কথা

THE FIRST
TEXT TO SPEECH SYNTHESIS
FOR BANGLA LANGUAGE

A Thesis

Submitted to the Department of Computer Science of

BRAC University

by

PROMILA KANTI NATH

Student ID: 01201026

Requirements for the Degree of

Bachelor of Science in Computer Science

Spring 2006

BRAC University, Dhaka, Bangladesh.

DECLARATION

I hereby declare that this thesis is based on the building of a demo software named কথা as a **TEXT TO SPEECH SYNTHESIS FOR BANGLA LANGUAGE**. It's a preliminary step to convert the Bangla text to its Bangla voice. As it is a demo, it may not fulfill all the requirements of the user and need more improvement of it's word-diphone bank to recognized all diphones for corresponding words given by any user.

Materials and third party utility that I used mention in reference. This Thesis, neither in whole nor in part, has been previously submitted for any Degree of BRAC University.

Signature of
Supervisor

Dr. Mumit Khan

Signature of
Author

Promila Kanti Nath

ACKNOWLEDGMENTS

My first thank is to my supervisor who gave me the brave of completing of such a big thesis. When I went to him with this thesis proposal, he just surprised and told me that it may not complete in one semester but the way you want to move would be a milestone for the researcher of Bangla Language. That was the real spirit, which help me to reached on this satisfactory stage. At the middle of it, I became hopeless to face some unsolvable errors. But soon we contact with the developer team Ksenia, Rob Clark, Roger Tuker of the third party utility software, Festival to produce sound.

This is not only a thesis; it is obviously a part of research on Bangla Language and Bangla voice processing.

At first I moved with sound syllables in .NET platform in C#. But soon I found that, It was not so perfect and difficulty occurs when stretch for expression involve. And the another problem was, some times the syllable break through from each other and not so continuous. So it loose the natural voice that we talk as usual and goes almost mechanic voice.

I want to give my heartiest gratitude to Matin Saad Abdullah to discover some new sound related method in C# which was helped to sync my syllable to each other in C#.

The Center for Research on Bangla Language processing, BRAC University was a great resource for many Bangla document on Bangla language and Bangla language processing book. Basa Bigganer Kotha by Daniul Haque and Dhani Biggan O Dhanitotto by Abdul Hai is remarkable for their rules of diphone design and syllabification.

Mrs. Naira Khan, Lecturer of English and Linguistic was always helpful to recognize the typical term of phonetics.

I am also grateful to my friend Naushad UzZaman, Research Programmer, Center for Research on Bangla Language processing, BRAC University who provided me full support all the time and by given his translation script through Bangla letter to it's corresponding English letters.

Another senior brother of that LAB, Mr. Zohurul Islam helped us to include this as a feature with there Bangla Pad. So it became very easy to the user to use.

Last but not at least, thanks to the Almighty for helping me in every steps of this thesis work.

ABSTRACT

In this Thesis paper, we represent Text to Speech (কথা) synthesis system for Bangla language and how can we develop it using phonology, prosodic rules, G2P conversion including prosodic information in the Festival [1] framework. Since Festival does not provide complete language processing support for all languages, it needs to be augmented to facilitate the development of TTS Systems (কথা) in certain new language Bangla. A new language needs inputs for resolving language specific issues requiring close collaboration between linguists and technologists. Large amount of annotated data is required for developing language-processing modules like text normalization, grapheme-to-phoneme (G2P), intonation, and time duration models. We propose how these modules can be develop and integrate it with Festival.

TABLE OF CONTENT

DECLARATION..2
ACKNOWLEDGEMENT..3
ABSTRACT 5
TABLE OF CONTENT 6
CHAPTER I: BACKGROUND 8
CHAPTER II: INTRODUCTION 10
CHAPTER |||: TEXT ANALYSIS 12
3.1 DEFINITIONS 12
3.2 TEXT NORMALIZATION 12
3.3 TRANSLITERATION 12
3.4 ALGORITHM FOR TRANSLITERATION 17
CHAPTER |V: PHONETIC ANALYSIS...19
4.1 DEFINITIONS 19
4.2 STEPS OF PHONETIC ANALYSIS 19
4.2.1 BUILDING LARGE AMOUNT OF LEXICON BY HAND........................19
4.2.2 BUILDING LETTER-TO-SOUND (LTS) RULES BY HAND 21
4.3 OUT OF VOCABULARY WORDS 21
CHAPTER V: FESTIVAL..22
5.1 FUNDAMENTAL COMPONENTS OF FESTIVAL......................................22
5.2 TEXT PRE-PROCESSING...22
5.3 PROSODY 24
5.4 CONCATENATION 27
CHAPTER VI: WORD-DIPHONE BANK GENERATION 28
6.1TRANSLITERATION..30
6.2 UNIQUE WORD EXTRACTION...30
6.3 ALGORITHM FOR UNIQUE WORD EXTRACTION 30
6.4 POST LEXICON RULES..33
6.5 RECORDING DIPHONES 34
CHAPTER VII: FUTURE IMPLEMENTATION 36
REFERENCES..37

LIST OF TABLES

1. Consonants 13
2. Vowels 14
3. Bangla Consonant and Vowel Phone sets 14
4. Vowels and Consonants Transliteration 15
5. Digits 16
6. Modifiers 16

LIST OF FIGURES

1. Bangla Text To Speech Block Diagram 11
2. Festival Fundamentals 22
3. Text Pre-Processing (Block Diagram) 23
4. Inside Prosody 24
5. Property Of Speech Signal 25
6. Property Of Speech Signal 26
7. Psola – Pitch Synchronous Overlap And Add 26
8. Pitch Marks 27
9. Word-Diphone Bank Generation 29
10. Praat Visual Output 34

CHAPTER I: BACKGROUND

Speech Synthesis System converts input text into speech waveforms. The input text is essentially a string of characters, might be data from a word processor, which is text as standard **Unicode** format. The first task is to analyze the raw text. The text analyzer is the conversion of input raw text into phonetic representation i.e. **grapheme-to-phoneme (G2P)** conversion. A grapheme is an actual text whereas the phoneme is a token directly maps to a signal unit in the voice database. Voice database will create based on the 48 phones [2] in Bangla. The one to one mapping is like:

বাংলাদেশ → baŋlad̪eʃ

This conversion is a required phonological, prosodic intonation rules. Text containing digits and numbers are converted into full words based on number system rules. After conversion of G2P the synthesizer will produce speech by taking phoneme string and information for intonation and prosody as input. Intonation, stress, duration etc. are called Prosodic or Suprasegmental Elements required introducing naturalness into the synthesized speech. They are also related with fundamental frequency, segmental duration and stress in first syllable.

Festival [1] framework was chosen for implementation our research synthesis system because of its flexibility and modularity architecture, eases of configuration and ability to ass new modules.

Steps

1. **G2P CONVERTION**
    a. Developing G2P rules
    b. Developing character to phone mapping

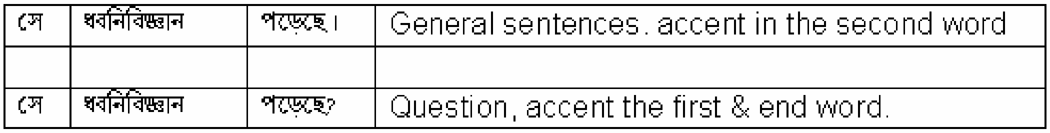
2. ADDING STRESS IN FIRST SYLLABLE

The first syllable in every Bangla word has a stress that’s why we need to include the stress of the first syllable of the word.

Stress in first syllable → **আছে**

1. ADDING INTONATION

Through the study of Bangla Language I observe that Bangla Language has different intonation pattern, like pitches in the word, syllables as well as sentences level. Syllabic pitches movements are represented by three types of syllabic intonations namely rise, fall and flat. Considering both word level as well as the sentence level intonation can be generating intonation pattern of a sentence. Different type of intonation like:



সে	ধ্বনিবিজ্ঞান	পড়েছে।	General sentences. accent in the second word
সে	ধ্বনিবিজ্ঞান	পড়েছে?	Question, accent the first & end word.

