2019 International Conference on Bangla Speech and Language Processing (ICBSLP) 978-1-7281-5241-7/20/$31.00 ©2020 IEEE 10.1109/ICBSLP47725.2019.202055

A Bangla Text-to-Speech System using Deep Neural Networks

Rajan Saha Raju, Prithwiraj Bhattacharjee, Arif Ahmad, Mohammad Shahidur Rahman
Department of Computer Science and Engineering
Shahjalal University of Science and Technology
Sylhet, Bangladesh
{rajan10, prithwiraj12}@student.sust.edu, {arif_ahmad-cse, rahmanms}@sust.edu

***Abstract*—We present a Deep Neural Network (DNN) based statistical parametric Text-to-Speech (TTS) system for Bangla (also known as Bengali). A first step in building a DNN-based TTS system is having large speech data. Since good speech dataset for Bangla TTS is not available publicly, we created our own dataset for our system. We prepared a phonetically rich studio-quality speech database containing more than 40 hours of speech. The database consists of $12,500$ utterances. We also prepared a pronunciation dictionary (lexicon) of $1,35,000$ words for front-end text processing, which, to our knowledge, is the largest lexicon for Bangla. Our system extracts linguistic features from input text. Then it uses deep neural networks for mapping these linguistic features to acoustic features. We developed two TTS voices using our dataset - one male and one female voice. Both objective and subjective evaluation tests show that our system performs significantly better than the traditional Bangla TTS systems and is comparable to the commercially available best Bangla TTS system.**

***Index Terms*—spss, dnn, bangla speech corpus, lexicon, open source**

I. Introduction

Generating natural sounding speech from text (known as Text-to-Speech synthesis, or TTS) remains an interesting research problem despite decades of efforts. With the advent of smart devices, the necessity of TTS systems become more prevalent in recent days. TTS systems not only aid human-machine communication, but also help spreading knowledge and helping physically impaired people (e.g. blind people).

There are several ways to build a TTS system. Among them, two types of approaches dominant the development of TTS systems over the past few decades. One of the approaches, called concatenative synthesis [1] was the most popular method during the 1990s. Although this method produces highly natural synthetic voice, it requires a huge amount of human effort and expertise in specific areas. To build a typical concatenative unit-selection TTS, we need to record many hours of professional speech. Then we need to invest time in careful lexicon development and in creating complex rules for text normalization, among other things.

An attractive alternative of concatenative synthesis is the so called statistical parametric speech synthesis, or SPSS in short. The advantages of SPSS are the flexibility, control and small footprint, among others. In the SPSS systems, instead of concatenating smaller phonetic units, we generate acoustic parameters from which speech can be synthesized with the help of a vocoder. Among the earlier initiatives of developing SPSS systems, Hidden Markov Models (HMMs) based models [2] became common to the research community. A popular implementation of HMM-based approach is HTS [3], which was developed in the early 2000s. It has led the way in developing parametric synthesis approaches and algorithms. But due to the over-smoothing in acoustic modeling and the limitations of vocoders, HMM-based systems may not produce the most natural output [4].

The introduction of deep neural networks (DNNs) [5] opened a new research direction for acoustic modeling in SPSS [4] [6]. Although neural networks had been used in acoustic modeling in 1990s [7], it has shown impressive results [8] only in recent days thanks to the advancements in computational power and the availability of huge datasets. The research on SPSS systems gains a huge boost from that point on. The use of recurrent neural networks (RNNs) [4], long short-term memory RNNs (LSTM-RNNs) [9], and deep bidirectional LSTM-RNNs (BLSTM-RNNs) [10] improved the performance of SPSS systems significantly. SPSS systems thus become a strong contender of concatenative systems for producing natural sounding speech.

In spite of the advancements in the SPSS research, developing a good TTS system for an under-resourced language (like Bangla) still remains a challenging task. The lack of high quality speech data and the absence of language-specific text processing tools prevent us from building a high quality SPSS system. We intend to address these issues of Bangla TTS in this paper. The work we present here is based on the state-of-the-art algorithms of DNN-based synthesis. Our aim is to provide useful resources and recipes for developing Bangla TTS systems to the research community. Our contributions in this work include:

- We present a deep learning based Bangla Text-to-Speech system that can generate speech directly from Bangla text (in UTF-8 format) without any intermediate hand-engineered steps.
- We prepared 40 hours of studio-quality speech data for our TTS system. We employed two professional voice artists (one male and one female, both providing 20 hours of recordings each) to record the speech data. We plan

978-1-7281-5241-7/19/$31.00 ©2019 IEEE

to release these speech data under liberal open source license to the research community.

