TEXT TO SPEECH SYSTEM FOR KONKANI (GOAN) LANGUAGE

Sangam P. Borkar
M.E. (Electronics)Dissertation
Guided by
Prof. S. P. Patil
Head of Electronics Department
Rajarambapu Institute of Technology
Sakharale, Islampur, Maharashtra, India

ABSTRACT

A text to speech (TTS) synthesizer is a computer based system that should be able to read any text aloud. The TTS systems are commercially available in English, and in some of the Indian languages like Hindi, Tamil, and Urdu etc. Till now, the text to speech system had not been developed for Konkani language. This is the first TTS system developed for the Konkani (Goan) language. Concatenation technique is used to develop this system. A database of more than one thousand words in the Konkani language is prepared and these words can be read directly using the system. For reading other words Concatenation technique is used.

1. INTRODUCTION

A text to speech (TTS) synthesizer is a computer based system that should be able to read any text aloud [1]. Some systems that simply concatenate isolated words or parts or sentences are denoted as voice response systems. These systems are only applicable when a limited vocabulary is required, and when sentences to be pronounced have a very restricted structure, as in the case for the announcement of arrivals of train on a railway station for instance. In the context of TTS synthesis, it is impossible to record and store all the words of the language. It is thus more suitable to define TTS as the automatic production of speech [1].

2. PRESENT PRACTICES USED IN THE TEXT TO SPEECH SYSTEM

Traditionally, text to speech system converts input text into voice by a set of manually delivered rules for voice synthesis. While these systems can achieve a high level of intelligibility, they typically sound unnatural. The process of deriving these rules is not only intensive but also difficult to generalize to a new language, a new voice, or a new speech style [2].

For speech generation, there are two main methods used. These methods are format synthesis and concatenation synthesis [2]. The format synthesizer uses a simple model of speech production and a set of rules to generate speech. While these systems can achieve high intelligibility, their naturalness is typically low, since it is very difficult to accurately describe the process of speech generation in a set of rules. In recent years, data driven approaches such as concatenation synthesis has achieved a higher degree of naturalness. Format synthesizers may sound smoother than concatenation synthesizers because they do not suffer from the distortion encountered at the concatenation point. To reduce this distortion concatenation synthesizers select their units from carrier sentences or monotone speech.

The text to speech systems are commercially available in English, and in some of the Indian languages like Hindi, Tamil, and Urdu etc. Till now, the text to speech system had not been developed for Konkani language. This is the first TTS system developed for the Konkani (Goan) language.

3. ARCHITECTURE OF TTS

Speech synthesis involves algorithmically converting an input text into speech waveforms and some previously coded speech data. Figure1

introduces the functional diagram of a very general TTS synthesizer [4].

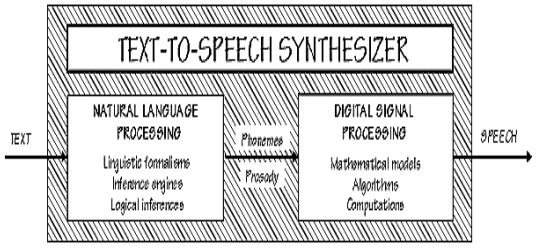


Figure 1. General functional diagram of TTS system [4]

As for human reading, the Text to Speech system comprises of: (i) Natural Language Processing module (NLP): It is capable of producing a phonetic transcription of the text read, together with the desired intonation and rhythm (often termed as prosody), and (ii) Digital Signal Processing module (DSP) : It transforms the symbolic information it receives into speech.

3. ISSUES IN KONKANI LANGUAGE

3.1 Konkani Script

Konkani text is written in the Devanagari script. The alphabets used in the devanagari script are scientific and well organized. They are divided into two groups: (1) Vowels and (2) Consonants.

Vowel:
There are twelve vowel found in Devanagari language. Vowels have two forms, the independent form (the 'swaras') and the dependent form (the 'mAtrAs').
The independent form vowels are 'stand alone'. These forms are used when the vowels are pronounced in isolation, unattached and unassociated with any consonant. Figure 2. gives the list of the Devanagari vowels.

a	aa	i	ee	u	oo	e	ai	o	au	aM	aH
अ	आ	इ	ई	उ	ऊ	ए	ऐ	ओ	औ	अं	अः

Figure 2. Devanagari Vowels

The dependent form vowels are always attached to consonants. When a vowels is pronounced associated with a consonant, the dependent form of that vowel (the 'mAtrA') is used.

