Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020
30 October 2020, Shanghai, China

The Duke Entry for 2020 Blizzard Challenge

Zexin Cai, Ming Li

Electrial & Computer Engineering, Duke University, Durham, NC, United States

ming.li369@duke.edu

Abstract

This paper presents the speech synthesis system built for the 2020 Blizzard Challenge by team 'H'. The goal of the challenge is to build a synthesizer that is able to generate high-fidelity speech with a voice that is similar to the one from the provided data. Our system mainly draws on end-to-end neural networks. Specifically, we have an encoder-decoder based prosody prediction network to insert prosodic annotations for a given sentence. We use the spectrogram predictor from Tacotron2 as the end-to-end phoneme-to-spectrogram generator, followed by the neural vocoder WaveRNN to convert predicted spectrograms to audio signals. Additionally, we involve finetuning strategics to improve the TTS performance in our work. Subjective evaluation of the synthetic audios is taken regarding naturalness, similarity, and intelligibility. Samples are available online for listening. [1]

Index Terms: Text-to-speech, speech synthesis, prosody model, end-to-end

1. Introduction

The blizzard challenge is held annually to better understand and compare techniques in speech synthesis among individuals and institutes around the world [1]. The sixteen challenge held in 2020 contains two independent tasks. One of the tasks is to build a synthetic voice with 9.5 hours of Mandarin speech data spoken by a male native speaker, while the other task is to build a female voice to speak Shanghainese, a Chinese dialect, with 3 hours of speech data. The two tasks are identified as '2020-MH1' and '2020-SS1', respectively. Subjective evaluation is taken to measure the synthesis performance regarding several aspects, including naturalness, speaker similarity, and intelligibility.

Speech synthesis, also known as text-to-speech (TTS), typically refers to the technique that converts graphemes to audio signals. By adopting deep neural networks, researchers are able to develop TTS systems that can synthesize high-fidelity speech in recent years. For traditional methods like concatenating synthesis and statistic parametric speech synthesis (SPSS), incorporating deep neural network in part of the TTS pipeline can significantly benefit the performance in terms of naturalness and prosody [2, 3]. End-to-end TTS systems, compared to the traditional methods, was introduced with a less heavy pipeline and less handcrafted features. Moreover, the speech produced by end-to-end TTS systems can be very close to the natural speech spoken by human [4, 5, 6, 7]. However, either traditional TTS approaches or end-to-end based approach has been adopted and investigated by participants from previous Blizzard challenges.

This paper describes the entry built for the '2020-MH1' task. The system is extended from the system we built for 2019 Blizzard Challenge [8]. The input unit for spectrogram prediction is changed to phoneme. We also incorporate a prosody estimation module and use a neural vocoder for synthesizing audio signals. In our entry, we rely on neural network models to meet the goal of 2020 Blizzard Challenge. Specifically, there are three individual models:

- The end-to-end prosody prediction model for generating prosodic annotations. [9]
- The Tacotron-based text-to-spectrogram model that predicts Mel-spectrograms from phoneme sequences. [4]
- The neural vocoder WaveRNN that converts the predicted spectrograms into audio signals. [10]

In order to reduce the learning efforts caused by alphabet-level input representations, we use the phoneme sequence as the input. As for training strategies, we use a speech recognition model for obtaining the alignment and segment the long utterances into shorter pieces for pre-training. The model is fine-tuned with long concatenated utterances for stable performance in terms of long-form synthesis. External text data is used for prosody model training, and about 82 hours of accompaniment speech data are used for vocoder training.

This paper is organized as follows. Section 2 introduces the TTS pipeline and neural network architectures we employed for our entry. External Data, training strategies, and inference details are described in section 3. Results are shown in section 4, and conclusions are drawn in section 5.