CHAPTER II: INTRODUCTION

Speech is one of the most vital forms of communication in our everyday life. Since speech is a primary medium for communication among human beings, it is natural for the people to expect to be able to carry out spoken dialogue with computers. This involves the integration of speech technologies and language technologies. Speech synthesis is an automatic generation of artificial speech signal by the computer. In the last few years, this technology has been widely available for several languages for different platform from personal computer to stand alone system. But for Bangla language, there is no such system. The necessary of human computer interactions, a Text To Speech (কথা) system for Bangla language can overcome the human computer interaction, can help to overcome the illiteracy barrier of common mass, can also empower the visually impaired population and increase the possibilities of improved man-machine interaction through reading the online newspaper from Internet and enhancing other information system.

Bangla, also known as Bengali, is the native language of approximately 210 million people, the majority of whom live in Bangladesh and in the Indian state of West Bengal, making it the 4th most widely spoken language in the world. Since Bangla is the 4th most widely spoken which makes greater importance to develop a Text To Speech (কথা) system for Bangla language.

The function of Text To Speech (কথা) system is to convert the given text in to its sound waveform. The Text To Speech (কথা) system is based on the concatenation of basic speech units, diphones using Festival. The input text is transform to its corresponding voice form by a series of modules. These modules, constituting the TTS (কথা) system describe in detail. The system block diagram given in figure:1. The input text is essentially a string of words, might be data from a word processor that is text as standard **Unicode** format. The first task is to analyze the raw text. The text may contain numbers, dates, abbreviations, which need to convert as normal text is called text normalization. In **chapter III**, we describe in details the **Text Analysis**. The normalized text then convert into phonetic representation which is called phonetic analysis, this can be done by Grapheme to Phoneme (G2P) conversion or dictionary lookup or combination of both. In **chapter IV**, we present the **Phonetic Analysis**. After phonetic analysis we add stress and intonation in sentences, words and syllables, which is called **Prosodic Conversion** and the rest portion **Waveform Synthesis** is described in the report of my thesis partner Firoz Alam [ID: 01201056].

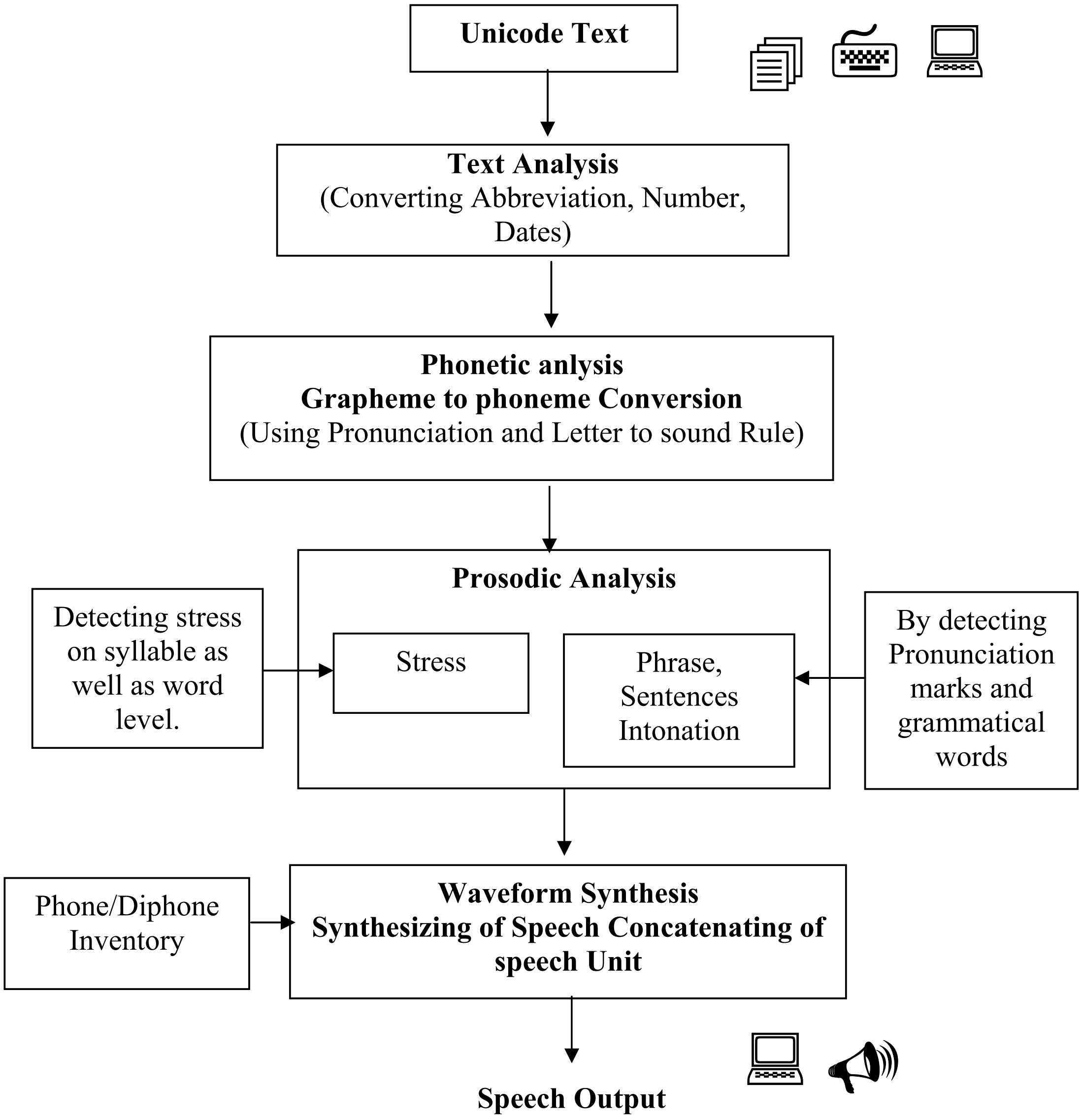

Unicode Text
Text Analysis
(Converting Abbreviation, Number, Dates)
Phonetic anlysis
Grapheme to phoneme Conversion
(Using Pronunciation and Letter to sound Rule)
Prosodic Analysis
Detecting stress on syllable as well as word level.
Stress
Phrase, Sentences Intonation
By detecting Pronunciation marks and grammatical words
Phone/Diphone Inventory
Waveform Synthesis
Synthesizing of Speech Concatenating of speech Unit
Speech Output

FIGURE 1 : BANGLA TEXT TO SPEECH BLOCK DIAGRAM

CHAPTER III: TEXT ANALYSIS

3.1 DEFINATIONS

The first step of Text To Speech (কথা) system is to analyze the text for containing raw text. Many Non-Standard Word (NSW) [2] may appears, like numbers (year, time, ordinal, cardinal, floating point), abbreviations, acronyms, currency, dates, URLs. All these non-standard texts must be normalize, or in other words have to convert to its text standard words. These NSW normalize by using text normalization and ambiguous token can be disambiguating using homograph disambiguation.

3.2 TEXT NORMALIZATION

Text normalization resolves tokenization by tokenize the input text into tokens; identify the Non Standard Words (NSWs) and their categories, and expansion of the NSWs into standard word representation. Tokenization is basically done based on the white spaces [Sproat et al., 2001]. Once tokenization is completed, each token has to be identified for its corresponding category. Then expansion of NSWs is accomplish by the combination of some steps (e.g., expanding numbers, currency, dates) and look-up tables (e.g., for abbreviations, acronyms). Work on text normalization in TTS (কথা) systems involves a set of rules. As our Bangla text is containing some of the issues such as: Text might contain interspersed English words using Roman script, making it a bilingual text. Numbers could be represented by English numerals at one place, and the native Bangla numerals at another place, in the same text. Special attention to foreign language tokens, when the tokens are written partly or fully in Bangla.

