Bangla Text to Speech using Festival

Conference Paper · May 2011

CITATIONS
11

READS
1,728

3 authors:

Firoj Alam
Qatar Computing Research Institute
60 PUBLICATIONS **485** CITATIONS
SEE PROFILE

Murtoza Habib
9 PUBLICATIONS **108** CITATIONS
SEE PROFILE

Mumit Khan
BRAC University
119 PUBLICATIONS **1,097** CITATIONS
SEE PROFILE

Bangla Text to Speech using Festival

Firoj Alam
Center for Research on Bangla Language Processing
BRAC University.
firojalam@gmail.com

S.M. Murtoza Habib
Center for Research on Bangla Language Processing
BRAC University.
murtoza@gmail.com

Mumit Khan
Center for Research on Bangla Language Processing and Department of Computer Science and Engineering
BRAC University.
mumit@bracu.ac.bd

Abstract

This paper describes the development of the first, usable, open source and freely available Bangla Text to Speech (TTS) system for Bangladeshi Bangla using the open source Festival TTS engine. Besides that, this paper also discusses a few practical applications that use this system. This system is developed using diphone concatenation approach in its waveform generation phase. Construction of a diphone database and implementation of the natural language processing modules are described. Natural language processing modules include text processing, tokenizing and grapheme to phoneme (G2P) conversion that were written in Festival's format. Finally, a test was conducted to evaluate the intelligibility of the synthesized speech.

Index Terms: speech synthesis, diphone

1 Introduction

Bangla (exonym: Bengali) is one of the most widely spoken languages of the world (it is ranked between four[1] and seven[2] based on the number of speakers), with nearly 200 million native speakers. However, this is one of the most under-resourced languages which lack speech applications. The aim of this project is to develop a freely available Bangla text to speech system. A freely available and open-source TTS system for Bangla language can greatly aid 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. A touch screen based kiosk that integrates a Bangla TTS has the potential to empower the 49% of the population who are illiterate[3]. A screen reader that integrates a Bangla TTS will do the same for the estimated 100 thousand visually impaired citizens of Bangladesh. A Text to Speech is a computer based system capable of converting computer readable text into speech. There are two main components such as Natural Language Processing (NLP) and Digital Signal Processing (DSP) [Thierry 1997][Paul 2009][A.W. Black et el. 2003]. The NLP component includes pre-processing, sentence splitting, tokenization, text analysis, homograph resolution, parsing, pronunciation, stress, syllabification and prosody prediction. Working with pronunciation, stress, syllabification and prosody prediction sometime is termed as linguistic analysis. Whereas, the DSP component includes segment list generation, speech decoding, prosody matching, segment concatenation and signal synthesis. Pre-processing is the process of identifying the text genre, character encoding issues and multilingual issues. Sentence splitting is the process of segmenting the document text into a list of sentences. Segmenting each sentence into a list of possible tokens can be done by tokenization. In text analysis part, different semiotic classes were identified, and then using a parser each token is assigned to a specific semiotic class. After that, verbalization is performed on non-natural language token. Homograph resolution is the process of identifying the correct underlying word for ambiguous token. The process of generating pronunciation from orthographic representation can be done by pronunciation lexicon and grapheme-to-phoneme (G2P) algorithm. Prosody prediction is the process of identifying the phrase break, prominence and intonation

[1] http://www2.ignatius.edu/faculty/turner/languages.htm, Last accessed December 26, 2007.

[2] http://en.wikipedia.org/wiki/List_of_languages_by_total_speakers, Last accessed December 26, 2007.

[3] Bangladesh Bureau of Statistics, 2004. http://www.bbs.gov.bd/dataindex/stat_bangladesh.pdf

tune. There has not been much work done on prosody in this paper. DSP component or waveform generation is the final stage of a TTS. This involves the production of acoustic signals using a particular synthesis approaches such as formant synthesis, articulatory synthesis and concatenation based synthesis. The attempt that has been made here is the second generation diphone concatenation based synthesis, using widely usable Festival framework [A.W. Black et al. 2003].

