End-to-End Speech Synthesis for Bangla with Text Normalization

Tanzir Islam Pial
Department of Computer Science and Engineering
University of Dhaka
Dhaka, Bangladesh
tanzir2430@gmail.com

Shahreen Salim Aunti
Department of Computer Science and Engineering
University of Dhaka
Dhaka, Bangladesh
shahreen33.csedu@gmail.com

Shabbir Ahmed
Department of Computer Science and Engineering
University of Dhaka
Dhaka, Bangladesh
shabbir@cse.du.ac.bd

Hasnain Heickal
Department of Computer Science and Engineering
University of Dhaka
Dhaka, Bangladesh
hasnain@cse.du.ac.bd

***Abstract*—Text to speech synthesis is a well-researched area, yet no system has been developed which can claim to be as convincing as a human voice. An end-to-end system in the context of speech synthesis denotes a system capable of synthesizing speech from text using training data as minimal as transcribed audio data without any language-specific knowledge and phoneme dictionaries. But an end-to-end system should also have the capability to integrate any language-specific rules to improve its performance. In this paper, we propose an end-to-end speech synthesis system for Bangla (also known as Bengali) which uses a minimal front end and a neural network as its statistical parametric model. We also propose a Text Normalization Procedure(TNP) for Bangla and incorporate it to the end-to-end system. We have conducted extensive experiments using different models. From the feedback from the participants of the experiment, we have found out that they felt more positively towards the system if TNP is incorporated. A Wilcoxon signed-rank test was conducted to validate the results of the experiment and the probability of the results being like this because of experimental errors rather than TNP was calculated to be less than 5%.**

***Index Terms*—Deep neural network, End-to-end speech synthesis, Natural language, Text normalization, Text to speech synthesis.**

I. Introduction

Human-computer interaction via speech is one of the most pertinent sectors in artificial intelligence. Speech synthesis is a part of this interaction that involves constructing a synthetic replica of human speech. The objective is to synthesize speech in such a way that it is both intelligible and similar to human speech.

Deshai Sidhi et al. [1] and Youcef Tabet et al. [2] conducted extensive surveys on existing speech synthesis techniques in 2017 and 2011 respectively. Text to speech synthesis systems can be divided into two broad categories. They are rule-based techniques and data-driven approach. Rule-based techniques, such as Formant synthesis [2] and Articulatory synthesis [2], try to synthesize speech using a fixed set of rigid rules mostly related to how the vocal system acts during the production of specific phonemes. In case of data-driven approach, natural pre-recorded speech units like words, syllables, phonemes, diphones, triphones etc. are used as data. In Concatenative synthesis [1] which is an example of data-driven approach, speech units are concatenated together to create speech. Statistical parametric speech synthesis [1] is another example of data-driven approach. Here speech is decomposed into parameters, such as acoustic features and duration features and the text is decomposed into various linguistic information. Then Hidden Markov Model(HMM) or Deep Neural Networks(DNN) can be used which will learn how to predict parameters such as acoustic features and duration features from the linguistic information of text data during the training phase [3]. In 2013, a research conducted by Heiga Ze et al. [4] showed that quality of speech synthesized by DNN can outperform that of conventional HMM with similar amount of training data and time. In 2016, researchers at DeepMind, created WaveNet [5], which is a text to speech synthesizer that uses a feedforward deep neural network. Wavenet has repeatedly outperformed existing text to speech systems in different experimental setups. Merlin [6] is an open source deep neural network based text to speech synthesis project by The Centre for Speech Technology Research (CSTR) of University of Edinburgh which uses Festival [7](an open source TTS engine) as its front end.

Bangla is the 6^{th} most spoken native language in the world by population with more than 200 million speakers using it as their first language. Yet text to speech synthesis in Bangla is an under-explored area. Katha, a Text to Speech Synthesis system is one of the few works that have been done for speech synthesis in Bangla language [8]. It has been developed using Festival. Festival requires a huge amount of language-specific data and grammatical rules to function and it uses Hidden Markov Model. But as far as we know, no significant work has been done on speech synthesis using neural networks in Bangla. So speech synthesis using artificial neural networks is an area that demands more attention for Bangla language.