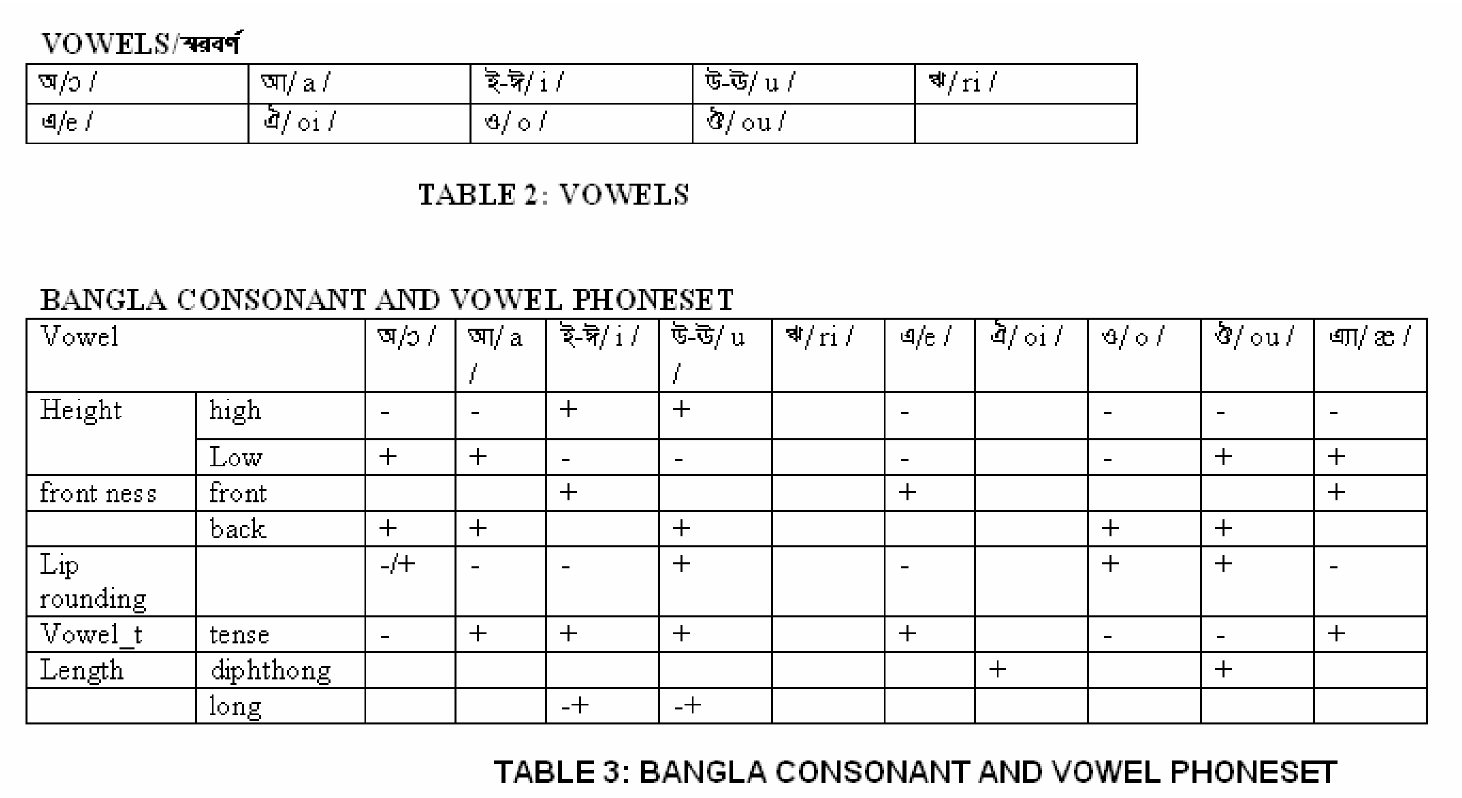
3.2 TRANSLITERATION

To translate Unicode Bangla letter to its corresponding English letters, I used standard rules given by Center for Research on Bangla Language processing, BRAC University. This is one kind of one to one translation. The rules are given bellow.

CONSONANTS/ব্যাঞ্জনবর্ণ

Manner of articulation		Place of articulation										
		Velar		Palato-alveolar		Alveolar		Dental		Bilabial		Glottal
		In- aspirate - অল্পপ্রাণ	Aspirate - মহাপ্রাণ	In- aspirate – অল্পপ্রাণ	Aspirate - মহাপ্রাণ	In- aspirate – অল্পপ্রাণ	Aspirate - মহাপ্রাণ	In- aspirate - অল্পপ্রাণ	Aspirate - মহাপ্রাণ	In- aspirate – অল্পপ্রাণ	Aspirate - মহাপ্রাণ	
Plossive/ Stop	Voiceless	ক/ k	খ/ k^h	চ/ c	ছ/ c^h	ট/ t	ঠ/ t^h	ত/ t̪	থ/ $t̪^h$	প/ p	ফ/ p^h	
	Voiced	গ/ g	ঘ/ g^h	জ/ ɟ	ঝ/ ɟ	ড/ d	ঢ/ d^h	দ/ d̪	ধ/ $d̪^h$	ব/ b	ভ/ b^h	
Fricative	Voiceless			শ,ষ /ʃ		স/ s						ঃ
	Voiced											হ /h
Nasal	Voiced	ঙ ং/ ŋ		ঞ/ ɲ		ন/ n, ণ/ n				ম/m		
Liquid/ Lateral	Voiced			য/ ɟ		ল/ l						
Trill	Voiced					র/ r						
Flapped	Voiced					ড়/ ɾ	ঢ়/ ɾ					
Glide	Voiceless			য়/ j								
	Voiced									ব/ w		

TABLE 1: CONSONENTS



VOWELS/স্বরবর্ণ

অ/ɔ /	আ/ a /	ই-ঈ/ i /	উ-ঊ/ u /	ঋ/ ri /
এ/e /	ঐ/ oi /	ও/ o /	ঔ/ ou /	

TABLE 2: VOWELS

BANGLA CONSONANT AND VOWEL PHONESET

Vowel		অ/ɔ /	আ/ a /	ই-ঈ/ i /	উ-ঊ/ u /	ঋ/ ri /	এ/e /	ঐ/ oi /	ও/ o /	ঔ/ ou /	এ্যা/ æ /
Height	high	-	-	+	+		-		-	-	-
	Low	+	+	-	-		-		-	+	+
front ness	front			+			+				+
	back	+	+		+				+	+	
Lip rounding		-/+	-	-	+		-		+	+	-
Vowel_t	tense	-	+	+	+		+		-	-	+
Length	diphthong							+		+	
	long			-+	-+						

TABLE 3: BANGLA CONSONANT AND VOWEL PHONESET

VOWELS AND CONSONANTS TRANSLITERATION

অ	A
আ	aa
ই	I
ঈ	Ii
উ	U
ঊ	uu
ঋ	ri
এ	e
ঐ	oi
ও	o
ঔ	ou
◌া	aa
◌ি	i
◌ী	ii
◌ু	u
◌ূ	uu
◌ৃ	ri
◌ে	e
◌ৈ	oi
◌ো	o
◌ৌ	ou
ক	k
খ	kh
গ	g
ঘ	gh
ঙ	Ng
চ	C
ছ	Ch
জ	J
ঝ	Jh
ঞ	Nio

ট	T
ঠ	Th
ড	D
ঢ	Dh
ণ	N
ত	Ta
থ	To
দ	Da
ধ	Do
ন	N
প	P
ফ	Ph
ব	B
ভ	Bh
ম	M
য	Z
র	R
ল	L
শ	Sh
ষ	sh
স	s
হ	h
ড়	ra
ঢ়	rh
য়	y
ৎ	
◌ং	Ng
◌ঃ	:
◌ঁ	
◌্	

TABLE 4: VOWELS AND CONSONANTS TRANSLITERATION

DIGITS

Arabic numerals	0	1	2	3	4	5	6	7	8	9
Bangla numerals	০	১	২	৩	৪	৫	৬	৭	৮	৯
Bangla names	shunno	ek	dui	tin	char	paac	chy	shaat	aat	Ny
	শুন্য	এক	দুই	তিন	চার	পাঁচ	ছয়	সাত	আট	নয়

TABLE 5: DIGITS

MODIFIERS **

Symbol with [kɔ] (ক(	Name	Function	Transliteration
ক্	hôshonto	Suppresses the inherent vowel	-
কৎ	khônđo tô	Final unaspirated dental [t] (ত(	t1
কং	Ônushshôr	Final velar nasal	Ng
কঃ	Bishôrgo	Adds voiceless breath after vowel	:
কঁ	Chôndrobindu	Nasalises vowel	n1

TABLE 6: MODIFIERS

** In our implementation we omitted these modifiers.

Obviously I modify the rules of translation given from research center to simplification of tokenization of festival. I reduce the “%” sign as it is uses a direct symbol in festival. On the other hand I use more then one letter to represent a Bangla letter instead of using capital letter that the research centre does.

