Text To Speech for Bangla Language using Festival

Firoj Alam
BRAC University, Bangladesh
firojalam04@yahoo.com

Promila Kanti Nath
BRAC University, Bangladesh
bappinath@hotmail.com

Dr. Mumit Khan
BRAC University, Bangladesh
mumit@bracuniversity.net

ABSTRACT

In this paper, we present a Text to Speech (TTS) synthesis system for Bangla language using the open-source Festival TTS engine. Festival is a complete TTS synthesis system, with components supporting front-end processing of the input text, language modeling, and speech synthesis using its signal processing module. The Bangla TTS system proposed here, creates the voice data for festival, and additionally extends festival using its embedded scheme scripting interface to incorporate Bangla language support. Festival is a concatenative TTS system using diphone or other unit selection speech units. Our TTS implementation uses two different kinds of these concatenative methods supported in Festival: unit selection and multisyn unit selection. The function of a Text-to-Speech system is to convert some language text into its spoken equivalent by a series of modules. These modules, constituting the TTS system are described in detail which is very much helpful for future development. Finally, the quality of synthesized speech is assessed in terms of acceptability and intelligibility.

Key Words: TTS, Festival, Speech synthesis.

1. Introduction

Speech is one of the most vital forms of communication in our everyday life. So it is natural for people to expect to be able to carry out spoken dialogue with computers. This involves the integration of speech technology and language technology. A freely available (and hopefully open-source) TTS system for Bangla language can greatly aid the human computer interaction: the possibilities are endless – such a system can help overcome the literacy barrier of the common masses, empower the visually impaired population, increase the possibilities of improved man-machine interaction through on-line newspaper reading from the internet and enhancing other information systems. Festival[1] is a complete TTS synthesis system, with language modeling, and speech synthesis engine. The language model supports all language processing tasks. For example document analysis, text analysis, and phonological processing. We used festival to develop Text to Speech for Bangla language by providing language processing parameter in language model part and recorded speech in speech engine. Here we described the methodology and the implementation of a Text to Speech system for Bangla based on the Festival TTS engine. At the end of the paper, assessment results of the TTS system are given and some promising directions for future work are mentioned.

Organization of the paper are as follows. Section 2 discusses about related works. Section 3 discusses about methodology. Section 4 discusses results. Then section 5 discusses about future implementation. After that in section 6 we discusses conclusion.

2. Related works:

Following are the related works for the Bangla Text To Speech. Several attempts were made in the past, where different aspects of a Bangla TTS system were covered [2][3][4][5]. In [2] authors described about different modules (optimal text selection, G2P conversion, automatic segmentaiton tools) in detail and experiment results of the different module have shown. In [3] significant amount of work done for developing Bangla TTS. Phoneme and partname (similar to diphone) are used to develop voice database and ESOLA technique used for concatenation. But quality may suffer for lack of smoothness. In [4] authors showed some practical applications with Bangla TTS system using ESNOLA technique. But performance of the output not described. In [5] author showed the pronunciation rule and phoneme to speech synthesizer using formant synthesis technique. None of them have shown the naturalness and inteligibility of the system. This work is done with multisyn unit selection and unit selection technique within festival framework and performance of the intiligibility and naturalness of the system have shown.

3. Methodology

The TTS for Bangla is developed by widely used third party tool Festival. The different phases of the synthesis task are performed by several modules as shown in Figure 1. The text analysis part converts all non standard words to standard words. A grapheme-to-phoneme module produces strings of phonemic symbols based on information in the written text. The problems it addresses are thus typically language dependent. So is the prosodic generator, which assigns pitch and duration values to individual phonemes. Final speech synthesis is performed by concatenative unit selection technique and multisyn unit selection technique. We implemented all of modules by festival tools.