2 Literature survey

Significant effort has been made for different languages to develop TTS using the Festival framework such as English, Japanese [A.W. Black et al. 2003], Welsh [R.J. Jones et al. 2006], Telugu [C. Kamisetty et al. 2006], [S.P. Kishore et al. 2002], Hindi [S.P. Kishore et al. 2002], [A.G. Ramakishnan et al. 2004], Turkish [Ö. Salor et al. 2003] and Sinhala [Ruvan et al. 2007]. However, very little work has been done on Bangla. Several attempts have been made in the past, where different aspects of a Bangla TTS system were covered in [Tanuja et al. 2005], [Asok 2002], [Shyamal et al. 2002] and [Aniruddha et al. 2004]. [Tanuja et al. 2005] showed different TTS modules such as optimal text selection, G2P conversion and automatic segmentation with experimental results. Phoneme and partname were used to develop voice database and ESNOLA technique were used for concatenation [Asok 2002], but the quality of the system suffers due to the lack of smoothness. Shyamal et al. [Shyamal et al. 2002] showed some practical applications with Bangla TTS system using ESNOLA technique. In [Aniruddha et al. 2004] author showed the pronunciation rule and phoneme to speech synthesizer using formant synthesis technique. Another attempt has been made to develop Bangla TTS using multisyn unit selection and unit selection technique within Festival framework in [Firoj et al. 2007] but the system was developed on a limited domain and could not be used as a general purpose TTS system. To the best of our knowledge this is the first complete work for general purpose, open source, freely available and platform independent Bangla Text to Speech system for Bangladeshi Bangla.

3 Development

3.1 Bangla writing system

Bangla is written left to right in horizontal lines with a left-to-right heatstroke (is called matra). The presence and absence of heatstroke has significant implications to distinguish consonant conjunct, dependent and independent vowel. Words are delimited by a space in general. Vowels have corresponding full-character forms when they appear in an absolute initial position of a word. Generally a vowel followed by a consonant takes a modified shape and placed at the left, right or both, or at the bottom of the consonant which are signifies as vowel modifiers. The inventory of Bengali script is made up of 11 vowels, 39 consonants, 216 consonant conjuncts, 10 digits, modifiers, punctuation marks and a few symbols [Bangal Academy 1992][Wikipedia 2010]. The vowel and consonant characters are called basic characters. The consonant conjunct is joined by ‘hasanta’. The concept of upper and lower case is absent in Bangla script. English equivalent full stop is the Bengali punctuation mark the down-stroke dari (।); Unicode - \u0964. Commas, semicolons, colons, quotation marks, etc. are the same as in English.

3.2 Bangla phoneme inventory

The phoneme inventory of Bangla consists of 30 consonants, 14 monophthong vowels (oral and nasal vowels) and 21 diphthongs [Firoj et al. 2008 (b)] [Firoj et al. 2008 (a)]. Consonants and vowels are shown in Table 1 and Table 2. The diphthongs are the following: অও/ɔo/, আইai/, আউ /au/, আয়া/aja/, ইউ/iu/, ইএ-ইয়ে/ie/, ইও/io/, ইয়া-ইআ /ia/, উই/ui/, উয়া-উআ/ua/, উয়ে/ue/, উয়ো-উও/uo/, এই/ei/, এউ/eu/, এও/eo/, এয়া-এআ/ea/, এ্যায়া/æa/, ওই/oi/, ওউ/ou/, ওয়া-ওআ /oa/, ওয়ে/oe/.

	Front	Central	Back
High	ই/i, ঈ/ĩ,		উ/u, ঊ/ũ/
High-Mid	এ/e, এঁ/ẽ		ও/o, ওঁ/õ
Mid-Low	এ্যা/æ,এ্যাঁ/æ̃		অ/ɔ/, অঁ/ɔ̃/
Low		আ/a, আঁ/ã	