2. Methods

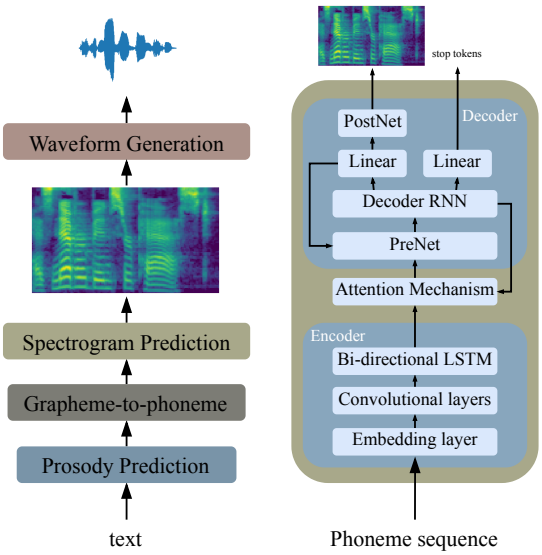
In our entry, the text-to-speech pipeline goes as (a) in figure 1. For a target text, the prosody prediction module adds prosodic annotations, which represent different pause durations, between words and phrases. The sentence with prosody annotations is then converted to the phoneme sequence by the grapheme-to-phoneme module. The phoneme sequence is taken as the input of the spectrogram prediction module, shown in (b) from figure 1, to generate the corresponding Mel-spectrogram. At last, the waveform generation module converts the frequency-level spectrogram to time-level audio signals. Except for the grapheme-to-phoneme module that is performed by using the Pypinyin toolkit [11], the other three modules are built with network-based models and trained separately.

2.1. Prosody prediction model

The prosody boundary, which normally refers to the pauses within human speech, plays an important role in Mandarin speech synthesis [12]. In general, various pauses that happen between words and phrases can deliver a rich rhythm that creates natural emotions in speech. Therefore we incorporated prosodic annotations into TTS training to further improve the naturalness of the synthetic voice. To this end, we adopt an end-to-end prosody prediction model to generate prosodic annotations for a given sentence at the inference phrase [9]. Three kinds of annotations referring to different pause durations are used in our system. One is the pause of the prosodic word

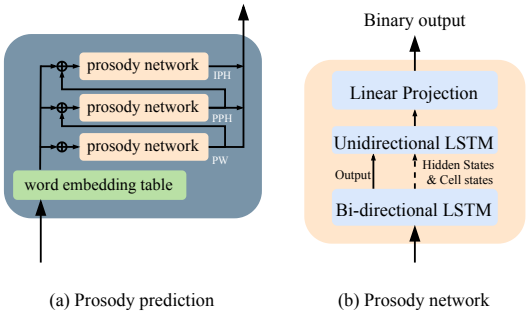
[1] https://caizexin.github.io/blizzard_2020/index.html

10.21437/VCC_BC.2020-5



(a) Text-to-speech framework (b) phoneme-to-spectrogram

Figure 1: *The overall Text-to-speech pipeline and the Mel-spectrogram prediction encoder-decoder framework.*



(a) Prosody prediction (b) Prosody network

Figure 2: *Prosodic annotation prediction network*

(PW) that denotes the short pause that appears when the former word is pronounced with stress. The second one is the prosodic phrase (PPH) that represent a certain level of pause between words or phrase where the tone before it is pronounced like the end of the sentence but actually not. The last prosody boundary is the intonational phrase (IPH) that indicates a long pause and normally occurs between sentences.

The prosody model we applied in our TTS system is shown in (a) from figure 2. The input text is converted to word embedding sequence by a pre-trained word embedding lookup table [13]. Three consecutive prosody networks are followed to predict whether to add a specific prosody boundary annotations at each time step. The prosody network is constructed by an encoder-decoder architecture shown in (b) from figure 2, where the encoder contains a bi-directional LSTM layer and the decoder has a unidirectional LSTM layer followed by a linear projection layer that provides prosodic boundary predictions. Whether to add the corresponding prosodic annotation after each input word is dependent on the binary output. However, only one prosody boundary needs to appear after each word. So after the independent binary decisions are made for the three annotations, we use the labeling priority following **IPH** $>$ **PPH** $>$ **PW** to select one as the final prediction when more than one annotation are predicted to be existed by the model.