- We developed a lexicon[1] of 1,35,000 words that can be used in a front-end text processing tool. The lexicon can also be used to develop an automatic grapheme-to-phoneme module for speech synthesis. We will also release this lexicon as an open source data.

The rest of the paper is arranged in the following order: we discuss the related research works in section II. After that, we describe our data preparation process in section III. The following section explains the model architecture in detail. Section V analyzes the performance of our system. Finally, we end the paper with some concluding remarks and by giving some directions to future improvements.

II. Related Works

Works on developing Bangla TTS systems are rare. The initial attempts on building Bangla TTS used concatenative approaches. Firoz Alam et. al. from BRAC University developed *Katha* [11] in 2007, which is a unit-selection TTS system based on Festival [12] toolkit. Abu Naser et. al. from Shahjalal University of Science and Technology developed a di-phone concatenation based TTS system *Subachan* [13] in 2009. An HMM-based SPSS system [14] was developed in 2012.

After the introduction of DNNs for acoustic modeling in SPSS systems, the TTS research has been accelerated by manyfolds. But we do not notice any significant efforts to develop DNN-based Bangla SPSS in any Bangladeshi or Indian institutions, although a commercial Bangla TTS was released by Google [15] in 2016. Google also released some TTS resources [16] to encourage Bangla research community to work on speech synthesis.

TTS researches can be benefitted by many open source tools developed and shared by prominent institutes and by individual researchers. Idlak Tangle [17] is one such TTS system which is developed based on another open source speech recognition system Kaldi [18]. Merlin [19] is another open source DNN-based TTS system developed in the University of Edinburgh. The developers of Merlin uses Ossian [20] and Festival for front-end text processing, and the WORLD [21] vocoder for synthesizing speech from acoustic parameters. All of those tools are freely available under open source licenses. These resources are now being used to develop TTS systems for under-resourced languages, such as [22].

Although SPSS methods were originally preferred over concatenative methods due to their small footprint, they still require a huge amount of human intervention specially for language-specific text processing. To resolve this issue, modern researchers are trying to develop end-to-end TTS systems where speech is generated directly from (text, speech) pairs without any intermediate steps. Many of the giant tech companies have been engaged in this research. They have been getting promising results in the last couple of years. Example of such attempts are WaveNet [23], Tacotron [24], and Tacotron 2 [25] from Google, Deep Voice 1 [26], 2 [27], 3 [28] from Baidu, Char2Wav [29] from MILA, FastSpeech [30] from Microsoft, etc. Although the commercial companies do not share their code publicly, many individual implementations of these systems are available online which are almost as good as the original ones. New TTS researchers can be benefitted from these open source implementations.

We employed some good implementations of open source tools for our Bangla SPSS system, which will be discussed in section IV. The following section describes how we prepare our speech data.

III. Data Preparation

The first step of building a parametric voice is to collect a large amount of speech data, along with the associated transcriptions. There is no good quality public TTS data available for Bangla. Google has released their Bangla TTS data [16], but it contains only three hours of speech recorded with multiple speakers. A few more public dataset are available, all of which contains merely a couple of hours of speech. Those datasets were prepared for unit-selection TTS and for acoustical analysis of Bangla speech. So we needed to prepare our dataset from scratch.

At first, we consulted the literature for developing *phonetically balanced* text corpus [31]–[33]. Then we gather text data from various domains. We ensured that our text corpus contains all possible pronunciations of Bangla. Our final corpus contains more than 12,000 utterances, each consisting of standalone sentences. A summary of our dataset is presented in table I.