Most of the currently existing text to speech synthesis systems require lots of language-specific rules and grammatical knowledge. They also require dictionary with

978-1-5386-7875-6/18/$31.00 ©2018 IEEE
DOI 10.1109/CSII.2018.00019

a proper phonetic breakdown of all the words of that language. But an end-to-end system tries to minimize human effort behind developing a system for text to speech synthesis in a particular language by creating a general text to speech synthesis system requiring data as minimal as <text, audio> pair. Nevertheless, an end-to-end system can benefit largely from language-specific knowledge and should keep options to add as many language-specific rules as possible. Most of the current works on end-to-end speech synthesis rely on deep neural networks. Tacotron [9] is an end-to-end system developed by Google which focuses on creating a deep neural network that needs almost no metadata, i.e. no language-specific knowledge is needed. But it still has a long way to reach the performance level of WaveNet. Ossian [10], another project of CSTR, is a toolkit for an end-to-end system which uses Merlin as its deep neural network but uses its own minimal front-end instead of Festival so that it does not have to rely on language-specific rules and grammatical knowledge. It requires data as minimal as transcribed audio data.

This paper proposes a methodology for implementing an end-to-end text to speech synthesis system for Bangla using a deep neural network and also proposes a Bangla language-specific text normalization procedure (TNP) to be incorporated into the system. The proposed TNP helps the front-end of the system in predicting information about individual tokens more accurately. We have implemented two separate models for speech synthesis in Bangla. After extensive experiments using speech synthesized by the models, we came to the conclusion that incorporating the proposed TNP has significantly improved the system.

II. Our proposed approach

In this section, we at first discuss the challenges of speech synthesis in Bangla and why we think separate text normalization procedures can improve speech synthesis systems for languages like Bangla. Then we discuss how we have implemented an end-to-end speech synthesis system for Bangla using neural networks and propose a Bangla language-specific text normalization procedure to be incorporated into the end-to-end system to improve its performance.

A. Challenges Faced in Bangla Language

Bangla language follows Abugida writing system [11] where the vowels can be attached to consonants and they are written as a unit. There are 11 vowel graphemes and 39 consonant graphemes in Bangla. Of these 11 vowels graphemes, there are two pairs of graphemes where, in each pair, one represents the short and another represents the long version of the same vowel sound. Also, there are 2 vowel diphthongs each of which can be further decomposed into 2 different vowels. So for vowels in Bangla text, we can see that there are scopes of normalizing the text to some extent. Also for consonants, two, three or four consonants can be concatenated together to create a consonant conjunct. For these consonant conjuncts, often it is the case that they are not pronounced as their individual components suggest. Rather separate rule sets are followed for deciding pronunciation of these conjuncts. So there is a scope of normalizing these consonant conjuncts to a form where their text form would be closer to their actual pronunciations. Many words and consonant conjunct forms in Bangla, have come from Sanskrit who have retained their original spelling but their pronunciations have gone through a lot of changes over the years and now they differ significantly from the pronunciation their spellings suggest. Even separate rules have been created to determine correct pronunciation of these words and conjuncts. This opens up new scopes of text normalization to make the spelling of text as close as possible to its actual pronunciation.

Now we will be discussing the different steps of speech synthesis and modules used in each step in our implementation.

B. Proposed Text Normalization Procedure (TNP) (Step 1)

We propose a text normalization procedure for Bangla here. As an input to the text to speech synthesizer, Bangla text will be given in UTF-8 format through an input device like a computer. The first step of our system is to normalize the text using our proposed TNP. Text normalization is the process of transforming text to a more consistent and easier to process form so that the next modules of the system have to deal with less complicated texts. There is no one consistent algorithm to normalize texts of all the languages. It is usually very language specific.

In our proposed TNP, the simplifications we have done are not all correct from the linguistic point of view but we hypothesized this would make the work of the neural network and other modules of the system less complicated and improve the quality of the synthesized audio at the same time. We have tried to overcome the challenges discussed in subsection II-A through the proposed TNP. More specifically, the normalization rules that we have proposed can be categorized into the following three categories:

- Replacing long version of vowels with the short version of the same vowel.
- Decomposing vowel diphthongs into two separate vowels.
- Breaking down consonant conjuncts to individual consonant graphemes and replacing some of the graphemes to other graphemes which actually represent the pronunciation of the conjunct. This decomposition of conjuncts can be affected by the position of the conjunct in the word which we have taken into account.