Consonant :
Devanagari script has about 36 consonants. Out of these 36 consonants, first 30 is divided into 6 groups. Each group has five letters (sounds) and these sounds, in turn, are divided into three other subgroups {voiced, unvoiced and nasal). The last letter in each group requires 'nasal' pronunciation and is called 'anuNasik' (nAsikA=nose). Figure 3 gives the list of the Devanagari consonants.

k: - Group	k:	K;	g;	G;	V
c; - Group	c;	%	j;	z;	J;
! - Group	!	@	#	$	[;
t; - Group	t;	q;	d	Q;	n;
p; - Group	p;	f:	b;	B;	m;
y; - Group	y;	r	l	v;	x;
Other	{;	s;	h	L	Z;
	N				

Figure 3. Devanagari Consonants

4. IMPLEMENTATION OF TEXT TO SPEECH SYSTEM

4.1 Implementing Steps

Following are the steps taken in order to implement this TTS system.

4.1.1 Study of various Devanagari Fonts

Only on the basis of the chosen font, the ASCII value of the various characters (Vowel and Consonants) and in turn the words can be found out. The comparison of various fonts revealed that Nutan was the best font that can be used for the project.

4.1.2 Sound Recording and Elimination Of Noise

As this project is a text to speech converter, it has to convert the input text fed to it into speech. In order to do so it was necessary that a sound file is created for each and every character of Konkani language, so that when any character is typed the system will search for its sound file and read out the text aloud. Figure 4. shows the

wave file with noise (unwanted signal) for the recorded word **k:uldev;I.**

The noise from the recorded voice signal need to be eliminated which will result in a pure voice signal. Figure 5 shows the noise free signal for the recorded word **k:uldev;I**

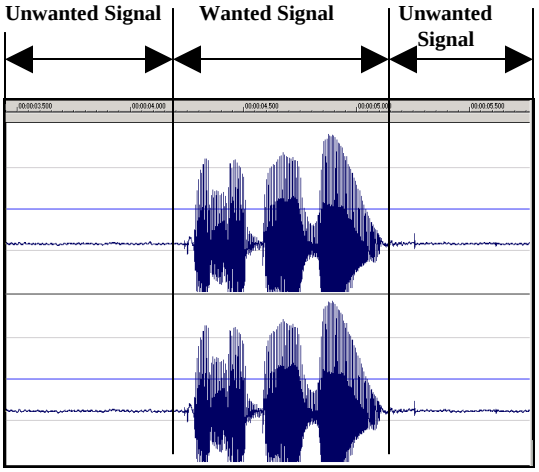


Figure 4. Wave file with noise (unwanted signal) for the recorded word **k:uldev;I.**

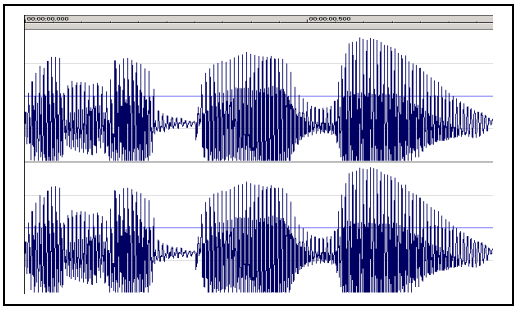


Figure 5. Noise free signal for the recorded word **k:uldev;I**

4.1.3 File Naming by using ASCII codes

The recorded sound files are then named and stored by using the ASCII values of the keys that needs to be pressed for typing that character. For example the sound file of 'a''"'" ' is named as 97 because the character is obtained by pressing the key 'a' which has a ASCII value of 97. Similarly the sound file of '"k: ' is named as '10758' since the character 'k:' is obtained by pressing two keys 'k' and ':' which have the ASCII values 107 and 58 respectively. All the sound files recorded are named and stored in the similar way.

5. SOFTWARE DESIGN

5.1 Algorithm for playing a Complex word

Step 1 : Start

Step 2 : Enter any word
Step 3 : Collect ASCII of entered word.