Table 1: Vowel phoneme inventory

3.3 Natural language processing in Festival

Festvox [A.W. Black et al. 2003] provides different natural language processing modules for building a new voice. These modules can be generated automatically which appears as a form of scheme files. The scheme files need to be customized for a new language. The language spe-

cific scripts (phone, lexicon and tokenization) and speaker specific scripts (duration and intonation) can be externally configured and implemented without recompiling the system [A.W. Black et al. 2003]. Since the templates are scheme files, which is typically an interpreted language, so recompilation is not required. The following NLP related tasks are involved when building a new voice in Festvox:

• Defining the phoneset
• Tokenization and text normalization
• Pronunciation: Lexicon and grapheme to phoneme conversion.
• Implementation of syllabification and stress
• Prosody: Phrase breaking, accent prediction, assignment of duration to phones and generation of f0 contour.

whitespace and the punctuation marks, which is used in our implementation to tokenize Bangla text. After tokenization, text normalization is performed. In text normalization, the first task is to identify the semiotic classes. The following section discusses the semiotic class [Paul 2009] (as opposed to say NSW) identification, tokenization and standard word generation and disambiguation rule. Moreover, this work has been done separately before implementing into Festival.

We identified a set of semiotic classes which belong to the Bangla language. To do this, we have selected a news corpus [Prothom-Alo 2009] [Khair et al. 2006] with 18100378 tokens and 384048 token types [Khair et al. 2006], forum [forum.amaderprojukti.com 2008] and blog [www.somewhereinblog.net 2008]. After that we

		Bilabial		**Dental**		**Alveolar**		**Post- Alveolar**		**Palatal**	**Velar**		**Glottal**
Stops	voiceless	প /p/	ফ /p^h/	ত /t̪/	থ /$t̪^h$/	ট /ʈ/	ঠ /$ʈ^h$/	চ /c/	ছ /c^h/		ক /k/	খ /k^h/	
	voiced	ব /b/	ভ /b^h/	দ /d̪/	ধ /$d̪^h$/	ড /ɖ/	ঢ /$ɖ^h$/	য, জ/ɟ/	ঝ /$ɟ^h$/		গ /g/	ঘ /g^h/	
Nasals		ম/m/				ন,ণ /n/					ঙ,ং /ŋ/		
Trill						র /r/							
Flap						ড়, ঢ় /ɾ/						
Fricatives						শ,স /s/				শ,ষ,স /ʃ/			হ,ঃ /h/
Lateral						ল /l/							
Approximant										য় /j/			

Table 2: Consonant phoneme inventory

3.3.1 Defining phoneset for Bangla

The phoneset that has been explained in section 3.2 was implemented in festival format which had to be transcribed into ASCII. Festvox has a separate module to implement this phoneset with their features. For vowel, the features include height, length (e.g: short, long, diphthong and schwa), front vs back, lip rounding, and tense vs lex. For consonant, the features include place of articulation (e.g: bilabial, dental, alveolar, post-alveolar, palatal, velar and glottal), manner of articulation (e.g: stop, nasal, tril, flap, fricative, lateral, glide/approximant), aspiration and voicing.

3.3.2 Text analysis and text normalization

Like any conventional writing system, Bangla script uses whitespace as a delimiter which helped us to make a one-to-one mapping between tokens and words. Besides whitespace, Bangla script also uses punctuation (such as dari-।, ?, !, ;) as a delimiter. The default text tokenization methodology available in Festival is the proceeded in two steps to identify the semiotic classes. Firstly, a python [Python 2008] script was used to identify the semiotic class from news corpus and manually checked the semiotic classes in the corpus of forum and blog. Secondly, we defined a set of rules according to context of homographs or ambiguous tokens to find the semiotic classes. The resulted set of semiotic classes of Bangla text is shown in Table 3.