TABLE I
Summary of speech data prepared for Bangla SPSS

Total sentences	$12,537$
Total words	$1,22,627$
Total unique words	$24,582$
Minimum words in a sentence	3
Maximum words in a sentence	20
Average words in a sentence	9.78
Total duration of speech (hours)	$20:14:21$
Average duration of each sentence (seconds)	5.81

After preparing the text corpus, we started recording the speech. We have built a sound-proof audio recording studio solely for this purpose. Then we employed two professional voice artists (one male and one female), both of whom have a clear, strong voice and can record speech for several consecutive hours. We recorded their voices in the raw wave format, at a sampling rate of 48 KHz. The total duration of collected speech is around 20 hours (for each speaker).

IV. Model Architecture

Figure 1 shows the steps in building the Bangla SPPS system. The major processing units of this system are: (i) a text normalizer, (ii) a front-end for processing text input, (iii) & (iv) two deep neural networks (DNNs) for duration and acoustic modeling, and (v) a vocoder for synthesizing speech

[1] A lexicon is a dictionary of pronunciation.

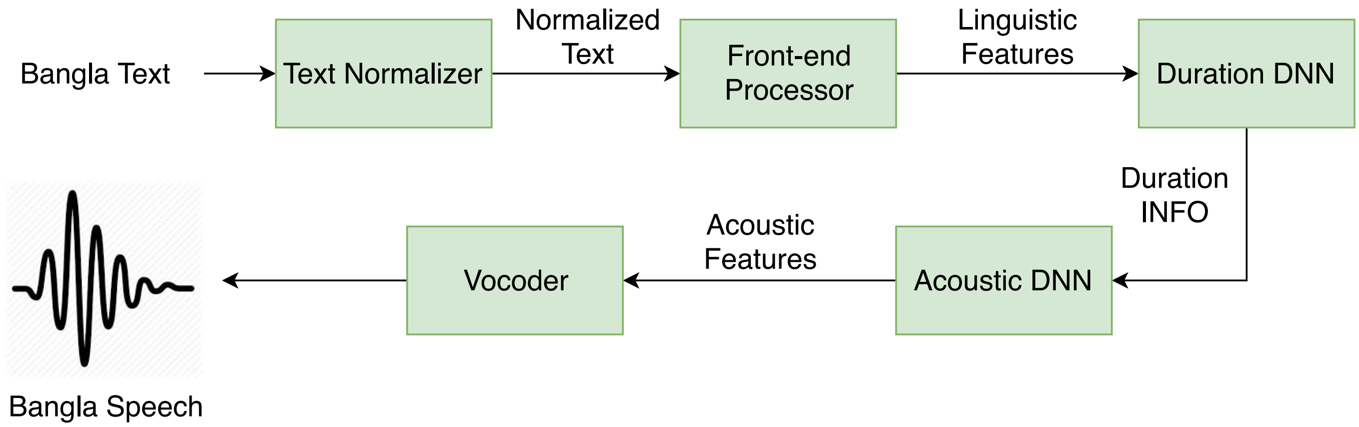


Fig. 1. Architecture of Bangla SPSS

from acoustic parameters. An elaborate description of these units are given below.

A. Text Normalizer

Text normalization is the process that converts non-standard words (NSWs) into pronunceable forms. In a typical TTS system, the raw input text is passed to a text normalizer, and a *normalized* text is produced as output. We developed a text normalizer for our Bangla SPSS system. To develop the normalizer, we consulted the literature thoroughly and implemented some methods proposed by the researchers [34].

B. Front-end

A DNN-based TTS systems need a front-end text processor to get the linguistic features from the input text. We have successfully utilized two open source front-end tools - Ossian [20] and Festival [12].

The Ossian open-source text processor allows us to extract linguistic features from the input text. It is language-independent, and can work with any text given in the UTF-8 format. It performs the feature extraction by mapping each character of the input text to a corresponding phoneme. Although modeling speech units from characters is a naive technique, it works surprisingly well if provided with a sufficiently large dataset and a phonetically balanced text corpus.

To obtain a better control over the linguistic feature computations, we might want to use a language-specific front-end tool. The Festival speech synthesis tool can help us in this regard. The front-end text processor of festival is language specific. It requires a well-defined lexicon and phonology (collection of phonetic information) of the target language. We have prepared a lexicon of 1,35,000 words and used a grapheme-to-phoneme converter [35] to handle out-of-vocabulary words. We have adopted and updated a phonology released by Google [16].

The front-end outputs HTS-style [36] labels with the state-level alignment. From these labels, a vector of linguistic features are generated by Ossian (or Festival). This feature vector is then fed to the duration model and acoustic model.