Step 4 **:** Play the files collected in the table clear the table ,after playing all files

Step 5 **:** If ASCII=" **i** " Goto Step 6
Else Goto Step 7

Step 6 :
6**.**1 Get next ASCII and store it in variable "aa" i.e. aa=aa & ASCII

6.2 Get next ASCII

6.3 If ASCII = **k" , K ,g .....** (i.e. any character) goto step 6.4 Else goto step 6.5

6.4 The word is a " jod-akshar"
If ASCII = "Defaultor"
{ Update last entry in the table & keep collecting remaining ASCII values till complete character is formed. }
Else
{ Without updating the table ,keep collecting next ASCII values , till complete character is formed. }

6.5 The word is not a jod-akshar
keep collecting next ASCII values till complete character is formed .

6.6 Store the collected ASCII sequence of a " complete character " in "aa" into the table .
Go to step 4

Step 7 **:** If ASCII ="Full character " (i.e. **! ,@,#.....w...**) Goto Step 7.1 Else Goto Step 8

7.1 If ASCII= " **w** " & next ASCII=" ***** "
{ Then don't update database (to neglect effect " ***** ") }
Else

{ If next ASCII= ” * “{Then update database }
}

7.2 If next ASCII = ” < “
{ /complete character belongs to “jod-akshar”/
Update last in table
}
End If

7.3 keep collecting ASCII values till complete character is formed
- Store the collected ASCII sequence of a “ complete character “ in “aa” into the table
- go to step 4
Step 8 **:** If ASCII=”half character “ **(k” , K ,g , . .)** Goto step 8.1 Else Goto step 8.2

8.1 If next ASCII = **k” , K ,g**
{ / it is a jod-akshar/ Update database }
Else
keep collecting ASCII values till complete character is formed.

8.2 Store the collected ASCII sequence of a “ complete character “ into the table
- go to step 4

Step 9 **:** Stop

6. OUTPUT & CONCLUSION

6.1 Playing a Single word whose wave file is present in the database

The wave files for more than 1000 commonly used words in Konkani are already prepared and are stored in their pure form i.e. noise free form in the “sample” folder. The database of these more than 1000 words with their ASCII values is also prepared using Microsoft Access. Figure 6. gives the GUI view of the output when a simple word **k:;ek:[;I** is typed in the text box of the GUI and is played. Figure 7. gives the time domain representation of the wave file for the simple word **k:;ek:[;I .**

Figure 6. The GUI view of the output when a simple word **k:;ek:[;I** is typed

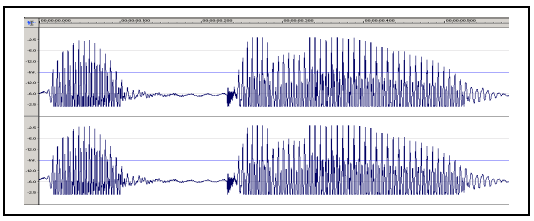


Figure 7. The time domain representation of the wave file for the simple word **k:;ek:[;I .**

6.2 Playing a Single word whose wave file is not present in the database

If the wave file of a particular word in Konkani is not present in the database then concatenation technique is used to play the word. In this the word is first broken down into its characters and then the individual characters are played. Figure 8 gives the GUI view of the output when a word **s;;Q;n;;** is typed in the text box of the GUI. Since the word is not present in the database a message is displayed which indicates that the word is not present in the database and hence concatenation technique will be used.

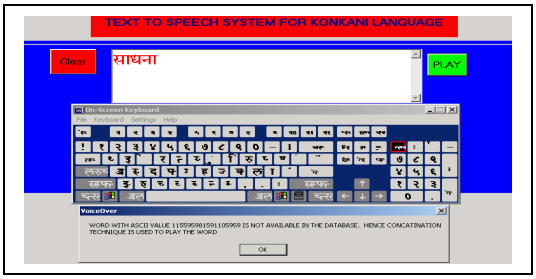


Figure 8. GUI view of the output when a word **s;;Q;n;;** is typed in the text box of the GUI

Since the word **s;;Q;n;;** is not present in the database, it is broken down into its characters as **s;;** , **Q;** and **n;;** . The ASCII values of the se

characters are stored in the Search Engine of Microsoft Access and then they are played on the speaker. Figure 9 give the view of the Search Engine data of the characters **s;; , Q;** and **n;;.** The ASCII value of the characters **s;;** is 1155959, of **Q;** is 8159 and of **n;;** is 1105959.