2.2. Synthesizer

The synthesizer includes the spectrogram prediction module and the waveform generation module. We use Mel-spectrogram as the acoustic feature to bridge two modules. The spectrogram prediction network is presented in (b) from figure 1. This is the attention-based encoder-decoder network from Tacotron2 [4]. We use phoneme set from CMU dictionary [14] as the input unit instead of alphabet characters to free the model from learning complex pronunciation rules. The encoder contains several layers of convolutional neural networks (CNN) that collect the local context information. Then a bi-directional LSTM is applied to obtain long-term context information.

The attention module calculates the most related context at each step with respect to the previous decoding output and the encoder output states. Thus the decoder knows which specific phoneme needs to be pronounced at every decoding time step. In addition, the attention module can perform soft alignment between the phoneme sequence and the output Mel-spectrogram. The PreNet in the decoder contains several fully connected linear layers, while the PostNet is constructed by several convolutional layers to reduce the information loss caused by the unidirectional decoding characteristic of the decoder. The stop tokens are to predict the end of the decoding process. The hyperparameter settings of the spectrogram prediction module are shown in table 1.

We adopt the neural vocoder WaveRNN [10] for backend waveform reconstruction in our entry. As for a neural vocoder, WaveRNN can achieve the same performance as the state-of-the-art model WaveNet, while WaveRNN has a faster waveform generation speed.

Table 1: *Hyperparameters of the phoneme-to-spectrogram model, including those start with 'Feature/' for Mel-spectrogram extraction.*

Hyperparameter	
Feature/number of Mel bands	80
Feature/FFT window length	800
Feature/hop length	200
Feature/frame window size	800
Feature/preemphasis	0.97
Feature/lowest frequency	55
Feature/highest frequency	7600
Encoder/embedding dimension	512
Encoder/number of Conv layers	3
Encoder/Conv kernel size	(5,)
Encoder/Conv channel size	512
Encoder/LSTM units per direction	256
Output frames per decoding step	1
Decoder/Attention dimension	128
Decoder/Attention filters	32
Decoder/Attention kernel	(31,)
Decoder/PreNet linear layers	[256, 256]
Decoder/number of LSTM layers	2
Decoder/LSTM units	1024
Decoder/PostNet Conv layers	3
Decoder/PostNet Conv kernel size	(5,)
Decoder/PostNet Conv channel size	512

3. System training and inference

3.1. Data

The Mandarin dataset provided by 2020 Blizzard Challenge contains 4365 audio-text pairs recorded by a male voice. Provided audios are formatted in 48 kHz sampling rate. The total duration of the dataset is about 9.5 hours. The provided dataset from Blizzard Challenge is notated as 'BC' in this paper. Other than the provided data, we also use external datasets for training. We have three commercial datasets bought from Data Baker [2]. One is the DB-1 dataset that contains 12 hours of speech data with a female voice. The second one is the DB-4 bilingual dataset that contains 18 hours of data recorded by a native Chinese female. There are 15000 audio-text pairs in this dataset, where 5000 of them are in English, and 10000 of them are in Mandarin. The third dataset is recorded by a native Chinese male, and there are about 10 hours in total. These three datasets are notated as 'DB1', 'DB4' and 'DBM' in the following paper. In addition, we also use a private dataset containing 18 hours of audio-text pairs for training. We notated this dataset with 'BLM' here. We also use a publicly available dataset LJ-Speech [15], which contains 24 hours of English data recorded by a female and is notated as 'LJS' in our paper. All audios are downsampled to 16 kHz in our training setup.

The transcripts from datasets 'DB1', 'DB4' and 'DBM' have prosodic boundary labels. So we use the transcripts from those three datasets to train the prosody prediction model. For this united set, 80% of the transcripts are used as the training set, while the remaining are used for validation. The spectrogram prediction model is trained only with dataset 'BC'. The neural vocoder WaveRNN is pre-trained with all datasets we mentioned above and then is finetuned with the ground-truth alignment Mel-spectrograms of dataset 'BC' obtained from the trained spectrogram prediction model.