The Algorithm I designed for translation given bellow.

3.3 ALGORITHM FOR TRANSLITERATION

```
[start comments]
we have to take one array containing all the Unicode of Bangla letter. And another Array will contain translated English letters of that Unicode in same index value.
[end comments]
Set and Assign the Array of Unicode of Bangla Letters;
Set and Assign the Array of English Letters of Corresponding Unicode;
Open Unicode File as Read;
Open Text File as WriteText;
Do While End of Unicode File
    Set IndexForEnglishArray;
    Set Index=1;
    Do While Match with Unicode Array
        if Read.UnicodeLetter =  UnicodeArray[Index] Then
            IndexForEnglishArray = Index
        End If
    Index = Index + 1;
    End Do While
    Write EnglishLetterArray[IndexForEnglishArray] to WriteText file.
    Next Unicode Letter Read;
End Do While
```

The code is written in Java because UTF-8 format text access is very simple in Java.

SAMPLE INPUT

আমরাই একমাত্র জাতি যারা ভাষার জন্য যুদ্ধ করেছি। ছিনিয়ে এনেছি মাতৃভাষা বাংলাকে। ভাষার এই যুদ্ধ এখানেই থেমে থাকেনি। শুরু হয়েছে বাংলাকে ভিন্ন আঙ্গিকে প্রতিষ্ঠিত করার জন্য প্রযুক্তির সাথে যুদ্ধ।

সেই যুদ্ধে ঝাঁপিয়ে পড়েছে ড: মুমিতের সাথে আরো অনেকে।

SAMPLE OUTPUT

aamraai ekmaatar jaatai zaaraa bhaashaar jnz zudado krechi. chiniye enechi maataribhaashaa baanglaake. bhaashaar ei zudado ekhaanei toeme toaakeni. shuru hyeche baanglaake bhinn aanggike prtaishthita kraar jnz przuktair saatoe zudado. sei zudadoe jhaapiye praeche d mumitaer saatoe aareaa aneke.

Now we have to extract unique word from these sentences. An approximately calculation, 200 sentences will carry 1400 new word if it is a part of news letter. Next section I briefly describe how I extract unique word.

CHAPTER IV: PHONETIC ANALYSIS

In this chapter I discuss the method of finding the pronunciation of word. This is either done by a lexicon (a large list of words and their pronunciations) or by some method of letter to sound rules. We use a large list of lexicon.

4.1 DEFINITIONS

In a phonological orthography, a **grapheme** can be equal to its corresponding to **phoneme**. That is how we pronounce a written text.

The root of Bangla language Sanskrit, which is phonetically perfect (i.e. there is very little or almost no discrepancy between written text and pronunciation), Bangla is pronounced almost as it is written. There are 48 phonemes in Bangla language, but there are 5 phonemes (anusker, bisurga, chandrabindu, nio, umma) which are pronounced with the help of other phoneme. They are not pronounced individually. For G2P conversion I have done one-to-one mapping for text to phone. At the first step of transliteration is to Bangla Unicode letter to English letter. The transliteration is given in table before this chapter.

4.2 STEPS OF PHONETIC ANALYSIS

1. Building large amount of lexicon by hand.
2. Building letter-to-sound rules by hand.

4.2.1 BUILDING LARGE AMOUNT OF LEXICON BY HAND

Initially I add some lexicon. Then I add letter to sound rules. As there is no difference between Bangla written text and pronunciation, building a letter to sound rule is much easier by hand, though letter to sound rule can be done automatically for large set of corpus by training. But there is certain exception in (anusker, bisurga, chandrabindu, nio, umma), conjugate cluster, I will deal it by rules in future implementation.

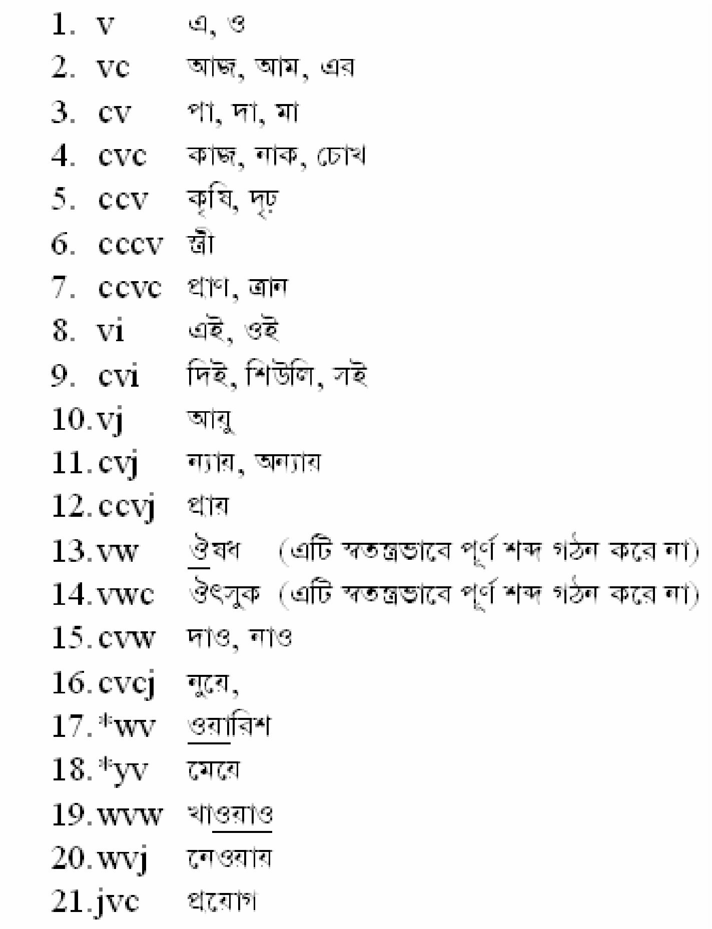
Pronunciation in Festival requires does not require just a list of phones but also a syllabic structure. I already discussed about our phones with their features. The lexicon structure that is basically available in Festival takes both a word and a part of speech (and arbitrary token) to find the given pronunciation. In our Bangla language I developed a large set of lexicon based on our syllabic structure. An example entry of lexicon is

("aapni" n (((aa p) 0) ((n i) 0)))

Explicit marking of syllable, a stress value is also given (0 or 1). Rules of syllabification are given here:

c - Consonant, v-vowel

B (i), q (j), e (w)



1.	v	এ, ও
2.	vc	আজ, আম, এর
3.	cv	পা, দা, মা
4.	cvc	কাজ, নাক, চোখ
5.	ccv	কৃষি, দৃঢ়
6.	cccv	স্ত্রী
7.	ccvc	প্রাণ, ত্রাণ
8.	vi	এই, ওই
9.	cvi	দিই, শিউলি, নই
10.	vj	আয়ু
11.	cvj	ন্যায়, অন্যায়
12.	ccvj	প্রায়
13.	vw	ঔষধ (এটি স্বতন্ত্রভাবে পূর্ণ শব্দ গঠন করে না)
14.	vwc	ঔৎসুক (এটি স্বতন্ত্রভাবে পূর্ণ শব্দ গঠন করে না)
15.	cvw	দাও, নাও
16.	cvcj	নুয়ে,
17.	*wv	ওয়ারিশ
18.	*yv	মেয়ে
19.	wvw	খাওয়াও
20.	wvj	নেওয়ায়
21.	jvc	প্রয়োগ

The most common rules v, vc, cv, cvc, vj, cvj are use for syllabification.

The basic assumption in Festival is that I gather a large number of lexicon entry, might be ten thousand of entries which is using as a standard part to implement of voice. Letter-to-sound rules are used as alternative back up when a word is not explicitly listed. However this is a very flexible way, an explicit lexicon is not necessary in Festival and it might be possible to do much of the work in letter-to-sound rules. In our implementation I include almost 965 lexicons from a text corpus. This lexicon can be added manually or automatically. Here I add it manually by hand. Explicitly I use the syllabification rules.

4.2.2 BUILDING LETTER-TO-SOUND (LTS) RULES BY HAND

In Festival there are letters to sound rules or grapheme to phoneme (G2P) system that allow rules to be written, but Festival also provide a method for building the set for rules automatically which will be often more useful. There is no difference between orthography and its phone set of Bangla Language, so latter-to-sound (LTS) rules can be written by hand. I use LTS rules in my implementation, but it is still a **research issue**. I implement this LTS rules based on my syllabification rules.