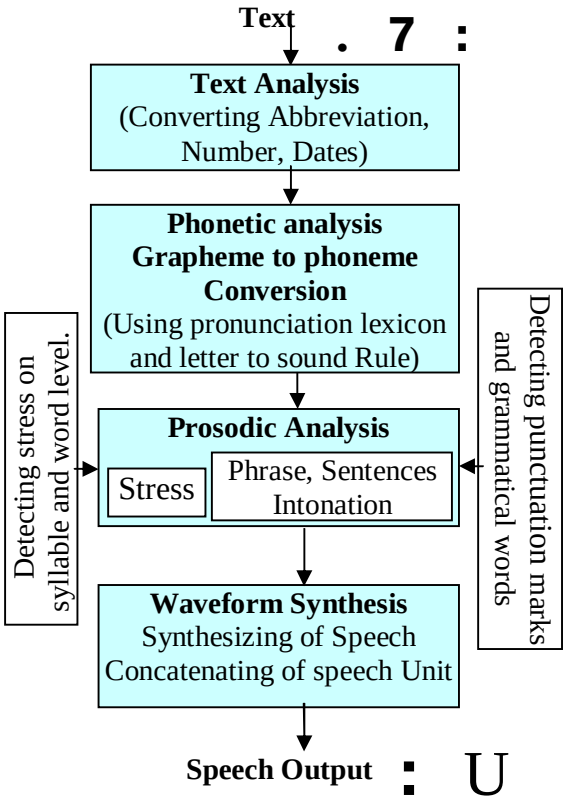

Figure 1 : Archetecture of TTS

3.1 Text analysis

The first step of Text to Speech system is text analysis [6] that means analysis of raw text into pronounceable word. It involves the work on the real text, where many Non-Standard Word (NSW) [7] representations appear, for e.g., numbers (year, time, ordinal, cardinal, floating point), abbreviations, acronyms, currency, dates, URLs. All of these non-standard representations should normalize, or in other words convert to standard words. These NSW should normalize using text normalization and ambiguous token should disambiguate using rules.

3.1.1 Text analysis part in Festival

Festival does not support Unicode directly, so in the first step we transliterated our Unicode text to ASCII code according Bangla phone set [8]. The transliteration table is given in table-1. In our system of text analysis parts we worked on standard words. We identified more than 10 types of NSW in Bangla Language, which in not implemented yet. Some example of NSW in Bangla Language is given in table 2 that can be implemented in future. Now our

system only supports Unicode, not ASCII coded Bangla text. As most of the existing Bangla text is written in ASCII code, so we have a plan to implement it later.

Letter	Transliteration	Letter	Transliteration	Letter	Transliteration	Letter	Transliteration	Letter	Transliteration
অ	a	ঔ	ou	ঝ	jh	দ	da	র	r
আ	aa	ক	k	ঞ	nio	ধ	dh	ল	l
ই	i	খ	kh	ট	t	ন	n	শ	sh
ঈ	ii	গ	g	ঠ	th	প	p	ষ	sh
উ	u	ঘ	gh	ড	d	ফ	ph	স	s
ঊ	uu	ঙ	ng	ঢ	dh	ব	b	হ	h
এ	e	চ	c	ণ	n	ভ	bh	য়	y
ঐ	oi	ছ	ch	ত	ta	ম	m	ড়	ra
ও	o	জ	j	থ	to	য	z	ঢ়	ra

Table 1: Bangla to ASCII transliteration table[1]

3.1.2 Steps of Text Analysis in Festival

Step 1: Split the token: We can split our token based on white-space and punctuation.

- White-space can be viewed as separators.
- Punctuation can separate the raw tokens.
- Festival converts text into: Ordered list of tokens, each with features of white-space, and punctuation.

NSW Category	Written format	Pronunciation	IPA transcription
Cardinal number	৯৫৬৭৪৪৭	নয় পাঁচ ছয় সাত চার চার সাত	noj p˜ac c j sat c ar sat
Ordinal number	১ম	প্রথম	pr t m
Date	০২/০৬/০৬	দুই জুন দুই হাজার ছয়	d ui un d ui ha ar c j
Time	৪:২০ মি:	চারটা বিশ মিনিট	c arta bi minit
Ratio	১:২	এক অনুপাত দুই	ek nupat d ui
Special	৳	টাকা	taka
character			
Acronym	ঢাবি	ঢাকা বিশ্ববিদ্যালয়	daka bi bid d al j
Abbreviation	ডঃ	ডক্টর	doctor

White-space is the most commonly used delimiter between words and is extensively used for tokenization. But using white-space as the only delimiter have some limitation: a token type which allows the occurrence of white-space within the token will not recognize as a single token, but split up into two or more tokens. For example, consider a telephone number ৮৮০ ২ ৯৫৬৭৪৪৭ [880 2 9567447]$_{IPA}$. This can identify as a single token of type ‘telephone number', but if tokenization is exclusively based on white-space, then we end up having 3 tokens. Further, an important limitation is that every token will then have to go through a token identification process that identifies its token type/category.