Semiotic class/token type	**Example**
English text	জাভা Platform Independent বলে
Bangla text	আমি বাংলায় কথা বলি
Numbers (cardinal, ordinal, roman, floating number, fraction, ratio)	১২১,২৩,২৩৪; ১ম, ২য়, ৩য়; I, II, III, ১২.২৩, ২৩,৩৩.৩৩; ১/২, ২৩/২৩; ১২:১২
Telephone and mobile number	০২৯৫৬৭৪৪৭; ০১৫২৩০৩৩৯৮ (19 different formats)
Years	২০০৬; ১৯৯৮; ৯৮ সালে
Date	০২-০৬-২০০৬ (12 different formats)

Time	৪.২০ মিঃ; ৪.২০ মিনিট;
Percentage	১২%
Money	১০ ৳
E-mail	আমার ই-মেইল ঠিকানা: abc@yahoo.com
URL	সফটওয়্যারটি সাইট http://googlecode.com
Abbreviation	ডঃ; মোঃ; সাঃ
Acronym	ঢাবি; বাউবি, কেবি
Mathematical equation	(১+২=৩)

Table 3: Possible token type in Bangla text

A set of tags defined for each semiotic class and assigned these tags to each class of tokens. The tokenization undergoes three levels such as: i. Tokenizer ii. Splitter and iii. Classifier. Whitespace is used to tokenize a string of characters into a separate token. Punctuations and delimiters were identified and used by the splitter to classify the token. Context sensitive rules were written as whitespace is not a valid delimiter for tokenizing phone numbers, year, time and floating point numbers. Finally, the classifier classifies the token by looking at the contextual rule. For each type of token, regular expression were written in festival scheme.

The token expander expands the token by verbalizing and disambiguating the ambiguous token. Verbalization [Paul 2009] or standard word generation is the process of converting non-natural language text into standard words or natural language text. A template based approach [Paul 2009] such as the lexicon was used for number cardinal, ordinal, acronym, and abbreviations. Abbreviations are productive and a new one may appear, so an automatic process may require solving unknown abbreviations. In case of Bangla acronyms, most of the time people say the acronym as a full form without expanding it. For example, দুদক /d̪ud̪ɔk/ expands to দুর্নীতি দমন কমিশন /d̪urniti d̪ɔmon komiʃɔn/ but people say it as দুদক /d̪ud̪ɔk/. Bangla has the same type of non-natural language ambiguity like Hindi [K. Panchapagesan et al. 2004] in the token year-number and time-floating number. For example: (i). the token ১৯৯৮ (1998) could be considered as a year and at the same time it could be considered as number and (ii). the token ১২. ৮০ (12.80) could be considered as a floating point number and it could be considered as a time. Context dependent hand written rules were applied for these ambiguities. In case of Bangla, after time pattern ১২. ৩০ (12.30) we have a token মিঃ (minute), so we look at the next token and decide whether it is time or a floating point number. There are rare cases where context dependent rules fail in year-number ambiguity then we verbalize the token as a pair of two digits. For example, the token ১৯৯৮ (1998), we expand it as উনিশ শত আটানব্বই (Nineteen hundred ninety eight) rather than এক হাজার নয় শত আটানব্বই (one thousand nine hundred ninety eight). The natural language text is relatively straightforward, and Bangla does not have upper and lower case. The system implemented based on the work of [Firoj et al. 2009], claims that the accuracy of the ambiguous token is 87%.

3.3.3 Pronunciation

This system takes the word based on orthographic linguistic representation and generates a phonemic or phonetic description of what is to be spoken by the subsequent phases of TTS. In generating this representation we used a lexicon of known words and a grapheme-to-phoneme (G2P) algorithm to handle proper names and unknown words.

We developed a system lexicon [2, pp215] where the entries contain orthography and pronunciation in IPA. Due to the lack of a digitized offline lexicon for Bangla we had to develop it manually by linguistic experts. To the best of our knowledge this is the first digitized IPA incorporated and syllabified lexicon. The lexicon contains 93K entries where 80K entries entered by hand and the rest of them were automatically generated by G2P system [Ayesha et al. 2006]. The performance of this G2P system is 89.48%. Therefore, the automatically generated entries had to be checked manually to maintain the quality of the lexicon by expert linguists. The system is now available in online for public access [CRBLP 2010]. Another case needs to be handled in order to implement the lexicon into Festival. The Unicode encoded phonetic representation needs to be converted into ASCII to incorporate into festival.