4.3 OUT OF VOCABULARY WORDS

It is impossible to list all words in a natural language for general text-to-speech. I need to provide something to pronounce out of vocabulary words i.e large number of names, local language. By default a lexicon in Festival will throw an error if a requested word isn't found. Festival uses *symbolexplode* function. This recursively calls the lexical lookup function on the characters in a word. Each letter appears in the lexicon with its pronunciation (in isolation). But a check is made to ensure I don't recourse for ever. The *symbolexplode* function assumes that letters are single bytes, which may not be true for some case and that function would need to be replace for that cases. Note that I append the syllables of each of the letters in the word. For long words this might be too naive as there could be internal prosodic structure in such a spelling that this method will not allow for. In that case Festival builds letters to be words thus the symbol explosion to happen at the token to word level. This *symbolexplode* function may be the worst case solution. I have to build LTS rule or G2P rule for proper noun (names) and out of vocabulary words. This is also a **research issue**.

CHAPTER V: FESTIVAL

5.1 FUNDAMENTAL COMPONENTS OF FESTIVAL

Festival is designed as a speech synthesis system for at least three levels of user. First of all, those who simply wants high quality speech from arbitrary text with the minimum of effort. Second, those who are developing language systems and wish to include synthesis output. In this case, a certain amount of customization is desired, such as different voices, specific phrasing, dialog types etc. The third level is in developing and testing new synthesis methods.

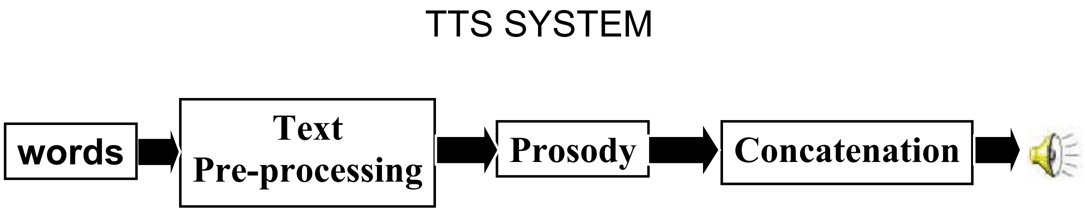


FIGURE 2 : FESTIVAL FUNDAMENTALS

5.2 TEXT PRE-PROCESSING

- Input
    - String of characters (sentence)
- Output
    - String of diphone symbols
- Objective
    - Perform sentence level analysis
        - Punctuation marks
        - Pauses between words
    - Convert all input to corresponding diphones

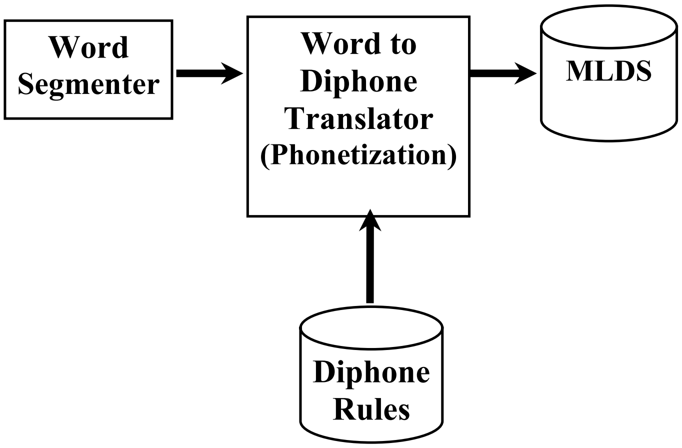


FIGURE 3: TEXT PRE-PROCESSING (BLOCK DIAGRAM)

WORD SEGMENTER

- Divide sentence into word segments
  - Special delimiter to separate segments (i.e. '||')
- Segments can be:
  - A single word
  - An acronym
  - A numeral
- Identify punctuation marks

WORD TO DIPHONE CONVERTER (PHONETIZATION)

- Purpose
  - Translate words to their diphone representations
- Resource
  - Dictionary of words and their diphones (derived from CMU phoneme database)
  - Over 175,000 words supported
- Implementation
  - Binary Search Algorithm in C

    - Start with whole dictionary as search range start index, end index, middle index
    - If target word alphabetically less then middle word, then ignore second half (i.e. end index = middle index) else ignore first half (i.e. start index = middle index)
    - Repeat until word found or range contains zero words
- Advantages
    - Fast search times
        - Search range decreases exponentially with each iteration (max of 1 sec currently)
    - Less complicated to implement
        - Compared to indexing dictionary or
        - Importing the dictionary to an internal structure

THE MULTI-LEVEL DATA STRUCTURE

- Contains all necessary data for the next sub-system:
    - Word
    - Diphone representation
    - Prosodic parameters for each diphone
        - This reflects both word-level and sentence- level prosody
- Allows for modularization

5.3 PROSODY

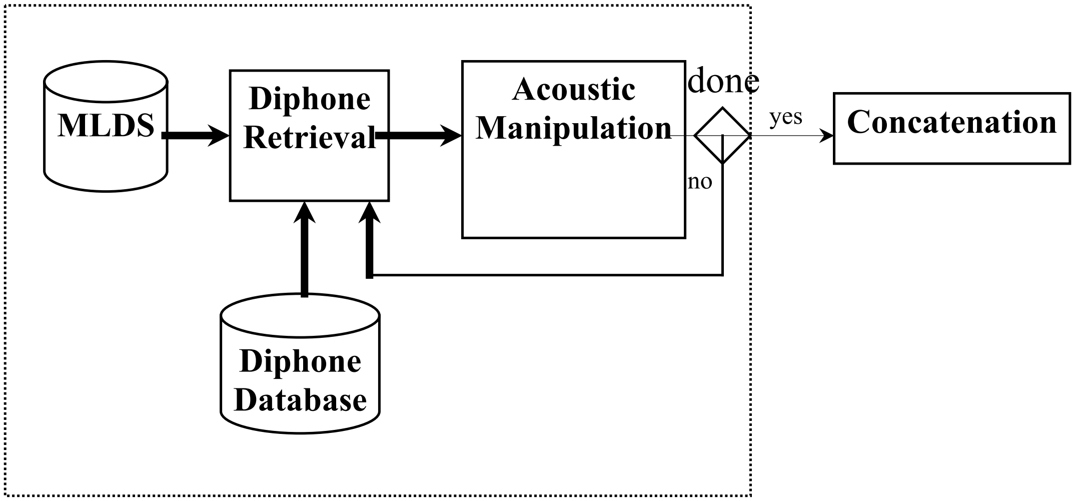


FIGURE 4: INSIDE PROSODY

DIPHONE RETRIEVAL

- Database of recorded diphones
- Every diphone matched with txt file
  - Distinguished by type (CC, CV, VC, VV)
  - References to specific components within waveform
- Store diphone waveform and prosodic parameters in variables

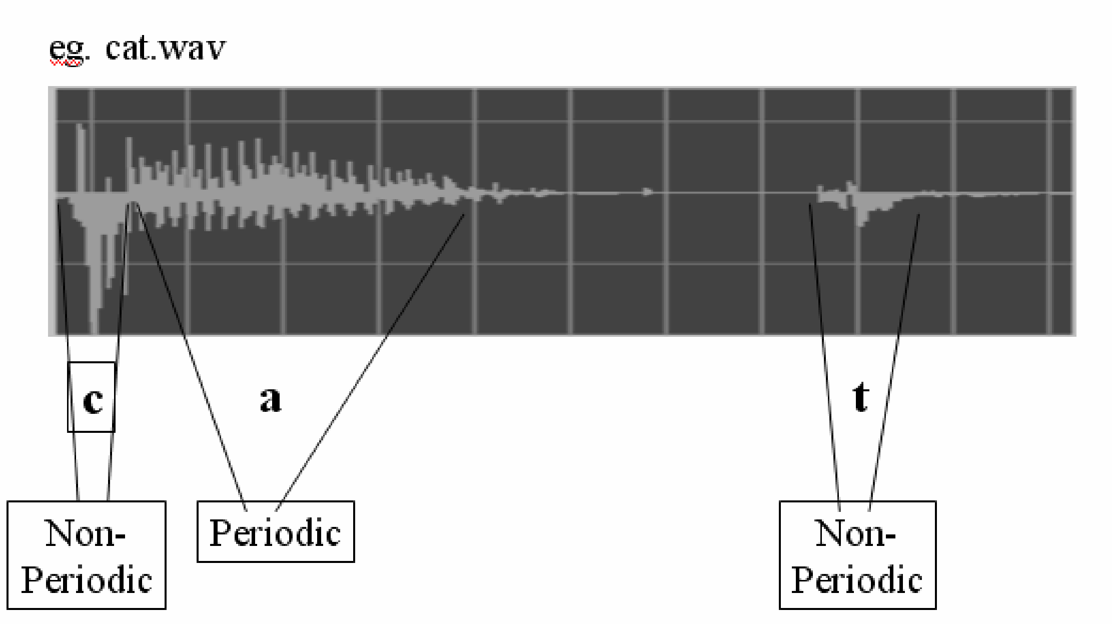


FIGURE 5: PROPERTY OF SPEECH SIGNAL

ACOUSTIC MANIPULATION

- Recognizes wave files (.WAV)
  - load, play, write
- Vast array of signal processing tools
- Built-in functions
- Ease of debugging
- GUI-capable