C. Duration Model

The back-end of our Bangla TTS system consists of two deep neural networks. The first one, duration DNN, takes linguistic features generated from front-end processing as input, and learns the proper duration information by updating weights. We split the training data into three sets: training (94%), testing (3%) and validation (3%).

We employed feed forward networks for both duration and acoustic modeling. The Merlin toolkit allows us to use other variants of neural networks as well. The basic architecture of the preferred network can be specified in a configuration file and the model will perform accordingly. We get this flexibility due to the availability of recipe-like structure implemented by the Merlin developers.

Our duration DNN consists of 3 layers of hidden units where each layer contains 512 neurons. The network uses Gradient Descent optimizer with the learning rate of 0.002. The output of this network is then fed to the acoustic model.

D. Acoustic Model

The acoustic DNN was trained to map the input linguistic features and the associated duration features into acoustic features. Linguistic features (sequence binary vectors) are normalized in the range of [0.01, 0.99] before passing to input layers of DNN. The acoustic DNN consists of 6 layers of hidden units where each layer contains 1024 neurons. The network applies the Gradient Descent algorithm with learning rate 0.002 to minimize the errors between predicted outputs and target outputs and updates the weights in every iteration.

E. Vocoder

Acoustic features, the output of acoustic modeling DNN are normalized appropriately so that they can be used by a vocoder. They are reduced to zero mean and unit variance. Finally, acoustic features are sent to a vocoder for synthesizing waveform. We used WORLD [21], an open source vocoder modified by Merlin for making compatible with the entire system.

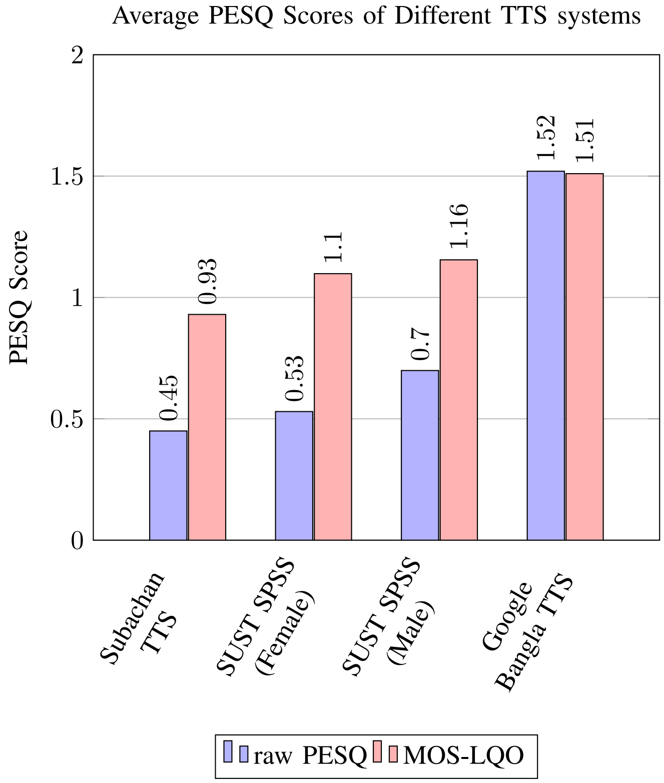
V. Results and Discussion

To assess the quality of our proposed system, we conducted both objective and subjective evaluation.

A. Objective Evaluation

We choose the Perceptual Evaluation of Speech Quality (PESQ) [37] score, specifically raw-PESQ and MOS-LQO for the purpose of objective evaluation. The intervals of raw-PESQ and MOS-LQO are [-0.5, 4.5] and [1, 5], respectively. In the measurement of PESQ, two waveforms (one is original and the other is synthetic) are required. The experimental set-up for determining the PESQ score is as follows.