We have implemented the G2P algorithm that is proposed by Ayesha et al. [Ayesha et al. 2006] to handle unknown words and proper name. In Festival, the UTF-8 textual input was converted into ASCII based phonetic representation in a Festival's context sensitive rule [A.W. Black et al. 2003]. The rules were re-written in UTF-8

multi-byte format following the work done for Telugu [C. Kamisetta et al. 2006] and Sinhala [Ruvan et al. 2007]. The method was proven to work well with promising speed. The rules proposed in [Ayesha et al. 2006] were expanded up to 3880 rules when re-written in Festival context sensitive format.

Another attempt has been made to reduce the size of the lexicon for TTS. The lossless compression [Paul 2009] technique was applied to reduce the size of the lexicon. Lossless compression technique is a technique where the output is exactly the same as when the full lexicon is used. It is just the generalities of the lexicon that has been exactly captured in a set of rules. This technique reduces the size of our lexicon to ~50K entries from 93K.

3.3.4 Syllabification and stress

Festival's default syllabification algorithm based on sonority sequencing principle [A.W. Black et al. 2003] is used to syllabify the Bangla words. Besides the default syllabification algorithm, our lexicon has also been syllabified along with pronunciation.

Little work has been done on Bangla stress. Identifying the stress pattern for Bangla is beyond the scope of this paper. Considering Bangla as a stress less language we have used Festival's default stress algorithm. In our implementation of lexicon we have not incorporated the stress marker.

3.3.5. Prosody Implementation

Prosody is one of the important factors contributing to natural sounding speech. This includes phrasing, accent/boundary prediction, duration assignment to phones and f0 generation. The presence of phrase breaks in the proper positions of an utterance affects the meaning, naturalness and intelligibility of the speech. Festival supports two methods for predicting phrase breaks. The first one is to define a Classification and Regression Tree (CART). The second and more elaborate method of phrase break prediction is to implement a probabilistic model using probabilities of a break after a word, based on the part of speech of the neighboring words and the previous word [A.W. Black et al. 2003]. However, due to the lack of a POS tagger for Bangla, we have not able to construct a probabilistic model yet. Therefore, we decided to use a simple CART based phrase breaking algorithm described in [A.W. Black et al. 2003]. The algorithm is based on the assumption that phrase boundaries are more likely between content words and function words. A rule is defined to predict a break if the current word is a content word and the next is seemingly a function word and the current word is more than 5 words from a punctuation symbol. Since function words are limited in a language so we specified them as function words and considered rest of them as content words. The function words that we used here to implement the phrase break model is shown in Table 4. These function words need to be converted into ASCII form to incorporate into festival phrase breaking algorithm.

Function words
অ, অতএব, অথচ, অথবা, অধিকন্তু, অপেক্ষা, অর্থাৎ, আর, আরও, এ, এই, এবং, ও, কিংবা, কিন্তু, তথা, তথাপি, তবু, তবুও, তাই, তো, নইলে, নইলে, নতুবা, নয়তো, না- হয়, বটে, বরং, বরঞ্চ, বস্তুত, বা, যথা, যদি, যদিও, যে, যেন, যেহেতু, সুতরাং, হঠাৎ, হয়তো

Table 4: Function words

To predict accent and boundary tone, Festival uses simple rules to produce sophisticated system. To make a more sophisticated system such as a statistical model one needs to have an appropriate set of data. Due to the lack of the availability of this data we used a simple accent prediction approach [A.W. Black et al. 2003] which proved surprisingly well for English. This approach assigns an accent on lexically stressed syllable in all content words.

Festival uses different approach for F0 generation such as F0 by rule, CART tree and tilt modeling. In our implementation we used rule based approach. An attempt has been made to make a CART tree based model from the data; however, surprisingly that has not been work well.

Several duration models support by Festival such as fixed models, simple rules models, complex rules models and trained models. We used fixed duration model that was implemented from the work done by Firoj et al. [Firoj et al. 2008 (b)][Firoj et al. 2008 (a)].