PITCH/DURATION/AMPLITUDE ALTERATION

- Pitch – vowels only
- As pitch increases, pitch period shrinks
- As pitch decreases, pitch period expands

- Need to alter length between pitch marks in order to alter pitch of speech signal

ALTERING PITCH

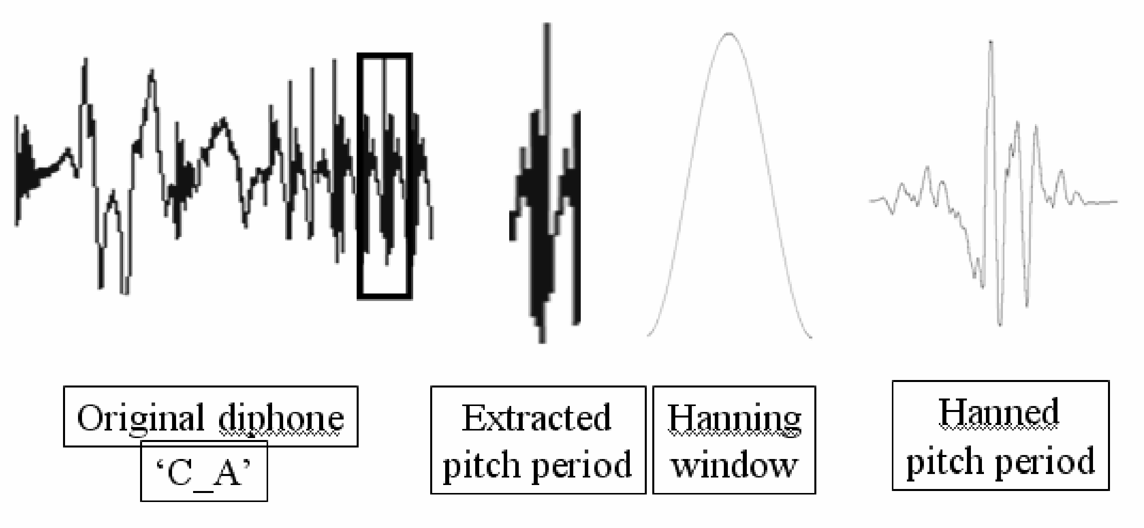


FIGURE 6: PROPERTY OF SPEECH SIGNAL

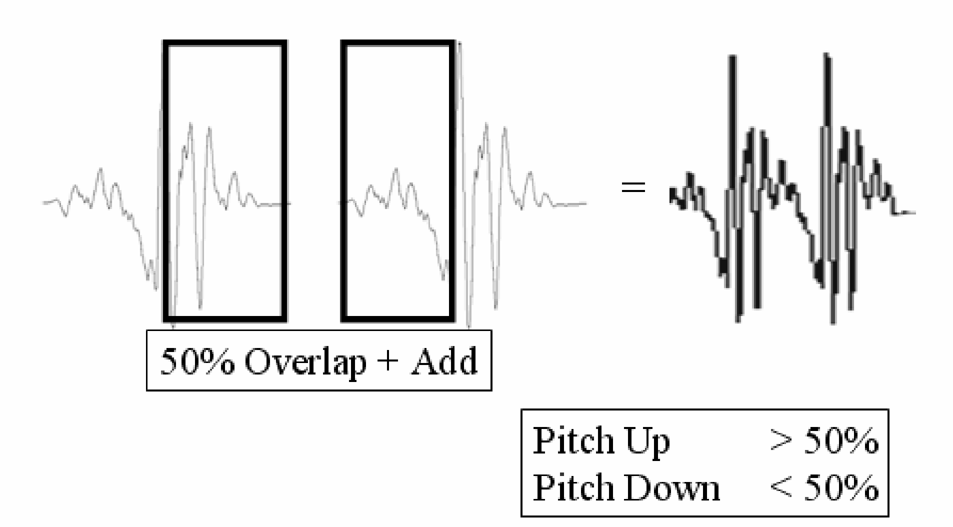


FIGURE 7: PSOLA – PITCH SYNCHRONOUS OVERLAP AND ADD

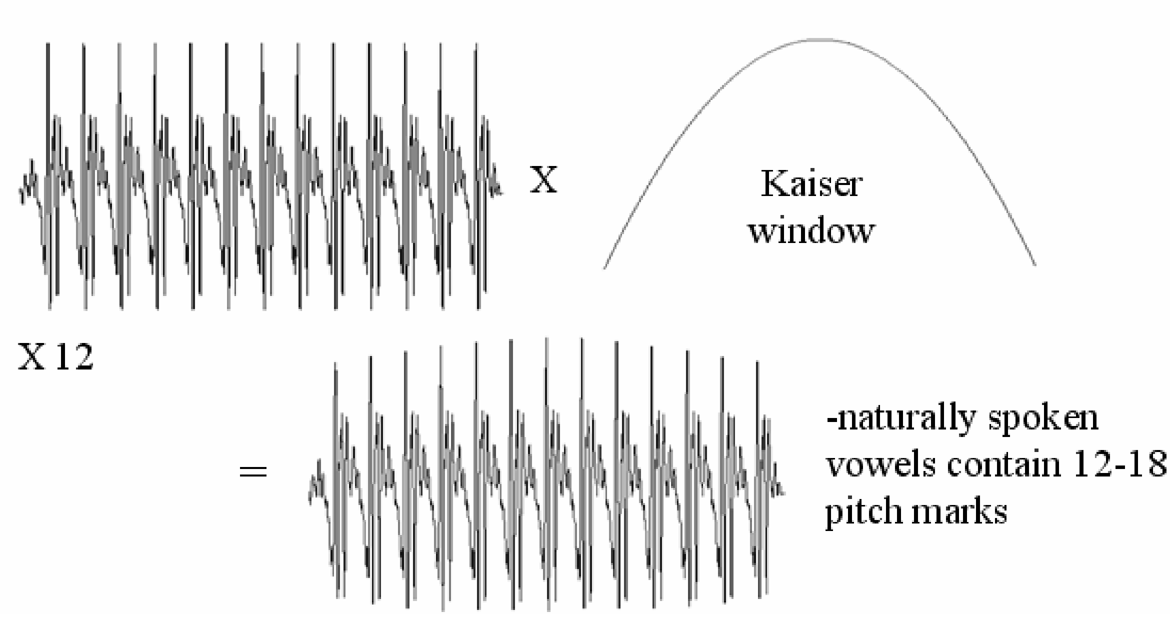


FIGURE 8: PITCH MARKS

ALTERING DURATION

- Increase number of PSOLA iterations (overlaps) to increase duration
- Decrease number of PSOLA iterations (overlaps) to decrease duration

ALTERING AMPLITUDE

- Multiplying the signal by a constant
- If constant > 1, amplitude increase
- If constant < 1, amplitude decrease

5.4 CONCATENATION

- Diphones → Words
- Using PSOLA at the joining ends
- Ensures smooth transition

  Words → Sentences
- Straight joining at the end points due to presence of pauses

CHAPTER VI: WORD-DIPHONE BANK GENERATION

The function of Text-To-Speech (কথা) system is to convert the given text to a spoken waveform. The TTS (কথা) system is based on the concatenation of basic speech units, diphones, using Festival. The input text is transformed into its spoken equivalent by a series of modules. These modules, constituting the TTS (কথা) system are described in detail.