Step 2: Type identifier: As we explained Bangla Language have more than 10 types of NSW, so each NSW can identify as separate token by token identifier rules. To identify the token we can use scheme regular expression in festival, which is not implemented yet. There is also an ambiguity in abbreviation, and number in Bangla language. We use colon [**:**] for abbreviation as well as middle of two sentences. For example, শিক্ষা প্রতিষ্ঠান বন্ধ: লাগাতার হরতাল ও ১৪৪ ধারার কারনে কানসাটের শিল্প প্রতিষ্ঠানগুলো বন্ধ রয়েছে। ড: [ডক্টর], আ: [আব্দুল]

ikkha prot i tan bond o: lagat ar h rtal o 144 d arar kar ne kan ater silp prot i tangulo bond o rojeche. d: (d kt r), a: (abdul)

Number/phone number: ৯৫৬৭৪৪৩ [9567447]$_{IPA}$. In this case we can’t exactly tell whether this is phone number or number.

Step 3: Token expander: After identification of all NSW we can convert these to standard word by pronunciation lexicon or (letter to sound) LTS rule.

3.2 Text analysis

[1] Bangla Script has some other characters that are not included here; also they are not phones but modify sound.

The second step of TTS system is to convert the text to its pronunciation form. For example we write ক্ষমা [ক+্+ষ+ম+া] [k+virama+s+m+a], but we pronounce it খমা [k^h ma]. For finding pronunciation of a word we need large list of lexicon and LTS rule. We used lexicon dictionary that contain 900 lexicons with its pronunciation.

Steps of Phonetic Analysis within festival:

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

3.2.1 Building large amount of lexicon by hand

We included 900 lexicons by scheme programming. Developing LTS rule for a language is too much difficult and much more computation is needed in run time of TTS. So lexicon dictionary is important in TTS system. We implemented our pronunciation lexicon by scheme within festival.

The basic assumption in festival is that we have a large set of lexicon that is a used as a standard part of an implementation of a voice. A pronunciation in festival requires not just a list of phones but also a syllabic structure. The lexicon structure that is basically available in festival takes word, part of speech (and arbitrary token) and stress marker to find the given pronunciation of a given word. We implemented our large set of lexicon based on Bangla syllabic structure. The syllable structure [9] of Bangla Language is V, VC, VV, CV, CVC, CVV, CCV, CCVC. An example lexicon format in festival is ("aapni" n (((aa p) 0) ((n i) 0))) → আপনি $[apni]_{IPA}$

3.2.2 Building letter-to-sound rules

Bangla language always borrows words from other languages like computer (কম্পিউটার-k mputar), competition (কম্পিটিশন - k mpiti n). To find the pronunciation of new arrival words that is not found in the lexicon we have to use LTS rule. In festival there is a letter to sound rule or Grapheme to Phoneme (G2P) system that allows rules to be written, but festival also provided a method for building rule sets automatically, which will often be more useful. An explicit lexicon isn't necessary in festival and it may be possible to do much of the work in letter-to-sound rules. But in this case we have to identify proper LTS rule and we have to consider performance issue because it may take lots of computation. We used some of the LTS rule in our implementation based on our syllabification rule.

3.3 Speech Database / Waveform Synthesis

This is one of the major parts in TTS. The general-purpose concatenative synthesis [10][11] translates incoming text onto phoneme labels, stress and emphasis tags, and phrase break tags. This information is used to compute a target prosodic pattern (i.e., phoneme durations and pitch contour). Finally, signal processing methods retrieve acoustic units (fragments of speech corresponding to short phoneme sequences such as diphones) from a stored inventory, modify the units so that they match the target prosody, and smooth (concatenate) them together to form an output utterance.