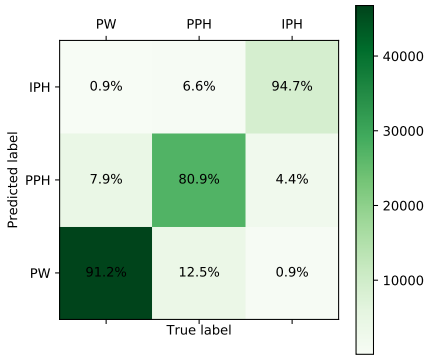


Figure 3: *The confusion matrix of prosody prediction result*

3.2. Training

We use a speech recognition model to perform force alignment of text-audio pairs from the provided data 'BC'. According to the alignments, we further segment the pairs into short utterances that have a duration ranging from 2 seconds to 9 seconds. After the segmentation, we have 7768 text-audio pairs for training. Prosody annotations are added to the transcript by the prosody prediction model. After converting the transcripts into phoneme sequences, then we asked a native speaker to check and revise the sequence regarding the tone annotations and prosody annotations. This processed data is used for training the phoneme-to-spectrogram model.

To improve our system's performance on the long-formed synthesis, we randomly concatenate short utterances to obtain long utterances for training. Specifically, 2000 utterances are obtained by concatenating two short utterances, 1500 utterances are obtained by concatenating three short utterances, 1000 utterances from four-concatenation, and 500 utterances from five-concatenation. The phoneme-to-spectrogram model is first pre-trained with all short utterances. Then the pre-trained model is finetuned with all utterances, including the concatenated utterances.

The neural vocoder WaveRNN is trained with all available audio data mentioned in section 3.1. To improve the vocoder's performance on reconstructing waveform from synthetic Mel-spectrogram, we use ground-truth alignment (GTA) Mel-spectrograms to finetune the neural vocoder. GTA Mel-spectrograms are obtained by feeding the training set 'BC' into the trained phoneme-to-spectrogram network.

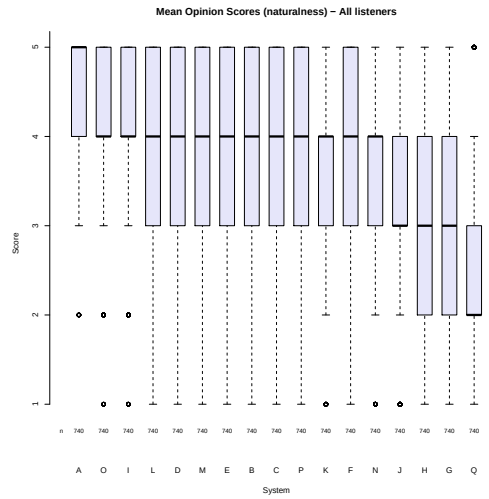


Figure 4: *The naturalness mean opinion scores (MOS)*

3.3. Inference

Three kinds of synthetic sentences are required for completing the '2020-MH1' task. The first task 'INT' is to synthesize semantically unpredictable sentences for testing the intelligibility. The second task 'NEWs' is to synthesize in-domain sentences for evaluating the performance concerning naturalness and speaker similarity. The task 'PSC' is also designed for evaluating naturalness and similarity but related to out-domain cases. For 'NEWs' and 'PSC', participants are required to synthesize both sentences and paragraphs.