3.4 Development of diphone database

Developing a speech database is always time consuming and laborious. The basic idea of building a diphone database is to explicitly list all phone-phone combination of a language. It is mentioned in section 3.2 that Bangla language has 30 consonants and 35 vowels (monophthong,

diphthong) phonemes. In general, the number of diphone in a language is the square of the number of phones. Since Bangla language consists of 65 phones, so the number of diphones are (65X65) 4225. In addition, silence to phones are (1X65) 65, phones to silence are (65X1) 65 and a silence. So the total number of diphones is 4336. In the first step, a list has been made to maintain all the possible vowel consonant combination with the following pattern: VC, CV, VV, CC, SIL_V, SIL_C, V_SIL, C_SIL and SIL. Here SIL is silence, V is vowel and C is consonant. Silence is considered as a phoneme, usually taken at the beginning and ending of the phonemes to match the silences occurring before, between and after the words. They are therefore an important unit within the diphone inventory. These diphones were embedded with carrier sentences using an external program. The diphone is inserted in the middle of the word of a sentence, minimizing the articulatory effects at the start and end of the word. Also, the use of nonsense words helped the speaker to maintain a neutral prosodic context. Though there have been various techniques to embed diphone with carrier sentences, here nonsense words were used to form carrier sentences [A.W. Black et al. 2003].
In this list, there could be redundant diphones those need to be marked and omitted. The study of phonotactics says that all phone-phone pair cannot be exist in a language. Due to the lack of existing work and linguistic experts we were not able to work on this phenomenon. Therefore, the whole diphone list was selected for recording.

Since speaker choice is perhaps one of the most vital areas for recording so a careful measure had taken. Two potential speakers was chosen and their recording were played to a listening group and asked them which they prefer. According to the measurement of the listening group a male speaker was chosen who is a professional speaker and aged 29.
As far as recording conditions is concerned, we tried to maintain as high quality as possible. The speech data was digitized at a sample rate 44.1 kHz, sample width 24-bit resolution and stored as wave format. After each recording, the moderator checked for any misleading pronunciation during the recording, and if so, the affected utterances were re-recorded.

There were a few challenges in the recording. First, speaker was asked to keep the speaking style consistent. Second, speaker was supervised to keep the same tone in the recording.

The most laborious and painstaking task is to clean the recording and then hand-labeled the diphone using the speech analysis software tool 'Praat' [4]. During labeling, at first we labeled phone boundary, then automatically marked the diphone boundary using Praat script. Another important factor is that, every boundary should be placed in zero crossing. Failing to do so produces audible distortions, this in turns generates clicks. Afterwards, a script was written to transform Praat textgrid files into diphone index file (.est) [A.W. Black et al. 2003] as required by Festival.
Festival, in its publicly distributed form only supports residual excited Linear Predictive Coding (LPC). This method requires pitch marks, LPC parameters and LPC residual values for each diphone in the diphone database. The script *make_pm_wave* provided by speech tools [A.W. Black et al. 2003] was used to extract pitch marks from the wave files. Then, the *make_lpc* command was invoked in order to compute LPC coefficients and residuals from the wave files [A.W. Black et al. 2003]. To maintain an equal power we used proprietary software tool to normalize it in terms of power so that all diphones had an approximately equivalent power. After that the diphone database was grouped in order to make it accessible by Festival's UniSyn synthesizer module, and to make it ready for distribution.

4 Integration with applications

The Bangla Text to Speech runs on Linux, Windows and Mac OSX. There is also a web-enabled front-end for the TTS, making this tool available at anytime and from anywhere.

Since Festival is incapable of reading UTF-8 text files with byte-order marker (BOM) so manual BOM removal patch was used which was written by Weerasinghe et al. [Ruvan et al. 2007]. This patch was incorporated with Festival text processing module.

To develop windows version we had motivated by the work carried out in the Welsh and Irish Speech Processing Resources (WISPR) project [B. Williams et al. 2006]. Following the work of WISPR, we implemented TTS using Microsoft Speech Application Programming Interface (MS-SAPI) which provides the standard speech synthesis and speech recognition interface within Windows applications [Microsoft

[4] Available from: http://www.praat.org

1999]. Consequently, the MS-SAPI compliant Bangla voice is accessible via any speech enabled Windows application. The system has been tested with NVDA[5] and Dolphin[6] screen reader. Moreover, it is also tested with WordTalk[7], a free text-to-speech plug-in for Microsoft Word which runs as a macro. Currently Bengali speaking print disabled community accessing local language content using Bangla Text to speech system via screen reader.