Concatenative synthesis techniques give the most natural sound in speech synthesis. Three techniques are available in concatenative synthesis: diphone, unit selection and multisyn-unit selection. Diphone based speech synthesis systems can produce very intelligible synthetic speech, but less natural than unit selection technique. Unit selection [12] database can be created by automatically clustering units of the same phone class based on their phonetic features and prosodic context. The appropriate cluster is then selected for a target unit offering a small set of candidate units. We used unit selection and multisyn unit selection technique [13] for waveform synthesis. To implement speech database using festival at first we have to identify all the features of the phonemes and total number of phones. It can be done by articulatory technique or acoustic technique. Acoustic technique is the best way to identify all the phoneme of a language. We identified 45

phones excluding 31 diphthongs with their features [14] based on articulatory analysis. To build diphone database we have to include diphthong as well. In our implementation we excluded the diphthongs.

As we explained earlier we added lexicon for pronunciation of words. Also duration of the each phone is added to implement the TTS for Bangla. The duration we added is taken from Kiswahili [15] TTS system. This is not exact duration for the phone set of Bangla language. Using acoustic analysis procedure we can measure exact duration of the phone set.

4. Results

The drawback of unit selection and multisyn unit selection is that a large set of speech corpus is required to develop speech database. Approximately 500-900 recorded utterance is better to cover most frequent words of language. In our implementation we recorded sentences and trained the system in both techniques. When train the system internally festival break its unit by diphone. Diphone is the combination of two phones that is at the middle of one phone to the middle of next phone. Festival breaks the signal at zero crossing position as shown in figure 2. When the system synthesizes the voice its try to match this position that's why there is lack of signal distortion and the produced sound is quite natural.

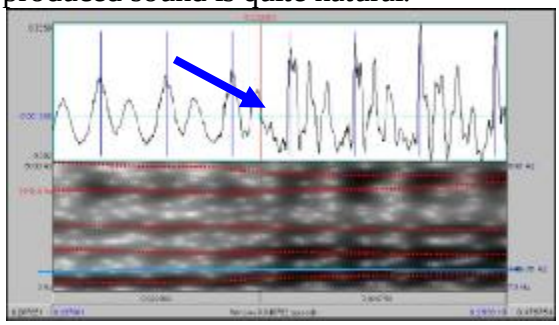

Figure 2: Splitting at the zero crossing position

Here we have shown the results based on the limited domain technique. The performance we gained from multisyn technique is 30% poor than limited domain technique. The adequacy of the system was tested in two ways: in terms of acceptability/naturalness and in terms of intelligibility. The synthesis output was directed to AC97 audio card. The experiment was performed in laboratory conditions with three participants. In our first experiment, intelligibility of synthesized speech was evaluated on three levels: sentence level, word level and phrase level based on the trained corpus. Each participant was asked to write down everything they heard. Figure 3 gives the percentage of correctly understood sentences, words and phrase. In case of sentences level the intelligibility rate being close to 85%. On phrase level it is 83.33% and word level it is 56.66%.

In our second experiment, degree of naturalness of the synthesized speech was assessed, again on sentence 90%, phrase 85% and word level 65%. The results obtained are shown in Figure 4. Despite a rather good naturalness of synthetic speech, utterances sometimes suffer a lack from intelligibility.

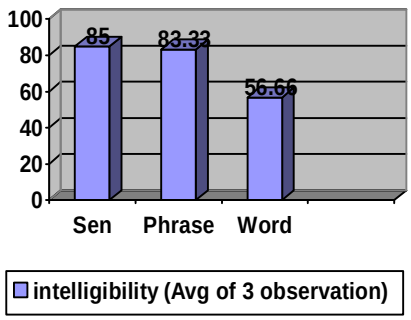

Figure 3: Intelligibility of pronunciation.

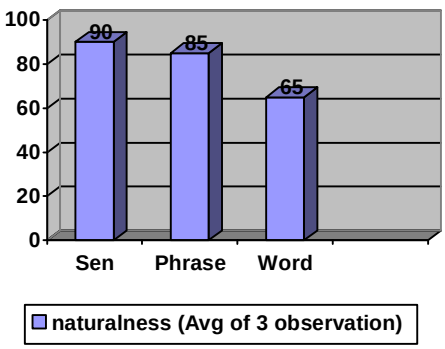

Figure 4: Naturalness of pronunciation.