Table I
PROPOSED BANGLA TEXT NORMALIZATION (TNP)

Sl. No	Scenario	Unicode Breakdown with Phonetic Transcription	Phonetic Transcription of Widely Used Pronunciation	Alternative Used by Proposed TNP	Phonetic Transcription of the Alternative
1	ঈ [Explicit and Implicit vowel]	ঈ/ ী (ɪ)	ɪ	ই [Explicit and Implicit vowel]	ɪ
2	ঊ [Explicit and Implicit vowel]	ঊ/ূ (u)	u	উ [Explicit and Implicit vowel]	u
3	ঐ [Explicit and Implicit vowel]	ঐ/ ৈ (ɔɪ)	ɔɪ	ও [Explicit and Implicit vowel] + ই [Explicit vowel]	ɔɪ
4	ঔ [Explicit and Implicit vowel]	ঔ/ৌ (ow)	ow	ও[Explicit and Implicit vowel]+ উ [Explicit vowel]	ow
5	ঋ [Explicit and Implicit vowel]	ঋ/ ৃ (ɹɪ)	ɹɪ	র [Consonant] + ই-কার(ি) [Implicit vowel] or রি	ɹɪ
6*	ক্ষ [Consonant conjunct]	ক(k)+ষ (ʃ)	kh:	খ + য-ফলা or খ্য [Consonant conjunct]	kh:
7**	ক্ষ [Consonant conjunct]	ক(k)+ষ (ʃ)	kh	খ [Consonant]	kh
8	জ্ঞ[Consonant conjunct]	গ(g)+ঞ (n)	g:	গ + য-ফলা or গ্য [Consonant conjunct]	g:
9	ণ [Consonant]	ণ (n)	n	ন [Consonant]	n
10	ড় [Consonant]	ড় (ɽ)	:r	র	ɹ
11	ঢ় [Consonant]	ঢ় (ɽ)	ɽ	র	ɹ
12	ষ [Consonant]	ষ (ʃ)	ʃ	শ	ʃ
13*	ব্যঞ্জন [Any consonant] + য-ফলা [Diacritic form of the consonant য] + কার [Any implicit vowel]	ব্যঞ্জন [Any consonant] + য [Consonant] (z) + কার [Any implicit vowel]	Stressed pronunciation of that consonant	ব্যঞ্জন [Any consonant] + ব্যঞ্জন [That consonant] + কার [Any implicit vowel]	Stressed pronunciation of that consonant
14*	ব্যঞ্জন [Any consonant] + য-ফলা [Diacritic form of the consonant য]	ব্যঞ্জন [Any consonant] + য [Consonant] (z)	Stressed pronunciation of that consonant	ব্যঞ্জন [Any consonant] + ব্যঞ্জন [Any consonant] + ও -কার [Implicit vowel ও (ো)]	Stressed pronunciation of that consonant
15*	ব্যঞ্জন [Any consonant] + ব-ফলা [Diacritic form of the consonant ব]	ব্যঞ্জন [Any consonant] + ব [Consonant] (b)	Stressed pronunciation of that consonant	ব্যঞ্জন [Any consonant] + ব্যঞ্জন [Any consonant] + ও -কার [Implicit vowel ও (ো)]	Stressed pronunciation of that consonant
* Only when the conjunct appears at the middle of at the end of the word. ** Only when the conjunct appears at the start of the word.					

The rules we have used for the proposed TNP is summarized in Table I. All the phonetic transcriptions have been applied according to [12].

For example, using rule 6 and 9, the spelling of the word লক্ষণ (meaning: sign or symptom) becomes লখ্যন. Then using rule 14 recursively, লখ্যন becomes লখখোন. The proposed TNP modifies the conjunct at the middle and the last letter of the original spelling here. The letters of the original spelling of the word have the following phonetic transcription: ল(l), ক(k), ষ(ʃ), ণ(n). But the actual phonetic transcription of the word is lɔkhon. After the last phase, the letters of the word লখখোন with modified spelling have phonetic transcription: ল(l), খ(kh), খো(kho), ন (n). As we can see, the phonetic transcription of the letters of the modified word has a lot more in common than those of the letters of the original spelling.

C. Front-End Text Processing (Step 2)

In step 2, the front-end text processor will receive normalized Bangla text as input and give linguistic representation of the given text as output.

