th ISCA/IEEE International Workshop on Speech Synthesis, (*2001)

[18] Rabiner, L.R and Juang , B.H.(1985), An Introduction to Hidden Markov Models, *IEEE ASSP Magazine,3(1), 4-16*

[19] Raman,R K V S, Sarma ,S., Sridevi S, Thomas R.,(2004) Matrubhasha - An integrated speech framework for Indian language

[20] Singh, P.P . Siddhanttik Bhasha Vigyan. Madan Publications, Patiala, 2002, 275-411.

[21] Shribahadurkar, S.D. and Bormane, D.S.(2009).Marathi Language Speech synthesiser Using Concatenative Strategy. *International Journal of Recent Trends in Engineering*, 2(4),2590-2593

[22] Styger,T. and Keller,E., (1994). Formant synthesis. In *E. Keller (ed.), Fundamentals of Speech Synthesis and Speech Recognition: Basic Concepts, State of the Art, and Future Challenges*, 109-128

[23] Tokuda, K., Zen, H and Black, A.W., An HMM-based speech synthesis system applied to English. In *Proceedings of IEEE SSW* (2006), 227-230

[24] Taylor, P. Hidden Markov Models for Grapheme to Phoneme Conversion. In *proceedings of INTER-SPEECH*, (2005), 1973-1976

[25] Udhyakumar, N., Kumar, C.S., Srinivasan, R. and Swaminathan, R. Decision Tree Learning for Automatic Grapheme to Phoneme Conversion for Tamil. In 9^{th} *Conference of Speech and Computer*, (2004).

[26] Wolter, M. A Diphone based Text to Speech Synthesis for Scottish Gaelic, (1997), Thesis for Degree of Diploma in Infomatik, University of Bonn.