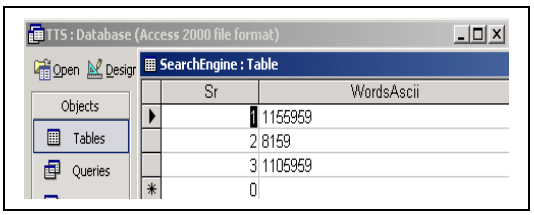
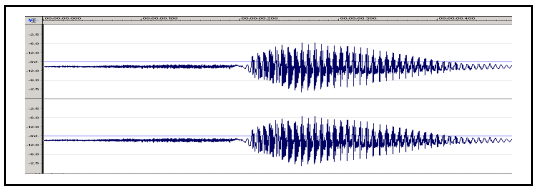
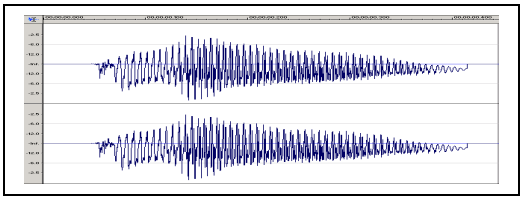


Figure 9. The view of the Search Engine data of the characters **s;; , Q;** and **n;;.**

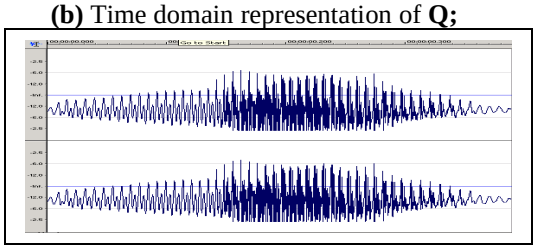
The figures 10(a), (b), (c) gives the time domain representation of the wave files for the characters **s;; , Q;** and **n;;.**



(a) Time domain representation of **s;;**



(b) Time domain representation of **Q;**



(c) Time domain representation of **n;;**

Figure 10. Time domain representation of the wave file for the characters **s;;, Q;, n;;.**

7. CONCLUSION

A speech synthesis system has been designed and implemented for Konkani Language. A database for more than 1000 commonly used words in Konkani language is made. These words can be played directly using this TTS system. The wave files are recorded in the students own voice Around 3000 wave files consisting of Vowels, Characters, Barakhadi and half Characters are prepared. For playing any complex word (Jod-Akshar) which is not present in the database Concatenation technique is used. The Synthesizer is coded using VB programming language platform.

8. REFERENCES

[1] Anupam Basu ,Debashish Sen, Shiraj Sen and Soumen Chakraborty "An Indian language speech synthesizer – techniques and application" IIT Kharagpur, pages 17-19, Dec 2003.

[2] Xuedong Huang, Alex Acero, Jin Adcode "Whistler: a trainable text-to-speech system".

[3] Kiyohiro Shikano "Free software toolkit for Japanese large vocabulary continuous speech Recognition".

[4] Thierry Dutoit "A short introduction to text to speech synthesis".

[5] K. Kiran Kumar, K. sreenivasa Rao and B. Yegnanarayana "Duration knowledge for text – to – speech system for telgu".

[6] Sireesh Sharma, R.K.V.S. Raman, S.Shridevi, Rekha Thomas "Matrubhasha – An integrated speech framework for Indian Languages".

[7] Roger Tucker "Local language Speech Technology" initiation, 2003.

[8] Kranti Goyal " Speech Synthesis for hindi language" M.Tech dissertation, Department of computer science engineering, IIT, Bombay.

[9] Klatt O.H. Review of Text- to- Speech conversion for English pages 737-793, 1987 .

[10] Douglas O'Shaughnessy. "Speech Communications: Human and machine." Universities Press, second edition, 2001.

[11] Vivekananda Shetty. ” Voice interaction system”. M.tech dissertion, Indian Institute Of Technology, Bombay, 1990.

[12] F H. Lochovsky D L. Lee.” Voice response systems.” ACM Comput. Surv., 15(4):351-374, 1983.

[13] Klatt D.H. “Review of text-to-speech conversion for English.” pages 737-793, 1987.

[14] Robert Edward Donovan. “Trainable speech synthesis”. PhD thesis, Cambridge University Engineering Department, 1996.

[15] Robert D.Rodman.” Computer Speech technology”. Artech House, 1999.

[16] I.H.Witten.” Principal of computer speech”. Academic Press, INC., 1982.