1) Linguistic Representation: Linguistic representations [3] mainly contain a tokenized version of the text with labels like the phoneme and contextual prosodic information. Contextual prosodic information is concerned with the pronunciation of that specific phoneme. In linguistics, prosody deals with the syllables or other larger units of speech rather than phonetic segments. A linguistic representation of a phoneme can contain information about many characteristics, such as the phoneme identity of the previous, current and the next phoneme, if the current syllable is stressed or not, the position of the current syllable in the current word, the number of accented syllables before the current syllable in the current phase etc. These, in fact, vary based on the front-end text processor used in the text to speech synthesizer.

2) Front-End text processor: There can be two types of front-end: trained front-end and minimal front-end.

A front-end text processor which is trained on a specific language using language-specific rules, grammar etc can be called a trained front-end, e.g. Front end text processors of Festival [7], Mary Text to speech synthesizer [13] etc. On the other hand, a minimal front-end can work with data as minimal as transcribed audio data. An end-to-end system by default must use a minimal front-end so that it does not have to depend on any language-specific expertise. The type of the front-end is the biggest difference between a language specific text to speech synthesis system and an end-to-end system.

We have used Ossian [10] as our front-end which is a minimal front-end. After the text has been normalized in step 1, Ossian tokenizes the normalized text into small tokens by using whitespace and punctuation marks as separators. It takes help from the Unicode table to classify characters of the text as punctuation marks or general characters of the alphabet. Then it creates linguistic representation of each of the individual tokens. It uses various pattern matching rules to create these linguistic representations. During the training phase, Ossian creates these pattern matching rules by analyzing patterns in the training text and audio data.

D. Prediction of Acoustic and Duration Feature using Neural Network (Step 3)

In the third step, a neural network will receive the linguistic representation generated in the second step as input and produce acoustic and duration features as output.

1) Acoustic And Duration Features: Acoustic features of an audio are totally language independent. There are mainly three important acoustic features relevant to our discussion which are fundamental frequency, F0 of the speech, spectral envelope and aperiodic energy. Duration features contain information about prosody and duration of the audio to be synthesized.

Neural Network We have used Merlin [6] as our statistical model. Merlin is an open source project developed by CSTR. The neural networks of Merlin treat the whole problem of text to speech synthesis as a sequence to sequence mapping regression problem. During the training phase, the neural network learns which sequence of acoustic features should be mapped to which sequence of linguistic representations and during synthesis phase, they try to predict a sequence of acoustic feature for each sequence of linguistic representation created by the front end, i.e. Ossian here. Merlin builds two separate feedforward neural networks, an acoustic model for acoustic feature prediction and a duration model for duration prediction. Duration model predicts the duration length, phrase breaks, intonation for different kinds of words, phonemes etc using information available from the linguistic representation of the text. During the training phase, one major problem is to determine which audio segment should be mapped to the linguistic representation of which token, since audio segments for two same sized tokens can be of very different lengths. Merlin uses a method named forced alignment to accomplish this.

To extract acoustic and duration features from training audio data, Merlin needs a vocoder which will extract them. We have used a vocoder named "WORLD" to extract them which are then used as class labels for the neural networks.

E. Speech Generation using Waveform Generator (Step 4)

In this step, a waveform generator receives acoustic and duration features generated in the third step as input and produce speech as output. All the training audio data were recorded in .wav format and used a mono channel with a sample rate of 48 kilohertz. The audio data synthesized by the system have the same characteristics. The "WORLD" vocoder is used as the waveform generator in this step.

III. Experimental Result and Discussion

Extensive experiments were conducted on data synthesized by two different models that we had created. We used two models in the experimentation process to get a more precise idea of how much the proposed TNP can affect or improve the performance of the system. We denote the models as model A and model B. Model B had the support of the proposed TNP whereas, model A did not have that. Apart from this, the basic structure of the models are the same and they both used Ossian as front-end, Merlin as statistical parametric model and World as waveform generator and were trained with the same training data set under the same environment. We did 3 different types of experiments and each of the experiment contained equal amount of data synthesized by both the models. We used 588 transcribed audio data from an audio book. Each data contained not more than 2 or 3 sentences.

A. Demographics of the Participants

There were 21 participants and all of them were undergraduate students from the Department of Computer Science and Engineering of University of Dhaka, Bangladesh.