Besides, there are few other applications that currently testing this system such as talking dictionary, DAISY[8] book, agro-information system and news reader. Using this system one of the newspapers in Bangladesh developed their audio version of newspaper to make mp3 of their daily content.

5 Evaluation

Any real system needs to undergo rigorous testing before deployment. Though TTS testing is not a simple or widely agreed area, it is widely agreed that a TTS system has two main goals on system test; that is a synthesized speech should be i) intelligible and ii) natural. Intelligibility test can be performed by word recognition tests or comprehension tests where listeners are played a few words either in isolation or in a sentence and asked which word(s) they heard. In naturalness test, listeners are played some speech (phrase or sentence) and simply asked to rate what they hear. This can be done by mean opinion score. Since these testing may not always be the best approach so people also use unit testing approach.

As our goal was to make a general-purpose synthesizer, a decision was made to evaluate it under the intelligibility criterion and unit testing on a few components. The most commonly used word recognition test - modified rhyme test (MRT) [Paul 2009] was designed to test Bangla TTS system. Based on the MRT we designed a set of 77 groups - 5 words each. Therefore a set of 385 words came into testing. The words in each group are similar and differ in only one consonant and the users were asked to account which word they have heard on a multiple choice sheet. Based on the test the overall intelligibility of the system from 6 listeners is 96.96%. Besides the intelligibility test, we have performed a unit test on text normalizer and G2P converter. The performance of text normalizer is 87% only for ambiguous tokens and that of G2P converter is 89%.

6 Conclusions

Here the development of the first-ever complete Text to Speech (TTS) system has described, that can convert a Unicode encoded Bangla text into human speech. It is distributed under an open source license to empower both the user and developer communities. This TTS system can also be used with any available Screen Reader. In addition to the standalone TTS client, it can be integrated into virtually any application, and can also be accessed as a Web Service. Incorporating this technology in various applications such as screen reader for the visually impaired, touch screen based agro-information system, talking books, telecenter applications, e-content, etc., can potentially bridge the literacy divide in Bangladesh, which in turn goes towards bridging the digital divide. An evaluation of the system has been done based on MRT and unit testing on a few components to check intelligibility.

Since the voice developed here is diphone concatenation based and it lacks proper intonation modeling so it produces robotic speech. Therefore, a natural sounding voice needs to be made in future, which could be performed by developing a unit selection voice. Besides that, a few works need to be done in future to improve the intelligibility of the system such as POS tagger, improvement of G2P algorithm, improvement of text normalizer and working on intonation modeling.

Acknowledgments

This work has been supported in part by the PAN Localization Project (www.panl10n.net), grant from the International Development Research Center (IDRC), Ottawa, Canada. We would also like to thank Dr Sarmad Hussain (NUCES), and Naira Khan (Dhaka University).

References

A.G. Ramakishnan, K. Bali, P.P. Talukdar and N.S. Krishna, 2004, Tools for the Development of a HindiSpeech Synthesis System, In 5th ISCA Speech Synthesis Workshop, Pittsburgh, 2004, pp. 109-114.

A.W. Black, and K.A. Lenzo, 2003, Building Synthetic Voices, Language Technologies Institute, Carnegie Mellon University and Cepstral LLC. Retrieved from: http://festvox.org/bsv/

[5] NVDA – NonVisual Desktop Access. www.nvda-project.org/

[6] Dolphin screen reader.www.yourdolphin.com/

[7] Wordtalk - www.wordtalk.org.uk/

[8] DAISY - Digital Accessible Information System

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

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

Ayesha Binte Mosaddeque, Naushad UzZaman and Mumit Khan, 2006, Rule based Automated Pronunciation Generator, Proc. of 9th International Conference on Computer and Information Technology (ICCIT 2006), Dhaka, Bangladesh, December 2006.

B. Williams, R.J. Jones and I. Uemlianin, 2006, Tools and Resources for Speech Synthesis Arising from a Welsh TTS Project, Fifth Language Resources and Evaluation Conference (LREC), Genoa, Italy, 2006.