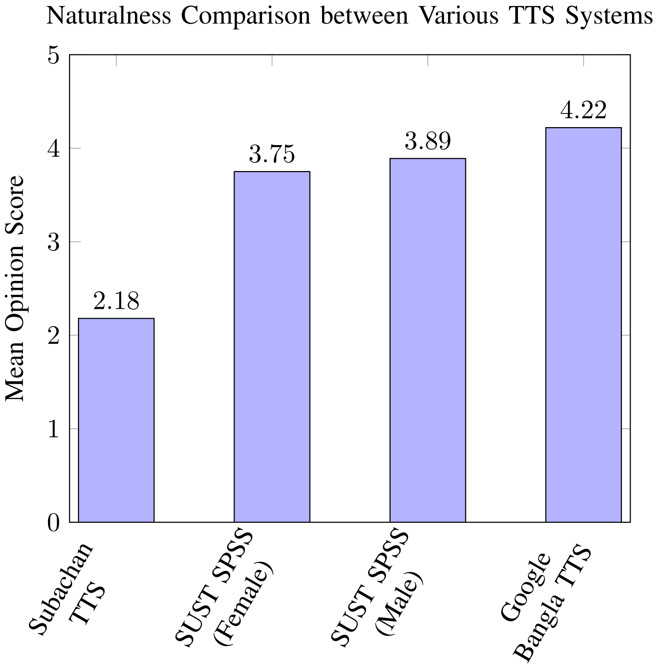
We calculated the PESQ scores of 4 systems (Subachan TTS, SUST SPSS Female, SUST SPSS Male, Google Bangla TTS) to compare speech quality. For calculating the PESQ scores of the TTS systems, we picked 100 random sentences and corresponding speech recordings as original speech data. Then, we synthesized 100 waveforms of selected sentences from the TTS systems as synthetic speech. Finally, the original and synthetic speech was sent to the PESQ system in pairs for determining the PESQ score. The following bar chart shows the average PESQ scores for the four systems mentioned above.



B. Subjective Evaluation

A benchmarking test for the TTS system is the Mean Opinion Score (MOS). So we selected the MOS test for our subjective evaluation. We invited some native Bangladeshi speakers to test the naturalness of various systems. 30 people volunteered us as the listeners for the MOS test. The listeners are aged between 20 and 35. 18 of them were male, rest were female. The listeners listened to 20 synthetic sentences generated by various TTS systems and gave a naturalness score between 0 and 5 to each of the systems. A higher score means better naturalness. All the scores are averaged to obtain the *mean score* of a system. In fact, we excluded the highest and lowest score obtained by each system to calculate a more accurate MOS named Robust-MOS.

The following bar chart summarizes the MOS scores obtained for various systems. The results of our MOS test supports our objective evaluation tests as well. Our system clearly outperforms the publicly available concatenative TTS system *Subachan*. Our male voice performs slightly better than the female voice. Both of the voices are comparable with the best known commercial Bangla TTS from Google.



VI. Conclusion

In this paper, we aim to address some of the challenges in the area of Bangla SPSS. We developed a large speech corpus and efficiently trained a deep neural network to synthesize Bangla speech. We also developed a large lexicon that will help researchers build a robust TTS front-end. Experimental analysis shows that our system performs significantly better than the traditional TTS systems, in terms of naturalness. Despite the promising performance, we still need to work harder in improving the scalability of the system, linguistic components, lexicon and text normalization. We hope that, with the availability of resources we developed, our research community will come forward to contribute more to Bangla speech synthesis research.

References

[1] A. J. Hunt and A. W. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1. IEEE, 1996, pp. 373–376.

[2] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis,” in *Sixth European Conference on Speech Communication and Technology*, 1999.

[3] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, “The hmm-based speech synthesis system (hts) version 2.0.” in *SSW*. Citeseer, 2007, pp. 294–299.

[4] H. Ze, A. Senior, and M. Schuster, “Statistical parametric speech synthesis using deep neural networks,” in *2013 ieee international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 7962–7966.

[5] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, “Deep neural networks for acoustic modeling in speech recognition,” *IEEE Signal processing magazine*, vol. 29, 2012.

[6] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. M. Meng, and L. Deng, “Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and tuture trends,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.

[7] T. Weijters and J. Thole, “Speech synthesis with artificial neural networks,” in *IEEE International Conference on Neural Networks*. IEEE, 1993, pp. 1764–1769.

[8] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, “From hmms to dnns: where do the improvements come from?” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5505–5509.

[9] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, “On the training aspects of deep neural network (dnn) for parametric tts synthesis,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3829–3833.

