# Text To Speech for Bangla Language using Festival

Firoj Alam, Promila Kanti Nath and Mumit Khan
*BRAC University, Bangladesh*
*firojalam04@yahoo.com, bappinath@hotmail.com, 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 modules of such a TTS system are described in this paper, followed by an evaluation of the quality of synthesized speech for acceptability and intelligibility.*

## 1. Introduction

The next step in the evolution of Human Computer Interaction (HCI) is the integration of speech and language technologies enabling users to carry out spoken dialogue with computers. a Text to Speech (TTS) system is the primary component required to make this happen, opening up a world of possibilities from empowering the visually handicapped users to removing the barrier to the information age by the illiterate masses in a country like Bangladesh. 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. In this paper, we describe the methodology and implementation of a TTS system for Bangla based on the Festival TTS engine. We also evaluate the TTS for Bangla and look at future possibilities.

The organization of the paper is as follows. Section 2 discusses related works. Section 3 discusses the methodology. Section 4 discusses results. Then section 5 discusses future implementation. After that in section 6 we discuss 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 segmentation tools) in detail and experiment results of the different module have shown. In [3], a significant amount of work has been 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 intelligibility and naturalness of the system have shown.

## 3. Methodology

The TTS for Bangla uses the widely used Festival TTS engine. [1] The different phases of the synthesis task are performed by several modules as shown in Figure 1. The text analysis module part converts all non standard words to standard ones. The phonemic analysis module is a grapheme-to-phoneme converter, converting the written text into a sequence of phonemic symbols. The prosodic analysis module then takes the phoneme sequence, and assigns to each phoneme the required pitch and duration. Both the

phonemic and prosodic analyses are typically language dependent. Then the final the speech synthesis is performed by using two different concatenative synthesis techniques available in the Festival engine – unit selection and multisyn unit selection. We implemented all the modules using Festival tools.

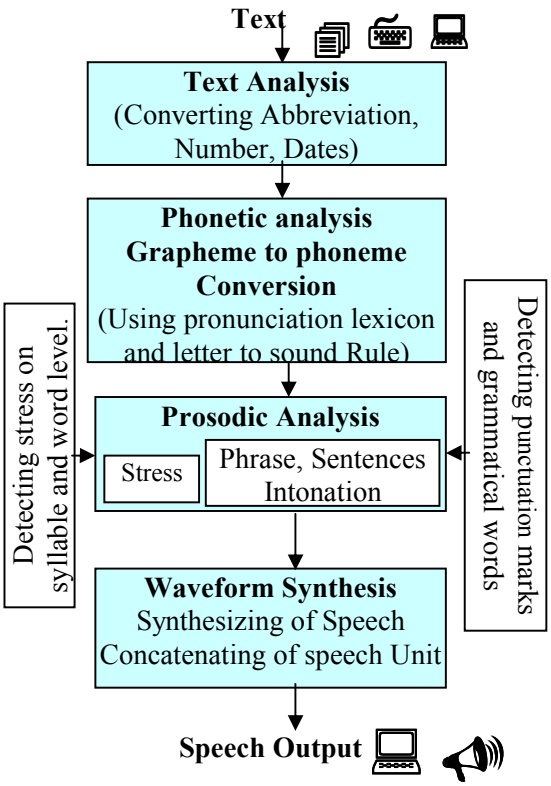


**Figure 1: Architecture of TTS**

## 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.

**Table 1: Bangla to ASCII transliteration table**[1]

| 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 |

**3.1.2. Steps of text analysis in Festival.** The steps for text analysis in Festival are as follows.

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.

**Table 2: NSW categories**

| NSW Category | Written format | Pronunciation | IPA transcription |
|---|---|---|---|
| Cardinal number | ৯৫৬৭৪৪৭ | নয় পাঁচ ছয় সাত চার চার সাত | noj p˜ac c$^{h}$oj sat̪ |

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

| | | | |
|---|---|---|---|
| | | | $c^{h}$ar sa̱t̪ |
| Ordinal number | ১ম | প্রথম | pr̪ɔ̱t̪$^{h}$ɔm |
| Date | ০২/০৬/০৬ | দুই জুন দুই হাজার ছয় | d̪ui ɟun d̪ui haɟar $c^{h}$ɔj |
| Time | ৪:২০ মি: | চারটা বিশ মিনিট | $c^{h}$arta biʃ minit |
| Ratio | ১:২ | এক অনুপাত দুই | ek ɔnupat̪ d̪ui |
| Special character | ৳ | টাকা | taka |
| Acronym | ঢাবি | ঢাকা বিশ্ববিদ্যালয় | daka biʃʃɔbid̪d̪al ɔj |
| 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] ʃikkha pro̱t̪iʃtan bond̪o: lagat̪ar hɔrtal o 144 d̪arar karɔne kanʃater silpɔ pro̱t̪iʃtangulo bond̪o rojeche. d: (dɔktɔr), 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

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 [$k^{h}$ɔma]. 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 a lexicon with 900 entries programmatically using the embedded Scheme interpreter. Developing the Letter-to-Sound (LTS ) rule for Bangla language however proved to much too difficult for this stage, so it is left for a future implementation. Also, using the LTS rules is much more computationally intensive, Festival engine prefers an elaborate lexicon instead. We implemented our pronunciation lexicon by scheme within festival.