Basically I divided our total thesis in two parts. One part is to generate word bank with its corresponding diphones. This portion is actually developer portion. And the rest portion is the user portion. In this portion user just give input Bangla text to its Unicode form and the festival generate Bangla voice to its corresponding input.

Now from chapter II, I am describing the steps to generate Word-Diphone bank.

Suppose I am going to generate 200 sentences of Word-Diphone bank. I am using a confusion word "Word-Diphone" which actually means the diphones extracted from word.

The Block Diagram is given in figure 1. Eventually I gave a text file containing 200 of Unicode Bangla sentences. To avoid complexity, lets gave normalized text. It follows the following steps to generate Word-Diphone Bank.

1. Translate normalized Bangla Unicode to corresponding English letter.
2. Extract unique word from translated sentences.
3. Impose post lexicon rules with basic lexical for syllable extraction of those unique words so that festival can generate diphones from that syllable.
4. Record voice for corresponding diphones using festival. It generates the diphone bank automatically in wave format.

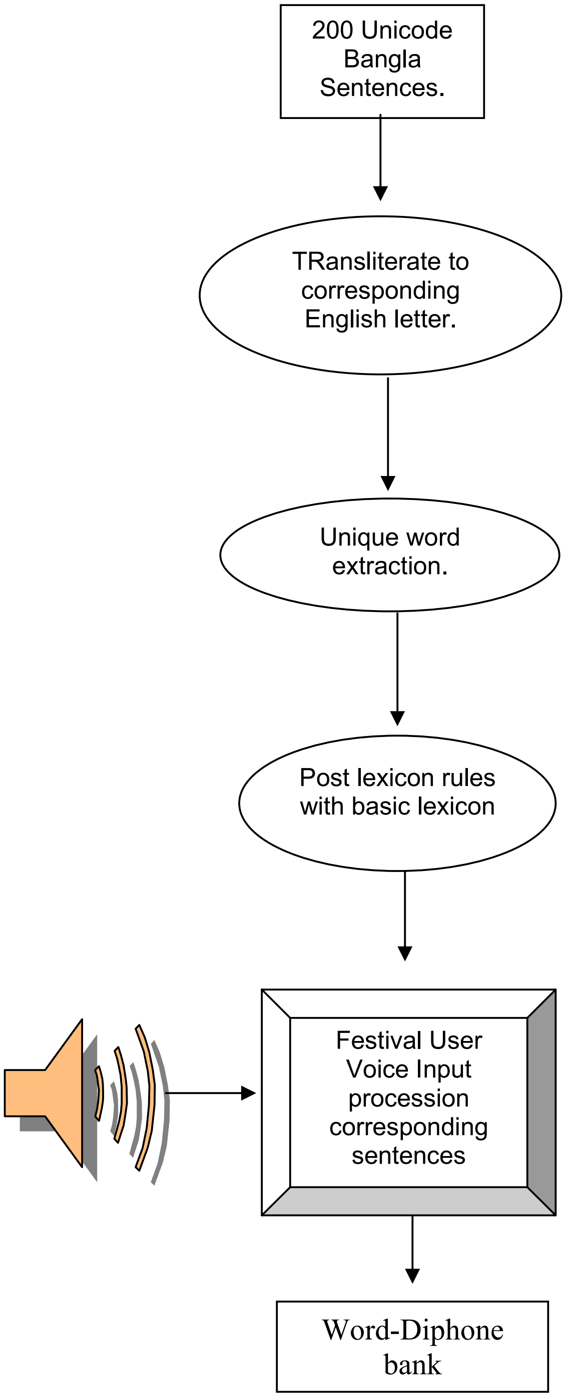

200 Unicode Bangla Sentences.
TRansliterate to corresponding English letter.
Unique word extraction.
Post lexicon rules with basic lexicon
Festival User Voice Input procession corresponding sentences
Word-Diphone bank

FIGURE 9: WORD-DIPHONE BANK GENERATION

6.1 TRANSLITERATION

I discussed huge in Chapter 3 in section 3.2.

6.2 UNIQUE WORD EXTRACTION

It is neither so simple nor complex but bit tricky to get the actual unique word. The algorithm should be recursive because I had to check with the immediate

Input also.

The Algorithm is given bellow.

6.3 ALGORITHM FOR UNIQUE WORD EXTRACTION

We divided the total Algorithm in two modules. First module is making a list of all words and save it in a file. And the second module is just reading from that file and compare it with another file that it going to insert as a unique word.

Module 1: (WordListing)

```
Set ReadFile as file stream to read input;
Set WriteFile as file stream to write output list;
Do While End of ReadFile
        Read Word and Save it to WriteFile as list.
End While
```

Module 2: (Compare)

```
Set ReadFile as file stream to read list file;
Set WriteFile as file stream to write output;
Do While End of ReadFile
        Set CurrentWord = read word from ReadFile
        Compare it with WriteFile saved words.
        If Compare result false then save it to WriteFile
        Otherwise go for next
End While
```

SAMPLE INPUT

aapni hytaobaa kmpiutaar er saamne bse aachen. kichu kraar paacchenaa. haatae kichu smy aache ki? tabe ektu aaddaa mere smytaake kaataale kemn hy? baahire brishti hcche tabe ek kaap caa niye ene kmpiutaare aaddaa maarle kemn hy?

OUTPUT OF FIRST MODULE

```
aapni
hytaobaa
kmpiutaar
er
saamne
bse
aachen
kichu
kraar
paacchenaa
haatae
kichu
smy
aache
ki
tabe
ektu
aaddaa
mere
smytaake
kaataale
kemn
hy
baahire
brishti
hcche
tabe
ek
kaap
caa
niye
ene
kmpiutaare
aaddaa
maarle
kemn
hy
```

OUTPUT OF SECOND MODULE

```
aapni
hytaobaa
kmpiutaar
er
saamne
bse
aachen
kichu
kraar
paacchenaa
haatae
smy
aache
ki
tabe
ektu
aaddaa
mere
smytaake
kaataale
kemn
hy
baahire
brishti
hcche
tabe
ek
kaap
caa
niye
ene
kmpiutaare
maarle
```

I wrote here the basic algorithm for comparing. But the original algorithm I used is bit complex. For 200 sentences it may work faster but for 2000 word it may takes some time.

Then we move for a manual process which known **Post Lexicon Rules.**

6.4 POST LEXICON RULES

In the rules of post lexicon, we just set some rules of stretch of lexicon by dividing in syllable which in Festival lexicon input file format. Here I give some sample format of rules. I did it manually given rules in book “Dhani Biggan O Dhanitotto”[2].

```
(lex.add.entry '("aapni" nn (((aapni) 0))))
(lex.add.entry '("hytaobaa" nn (((hytaobaa) 0))))
(lex.add.entry '("kmpiutaar" nn (((kmpiutaar) 0))))
(lex.add.entry '("saamne" nn (((saamne) 0))))
(lex.add.entry '("bse" nn (((bse) 0))))
(lex.add.entry '("aachen" nn (((aachen) 0))))
(lex.add.entry '("kichu" nn (((kichu) 0))))
(lex.add.entry '("kraar" nn (((kraar) 0))))
(lex.add.entry '("paacchenaa" nn (((paacchenaa) 0))))
(lex.add.entry '("haatae" nn (((haatae) 0))))
(lex.add.entry '("smy" nn (((smy) 0))))
(lex.add.entry '("tabe" nn (((tabe) 0))))
(lex.add.entry '("ektu" nn (((ektu) 0))))
(lex.add.entry '("aaddaa" nn (((aaddaa) 0))))
(lex.add.entry '("mere" nn (((mere) 0))))
(lex.add.entry '("smytaake" nn (((smytaake) 0))))
(lex.add.entry '("kaataale" nn (((kaataale) 0))))
(lex.add.entry '("kemn" nn (((kemn) 0))))
(lex.add.entry '("baahire" nn (((baahire) 0))))
(lex.add.entry '("brishti" nn (((brishti) 0))))
(lex.add.entry '("hcche" nn (((hcche) 0))))
(lex.add.entry '("kaap" nn (((kaap) 0))))
(lex.add.entry '("caa" nn (((caa) 0))))
(lex.add.entry '("niye" nn (((niye) 0))))
(lex.add.entry '("ene" nn (((ene) 0))))
(lex.add.entry '("kmpiutaare" nn (((kmpiutaare) 0))))
(lex.add.entry '("maarle" nn (((maarle) 0))))
```

6.5 RECORDING DIPHONES

As Festival written in C++, internally some Scheme programming is used here. As Scheme Programming is totally new for me, it takes some time to match with its environment.