[10] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, “Tts synthesis with bidirectional lstm based recurrent neural networks,” in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[11] F. Alam, P. K. Nath, and M. Khan, “Text-to-speech for bangla language using festival,” 2007.

[12] “The festival speech synthesis system,” accessed: 2019-07-28. [Online]. Available: http://www.cstr.ed.ac.uk/projects/festival/

[13] A. Naser, D. Aich, and M. R. Amin, “Implementation of subachan: Bengali text-to-speech synthesis software,” in *International Conference on Electrical & Computer Engineering (ICECE 2010)*. IEEE, 2010, pp. 574–577.

[14] S. Mukherjee and S. K. D. Mandal, “A bengali hmm based speech synthesis system,” *arXiv preprint arXiv:1406.3915*, 2014.

[15] A. Gutkin, L. Ha, M. Jansche, K. Pipatsrisawat, and R. Sproat, “Tts for low resource languages: A bangla synthesizer,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, 2016, pp. 2005–2010.

[16] “Google international language resources,” accessed: 2019-07-28. [Online]. Available: https://github.com/google/language-resources/blob/master/bn/festvox/phonology.json

[17] B. Potard, M. P. Aylett, D. A. Baude, and P. Motlicek, “Idlak tangle: An open source kaldi based parametric speech synthesiser based on dnn.” in *INTERSPEECH*, 2016, pp. 2293–2297.

[18] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,” in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[19] Z. Wu, O. Watts, and S. King, “Merlin: An open source neural network speech synthesis system.” in *SSW*, 2016, pp. 202–207.

[20] “Ossian: A simple language independent text to speech front-end,” accessed: 2019-07-28. [Online]. Available: https://github.com/CSTR-Edinburgh/Ossian

[21] M. Morise, F. Yokomori, and K. Ozawa, “World: a vocoder-based high-quality speech synthesis system for real-time applications,” *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[22] A. Deka, P. Sarmah, K. Samudravijaya, and S. Prasanna, “Development of assamese text-to-speech system using deep neural network,” in *2019 National Conference on Communications (NCC)*. IEEE, 2019, pp. 1–5.

[23] A. v. d. 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.

[24] 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 synthesis,” *arXiv preprint arXiv:1703.10135*, 2017.

[25] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[26] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, A. Ng, J. Raiman *et al.*, “Deep voice: Real-time neural text-to-speech,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 195–204.

[27] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou, “Deep voice 2: Multi-speaker neural text-to-speech,” in *Advances in neural information processing systems*, 2017, pp. 2962–2970.

[28] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, “Deep voice 3: Scaling text-to-speech with convolutional sequence learning,” *arXiv preprint arXiv:1710.07654*, 2017.

[29] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, “Char2wav: End-to-end speech synthesis,” 2017.

[30] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, “Fastspeech: Fast, robust and controllable text to speech,” *arXiv preprint arXiv:1905.09263*, 2019.

[31] P.-E. Honnet, A. Lazaridis, P. N. Garner, and J. Yamagishi, “The siwis french speech synthesis database? design and recording of a high quality french database for speech synthesis,” Idiap, Tech. Rep., 2017.

[32] R. Sonobe, S. Takamichi, and H. Saruwatari, “Jsut corpus: free large-scale japanese speech corpus for end-to-end speech synthesis,” *arXiv preprint arXiv:1711.00354*, 2017.

[33] L. Gabdrakhmanov, R. Garaev, and E. Razinkov, “Ruslan: Russian spoken language corpus for speech synthesis,” *arXiv preprint arXiv:1906.11645*, 2019.

[34] M. M. Rashid, M. A. Hussain, and M. S. Rahman, “Text normalization and diphone preparation for bangla speech synthesis.” *Journal of Multimedia*, vol. 5, no. 6, pp. 551–559, 2010.

[35] A. Ahmad, M. Raihan Hussain, M. Reza Selim, M. Zafar Iqbal, and M. Shahidur Rahman, “A sequence-to-sequence pronunciation model for bangla speech synthesis,” in *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, Sep. 2018, pp. 1–4.

[36] H. Group *et al.*, “Hmm/dnn-based speech synthesis system (hts),” *Available in: http://hts. sp. nitech. ac. jp*.

[37] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, “Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, vol. 2. IEEE, 2001, pp. 749–752.