The Festival TTS engine assumes that we have a large lexicon when building a voice. The lexicon contains not just the phonemic representation of each entry, but also the syllabic structure and other annotations such as part of speech tags, stress markers, etc. Festival uses the attributes of each entry to synthesize its pronunciation. 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-kɔmputar), competition (কম্পিটিশন - kɔmpitiʃɔn). To find the pronunciation of new arrival words that is not found in the lexicon we have to use LTS rule. The LTS, also called the Grapheme-to-phoneme (G2P), rules can be specified in Festival in two ways: by providing Festival the handcrafted rules, and by building the rules automatically. Having a proper LTS rule-set obviates the need for an explicit lexicon in Festival; however,

the difficulty in building LTS for a language like Bangla, and the associated computational complexity in interpreting these rules, make having an explicit lexicon much more convenient. In practice, both are used – the lexicon with the most frequently found annotated words, and the LTS rules for the rest. We used some of the LTS rule in our implementation based on our syllabification rule.

### 3.3. Speech Database / Waveform Synthesis

This forms the core of the TTS "backend" that is language independent by design. The first step in the concatenative synthesis is to translate text to the corresponding labeled phonemes, along with the various attributes such as stress and emphasis markers and phrase break tags. The next step takes this information and produces the target prosodic patterns, which is then processed through various steps to produce the output utterances. [10][11]

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 systems sound "wooded", i.e., unnatural, even if it's quite intelligible to native speakers. Unit selection produces more "natural" sounding speech, so it has an advantage over the diphone concatenation. Another advantage of unit selection systems is that the database can be created automatically. [12] 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.

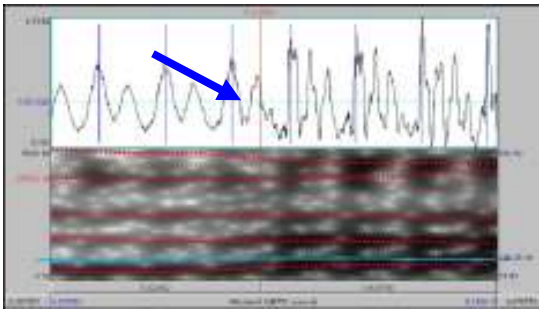

**Figure 2: Splitting at the zero crossing position**

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. There were two metrics used to evaluate the system – acceptability/naturalness and intelligibility – under laboratory conditions. 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.

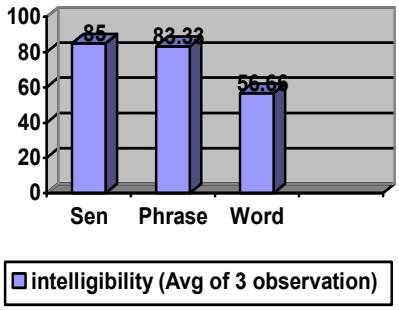


**Figure 3: Intelligibility of pronunciation**

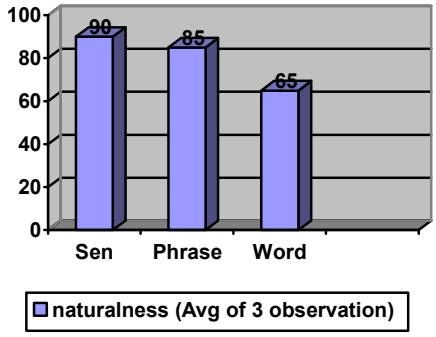


**Figure 4: Naturalness of pronunciation**

## 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.

## 7. References

[1] A. Black, P. Taylor, "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] T. Sarkar, V. Keri, M. Santhosh and K. Prahallad, "Building Bengali Voice Using Festvox", *ICLSI 2005*.

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

[4] S.K.D. Mandal and B. Pal "Bengali Text to Speech Synthesis System: A Novel Approach for Crossing Literacy Barrier", *CSI-YITPA(E)*, 2002

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

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

[7] R. Sproat, A. Black, S. Chen, S. Kumar, M. Ostendorf and C. Richards, "Normalization of Non-standard Words", *Computer Speech and Language, vol. 15*, 2001, 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] M.D.A. Hay, *Dhani Biggan*.

[10] P. Zervas, I. Potamitis, N. Fakotakis, G. Kokkinakis, "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] A. Conkie, "Robust Unit Selection System For Speech Synthesis", *The Journal of the Acoustical Society of America, Volume 105, Issue 2*, February 1999, pp. 978.

[12] R. Clark, 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] R. Clark, Multisyn Unit selection technique, http://www.cstr.ed.ac.uk/downloads/festival/multisyn_build, Unit selection technique, www.festvox.org.

[14] N. Khan, D. Haque, B.B. Kotha, A. Hai and D.B.O. Dhanitotto, "Phoneme set and their features", CRBLP.

[15] Kiswahili TTS system, www.llsti.org.