C. Kamisetty and S.M. Adapa, 2006, Telugu Festival Text-to-Speech System, Retrieved from: http://festivalte.sourceforge.net/wiki/Main_Page

CRBLP, 2010, CRBLP pronunciation lexicon, [Online], Available: http://crblp.bracu.ac.bd/demo/PL/

Firoj Alam, Promila Kanti Nath and Mumit Khan, 2007, Text To Speech for Bangla Language using Festival, Proc. of 1st International Conference on Digital Communications and Computer Applications (DCCA2007), Irbid, Jordan

Firoj Alam, S. M. Murtoza Habib and Mumit Khan, 2008 (a), Research Report on Acoustic Analysis of Bangla Vowel Inventory, Center for Research on Bangla Language Processing, BRAC University.

Firoj Alam, S. M. Murtoza Habib and Mumit Khan, 2009, Text Normalization System for Bangla, Conference on Language and Technology 2009 (CLT09), NUCES, Lahore, Pakistan, January 22-24.

Firoj Alam, S. M. Murtoza Habib and Professor Mumit Khan, 2008 (b), Acoustic Analysis of Bangla Consonants, Proc. Spoken Language Technologies for Under-resourced language (SLTU'08), Vietnam, May 5-7, page 108-113.

forum.amaderprojukti.com, 2008, Forum

K. Panchapagesan, Partha Pratim Talukdar, N. Sridhar Krishna, Kalika Bali, A.G. Ramakrishnan, 2004, Hindi Text Normalization, Fifth International Conference on Knowledge Based Computer Systems (KBCS), Hyderabad, India, 19-22 December 2004. Retrieved (June, 1, 2008).

Khair Md. Yeasir Arafat Majumder, Md. Zahurul Islam, Naushad UzZaman and Mumit Khan, 2006, Analysis of and Observations from a Bangla News Corpus, in proc. 9th International Conference on Computer and Information Technology, Dhaka, Bangladesh, December.

Microsoft Corporation.: Microsoft Speech SDK Version 5.1, 1999, Retrieved from: http://msdn2.microsoft.com/ens/library/ms990097. aspx. Smith, J. O. and Abel, J. S., Bark and ERB Bilinear Trans-forms", IEEE Trans. Speech and Audio Proc., 7(6):697-708.

Ö. Salor, B. Pellom and M. Demirekler, 2003, Implementation and Evaluation of a Text-to-Speech Synthesis System for Turkish, Proceedings of Eurospeech-Interspeech, Geneva, Switzerland, pp. 1573-1576.

Paul Taylor, 2009, Text-to-Speech Synthesis, University of Cambridge, February.

Prothom-Alo, 2009, www.prothom-alo.com, Daily Bengali newspaper

Python, 2008, www.python.org, version 2.5.2

R.J. Jones, A. Choy and B. Williams, 2006, Integrating Festival and Windows, InterSpeech 2006, 9th International Conference on Spoken Language Processing, Pittsburgh, USA.

Ruvan Weerasinghe, Asanka Wasala, Viraj Welgama and Kumudu Gamage, 2007, Festival-si: A Sinhala Text-to-Speech System, Proceedings of Text, Speech and Dialogue, 10th International Conference, TSD 2007, Pilsen, Czech Republic, September 3-7, Page 472-479.

S.P. Kishore, R. Sangal and M. Srinivas, 2002, Building Hindi and Telugu Voices using Festvox, Proceedings of the International Conference On Natutal Language Processing 2002 (ICON-2002), Mumbai, India.

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

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

Thierry Dutoit, 1997, An Introduction to Text-To-Speech Synthesis, Kluwer Academic Publishers

Wikipedia contributors. Bengali script [Internet]. Wikipedia, 2010, The Free Encyclopedia; 2010 Jul 22, 17:27 UTC [cited 2010 Jul 30], Available online at: http://en.wikipedia.org/w/index.php?title=Bengali_script&oldid=374884413.

www.somewhereinblog.net, 2008, Blog