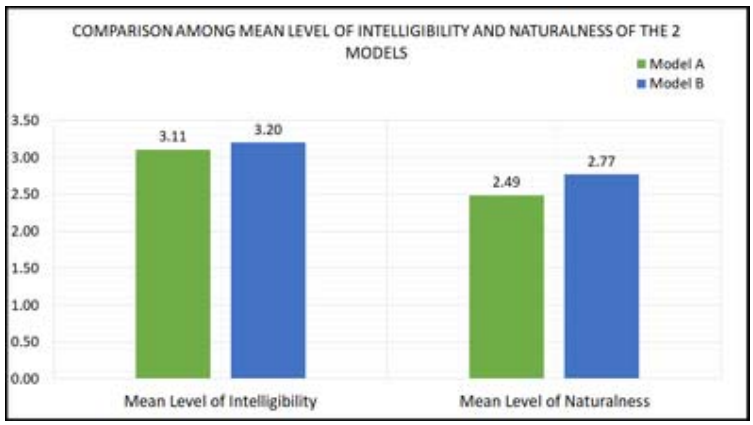

Figure 1. A column chart comparing the mean level of intelligibility and naturalness of the three models for Experiment 1-INLS.

All the participants are Bangladeshi by born and native Bangla speakers. All of them had used a text to speech synthesis system at least once before.

B. Experiment 1 - Intelligibility and Naturalness score on Likert Scale (INLS)

1) Experimental Setup: In this experiment, the participants were given a digital form where they were shown text first and then they could listen to the synthesized audio version of the text. There were in total 10 audio files. They had the option to play the audio as many times as they wish. They had to fill up a Likert-type survey questionnaire consisting of two questions after listening to each audio. The questions were about intelligibility and naturalness of the synthesized audios. In the Likert scale of the first question, 1 meant "Could not understand anything" and 5 meant "Understood completely". Similarly, for the second question, 1 meant "Extremely unnatural" and 5 meant "Fully natural".

2) Comparison of Mean Intelligibility and Mean Naturalness Score among the Models: A comparison of the mean scores is done using a column chart in Fig. 1.

From Fig. 1, we can see that model B had better scores than model A in both contexts. From this observed data, we can formulate a hypothesis that adding the proposed TNP has made model B better than model A both in terms of intelligibility and naturalness to the participants of this experiment. To prove that we can generalize our hypothesis, we have conducted a Wilcoxon signed-rank test.

3) Wilcoxon Signed-Rank test: Since our setup follows a repeated measures design, we concluded that Wilcoxon signed-rank test is the appropriate choice for our data analysis method. The null hypothesis for our experiment is that there is no significant difference between the two models and any observed difference is due to sampling and experimental error. Using Wilcoxon signed-rank test we can calculate the probability of the results being exactly like this if the null hypothesis were actually true. The sample size N for the test was 19 as we had to discard data of 2 participants since their data could not differentiate between the two models at all. We calculated the test statistic W for our experiment to be 40.5. Then we determined that the result of the experiment had a significance level $p \leqslant 0.05$ by taking help from values given for one-tailed tests in the table of critical values for Wilcoxon signed-rank test using that W value [14]. One-tailed tests are experiments where we already have a hypothesis that one of the models is better than the other and here we have claimed that model B is better than model A. This value of p tells us, if we hypothetically take the null hypothesis to be true, how much likely it was for the experiment to generate a scoring like this from the participants. Since our calculated p value is $\leqslant 0.05$, we can say that if the null hypothesis were actually true, the probability that the experiment could generate such a scoring distribution among the participants is less than or equal to 0.05.

C. Experiment 2 - Ranking Based on Intelligibility and Naturalness (RIN)

1) Experimental Setup: In this experiment, participants were given four groups of recordings. Each group consisted of one audio synthesized by each of the models. The participants were first shown the text and then asked to play the audio. After listening to both the audios of a group, they were asked to rank the audios based on intelligibility and naturalness with 1 being the best and 2 being the worst rank. The complexity level of all the sentences in the same group was higher than that of the sentences of the previous group. But within a group, the complexity of the sentences remained the same.

Table II shows the overall results of the two models and how the participants ranked them.
We can see that in case of group A, group C and group D, more participants ranked the audios generated from Model B as 1st.