By default, Festival gives some seconds for each sentence to record. This second calculate by Festival own. But its very difficult to sync with that seconds while recording.

To avoid that sync problem we can use another third party utility PRAAT [3]. It is simple wave file recording software in mono mode. Here I give some screen shot of PRAAT while recording.

The visual output of wave for sentence "aapni hytaobaa kmpiutaar er saamne bse aachen." is given bellow.

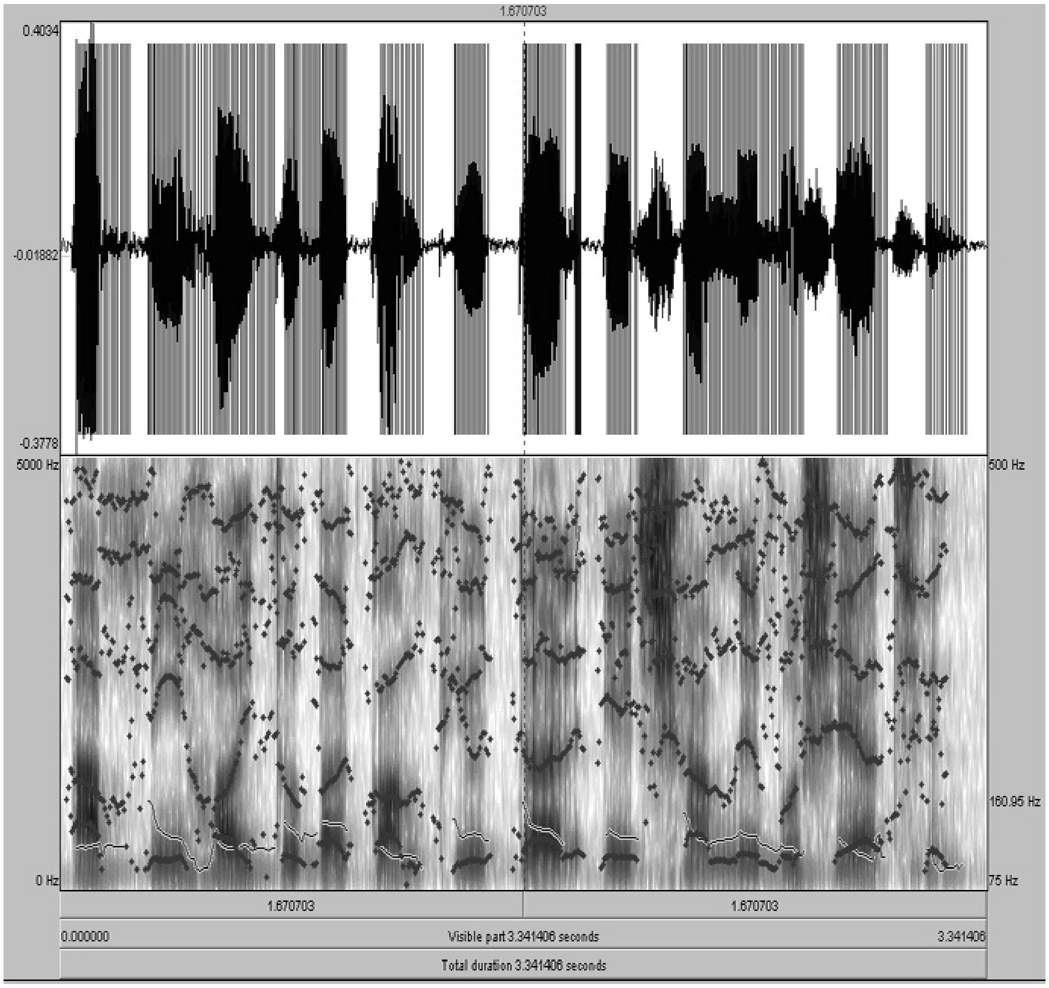


FIGURE 10: PRAAT VISUAL OUTPUT

And the info of analysis of that wave using PRAAT is given here.

Object: Sound sen001

Date: Fri Apr 21 00:49:00 2006

Time domain:

Starting time: 0 seconds

Finishing time: 3.34140589569161 seconds

```
  Total duration: 3.34140589569161 seconds
Time sampling:
  Number of samples: 73678
  Sampling period: 4.5351473922902495e-05 seconds
  Sampling frequency: 22050 Hz
  First sample centred at: 2.2675736961451248e-05 seconds
Amplitude:
  Minimum: -0.37780762 Pascal
  Maximum: 0.40344238 Pascal
  Mean: -0.00010667815 Pascal
  Root-mean-square: 0.057758328 Pascal
Total energy: 0.011147012 Pascal^2 sec (energy in air: 2.786753e-05 Joule/m^2)
Mean power (intensity) in air: 8.3400612e-06 Watt/m^2 = 69.21 dB
Standard deviation: 0.057758622 Pascal
```

But it is also difficult to match with the time given by Festival and the recorded time of PRAAT. If the difference is to high, the result causes missing diphone.

To generate diphones, Festival used its own algorithm. At first it sliced the wave of sentence to its contained words. Then the wave of word slice again by syllabification rules that we give in post rules of lexicon. Then the diphones is generated by the rules of stretch given the stretch rules of post lexicon file.

Thus I finished the generation of word-diphone bank.

Now I describe how we can use festival to recording diphones.

CHAPTER VII: FURURE IMPLEMENTATION

In preliminary stage in my implementation, Lots of issues undiscovered yet. I have to work out on the following issues:

1. **Text Analysis:** Text normalization using scheme for larger context.
2. **Phonetic Analysis:**
   1. Automatic lexicon entries instead of adding manually.
      Find out LTS or G2P rule.
3. **Word-Diphone Bank Generation:** The way I generate the word diphone bank is not the actual efficient way. To make it more accurate, I have to do studio work which was impossible in this short time.

REFERENCES

1. Festival Speech Synthesis, Speech Tools & documentation, http://www.festival.org/
2. Sproat, R., Black, A., Chen, S., Kumar, S., Ostendorf, M., and Richards, C. 2001. Normalization of Non-standard Words, Computer Speech and Language, 15(3):287-333
3. http://www.clsp.jhu.edu/ws99/projects/normal/slides/intro/nswintro.pdf
4. Phoneme set and their features, (1) Naira Khan, (2) Basa Bigganer Kotha by Daniul Haque, (3) Dhani Biggan O Dhanitotto by Abdul Hai
5. Bengali script – Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Bengali_script
6. Jurafskey Lectures, Spring 2006, www.stanford.edu/class/cs224s/
7. Hindi Text Normalization, K. Panchapagesan, Partha Pratim Talukdar, N. Sridhar Krishna, Kalika Bali, A. G. Ramakrishnan, Hewlett-Packard Labs India, 24 Salarpuria Arena, Hosur Road, Bangalore, India. Email: {*partha.talukdar, nsridhar, kalika}@hp.com* Indian Institute of Science Bangalore, India. Email: *ramkiag@ee.iisc.ernet.in* Birla Institute of Technology & Science Pilani, Rajasthan, India Email: *panchapagesan.k@gmail.com*
8. Rules of syllabification (Ref: Dhani Biggan by MD. Abdul Hay, pg-152)
9. Wagon description, Speech tools documentation. http://www.festival.org/
10. Kiswahili TTS system, www.llsti.org

10. IMPLEMENTATION OF INTONATION PATTERN IN BENGALI TEXT TO SPEECH SYNTHESIS, AN APPROACH, Asok Bandyopadhyay, Shyamal Kr. Das Mandal, Barnali Pal, Mridusmita Mitra Speech& Signal Processing Group ,ER&DCI,Calcutta Plot E2/1,Block GP,Sector V,Saltlake ,Kolkata-700 091

11. Third party utility for sound recording. http://www.praat.org/