5. Future Implementation

A number of plans are made to develop the complete TTS system for Bangla language including the following: Document Analysis (to analyze file type, file format, encoding, etc), Text Analysis (text analysis/normalization using scheme or java or C++ for larger context), Phonetic Analysis (proper acoustic analysis on Bangla phone set, developing large number pronunciation lexicon, automatic lexicon entries instead of adding manually, find out LTS or Grapheme-to-Phoneme (G2P) rule so that it can handle unknown words), Prosody Analysis, and Waveform synthesis by diphone technique.

6. Conclusion

The described speech synthesis system is the open source and freely distributable TTS system for Bangla language. This is the complete process to develop commarcial TTS system which includes most of the complexity of Bangla language. Besides the obvious uses of a TTS system, from listening to computerized books to ones email, it also allows the visually impaired and those who cannot read Bangla access to Bangla electronic content such as the World Wide Web. We have described a proof-of-principle implementation of a Bangla TTS, and there is much work to be done before we have a complete and commercial quality TTS system such as those available for many other languages. We have a plan to continue developing the Bangla festival voice to improve the quality of the synthesized speech. The synthetic speech produced by the system is intelligible, but lacks of naturalness. Improvement of intelligibility and naturalness depend on significant amount of work in each phase.

References:

[1] Black A., Taylor P., "The Festival Speech Synthesis System", Technical Report HCRC/TR-83, University of Edinburgh, Scotland, (1997), http://www.cstr.ed.ac.uk/projects/festival.html.

[2] Tanuja Sarkar, Venkatesh Keri,. Santhosh M and Kishore Prahallad, "Building Bengali Voice Using Festvox", ICLSI 2005

[3] Asok Bandyopadhyay, "Some Important Aspects of Bengali Speech Synthesis System" IEMCT Pune, June 24-25 2002.

[4] Shyamal Kr. DasMandal, Barnali Pal "Bengali text to speech synthesis system a novel approach for crossing literacy barrier" . CSI-YITPA(E)2002

[5] Aniruddha Sen, "Bangla Pronunciation Rules and a Text-to-Speech System", Symposium on Indian Morphology, Phonology & Language Engineering, 2004, pp. 39.

[6] K. Panchapagesan, Partha Pratim Talukdar, N. Sridhar Krishna, Kalika Bali, A. G. Ramakrishnan, "Hindi Text Normalization", Fifth International Conference on Knowledge Based Computer Systems (KBCS), 2004, Hyderabad India.

[7] Sproat, R., Black, A., Chen, S., Kumar, S., Ostendorf, M., and Richards, C. "Normalization of Non-standard Words", Computer Speech and Language, 2001, vol. 15, pp. 287-333. http://www.clsp.jhu.edu/ws99/projects/normal/slides/intro/nswintro.pdf

[8] Bengali script – Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Bengali_script

[9] MD. Abdul Hay, Dhani Biggan pg-152

[10] Zervas P., Potamitis I., Fakotakis N., Kokkinakis G. "A Greek TTS Based on Non Uniform Unit Concatenation and the Utilization of Festival Architecture", First Balkan Conference on Informatics, Thessalonica, Greece, 2003, pp. 662-668.

[11] Conkie and Alistair, "Robust Unit Selection System For Speech Synthesis", The Journal of the Acoustical Society of America, Volume 105, Issue 2, February 1999, p.978.
[12] R. Clark and K. Richmond and S. King, "Festival 2 – Build Your Own General Purpose Unit Selection Speech Synthesizer", 5th ISCA Workshop on Speech Synthesis, 2004, pp. 173
[13] Rob Clark, Multisyn Unit selection technique, http://www.cstr.ed.ac.uk/downloads/festival/multisyn_build, Unit selection technique, www.festvox.org.
[14] (1) Naira Khan (CRBLP), (2) Daniul Haque, Basa Bigganer Kotha, (3) Abdul Hai, Dhani Biggan O Dhanitotto, "Phoneme set and their features"
[15] Kiswahili TTS system, www.llsti.org.