Table II
Comparison among the number of votes per rank for the two models for experiment 2-RIN

	Group A		Group B		Group C		Group D	
Rank	Model A	Model B	Model A	Model B	Model A	Model B	Model A	Model B
1st	6	15	17	4	4	17	10	11
2nd	15	6	2	19	15	6	11	10

$$LCS(i,j) = \begin{cases} 0, & \text{if } i > |text| \vee j > |response|. \\ \max\left(LCS(i+1,j), LCS(i,j+1)\right), & \text{if } text_i \neq response_j. \\ LCS(i+1,j+1)+1, & \text{if } text_i = response_j. \end{cases} \quad (1)$$

D. Experiment 3 - Prediction Accuracy (PA)

1) Experimental Setup: In this experiment, participants were given four recordings but this time they were not shown any text data beforehand. Two synthesized audios from each model were taken. The participants had to write down what they have heard from the audio in Bangla. They were asked to put "?" mark in places where they could not understand anything.

2) Calculation of Accuracy: For experiment 3, we defined the accuracy of a participant for an audio to be the length of the longest common sub-sequence of the participants' prediction and the original text, divided by the length of the original text. We did not take the number of characters that the participant guessed incorrectly into account for the sake of simplicity.

We implemented a simple dynamic programming algorithm in Java for calculating the longest common subsequence between the answer of a participant and the original text. Equation 1 denotes the recurrence we used.

Here, we denote the original text as text and participants' individual answers as response. LCS(i,j) denotes the length of the longest common sub-sequence between the suffix of the text starting from the i^{th} index and suffix of the response starting from j^{th} index.

3) Accuracy of the participants and Analysis: The mean accuracy of the participants that we calculated are given in table III.

Table III
Comparison among the mean percentage of accuracy of the participants for the two models for Experiment 3-PA

Model	Accuracy for Simple Sentence	Accuracy for Moderately Complex Sentence
Model A	95.23%	60.05%
Model B	83.33%	73.80%

Finally we can conclude that in each of the three experiments, model B has outperformed model A in most of the cases.

IV. Conclusion

We implemented an end-to-end text to speech synthesis system for Bangla language and proposed a text normalization procedure for Bangla to improve its performance. With the data we have collected from the experiments, we have proved that the proposed TNP has improved the performance of the model by a good margin. We think that similar text normalization procedures can improve existing speech synthesis systems for languages close to Bangla such as Hindi, Tamil etc. Future scope of work may include implementing similar text normalization procedures for languages close to Bangla and analyzing its effect on existing speech synthesis systems for those languages.

References

[1] D. Siddhi, J. M. Verghese, and D. Bhavik, "Survey on various methods of text to speech synthesis," *International Journal of Computer Applications*, vol. 165, no. 6, 2017.

[2] Y. Tabet and M. Boughazi, "Speech synthesis techniques. a survey," in *Systems, Signal Processing and their Applications (WOSSPA), 2011 7th International Workshop on*. IEEE, 2011, pp. 67–70.

[3] S. King, "A beginners' guide to statistical parametric speech synthesis," *The Centre for Speech Technology Research, University of Edinburgh, UK*, 2010.

[4] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7962–7966.

[5] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[6] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," *Proc. SSW, Sunnyvale, USA*, 2016.

[7] "Festival," http://www.cstr.ed.ac.uk/projects/festival/, 18 Jun 2015, [Online; accessed 25-Dec-2017].

[8] F. Alam, S. M. Habib, and M. Khan, "Bangla text to speech using festival," in *Conference on Human Language Technology for Development*, 2011, pp. 154–161.

[9] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron: Towards end-to-end speech syn," *arXiv preprint arXiv:1703.10135*, 2017.

[10] "Ossian," http://simple4all.org/product/ossian/, [Online; accessed 25-Dec-2017].

[11] P. T. Daniels, "Fundamentals of grammatology," *Journal of the American Oriental Society*, pp. 727–731, 1990.

[12] S. ud Dowla Khan, "Bengali (bangladeshi standard)," *Journal of the International Phonetic Association*, pp. 221–225, 2010.

[13] "Marytts – introduction," http://mary.dfki.de/, 2016, [Online; accessed 25-Dec-2017].

[14] "The table of critical values for Wilcoxon Signed Rank Test," 2011. [Online]. Available: http://users.stat.ufl.edu/ winner/tables/wilcox$_s$*ignrank.pdf*