For the 'INT' task, we used Jieba package [3] to tokenize sentences. Then add the prosodic phrase (PPH) annotation between the words and phrases for generating pauses. As for the other two tasks, the prosodic annotations are predicted by the prosody model. Then the character sequence with prosody label is then

[2] https://www.data-baker.com

[3] http://pypi.python.org/pypi/jieba/

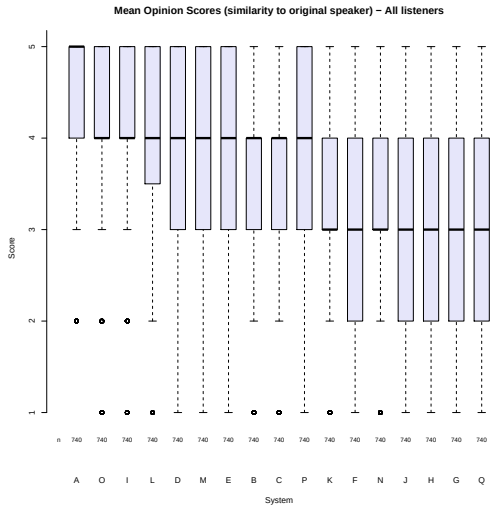


Figure 5: *The speaker similarity mean opinion scores (MOS)*

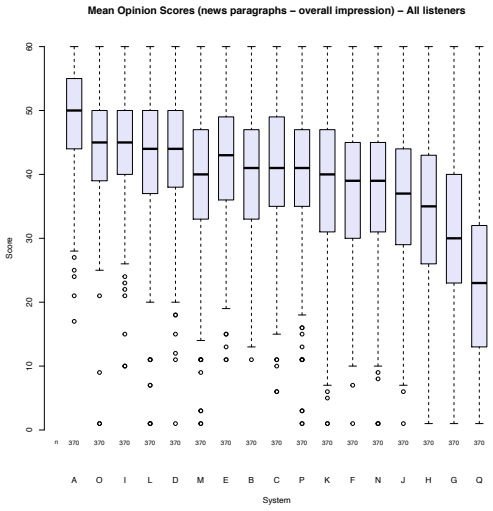


Figure 7: *Overall impression on synthetic voice*

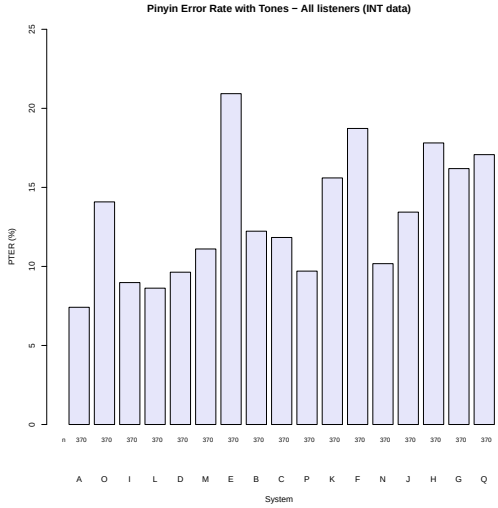


Figure 6: *The intelligibility mean opinion scores (MOS)*

converted to pinyin sequence by the Pypinyin package [4]. We use the phoneme set from CMU dictionary [14] in our entry. The pinyin sequences are converted to phoneme sequences by the pinyin-to-cmu mapping table. The synthesis pipeline is the same as the pipeline shown in (a) from figure 1. For paragraph synthesis, we simply concatenate the synthetic sentences and add silence between them.

4. Results

The performance of our prosody model is shown in figure 3. The accuracy of predicting prosodic annotations PW, PPH, IPH are 91.2%, 80.9%, and 94.7%, respectively. 12.5% of PPH annotations are predicted as PW annotations, as shown in the confusion matrix. However, the pause duration difference between PPH and PW is small since both of them are word-level pauses. Therefore minor mispredictions would not affect the overall performance in the TTS system.

Subjective evaluations are conducted regarding intelligibility, similarity and naturalness. Paid listeners, speech experts and volunteers are asked to listen and score the synthetic audios online. There are 16 participated teams in total this year. Our team is notated as system 'H'. System 'A' is the natural human voice for reference.

Naturalness reports the quality of synthetic speech. Scores are rated by the listeners from scale 1 to 5, where scale 1 refers to unbearable synthetic voice, and 5 means high-fidelity voice. The mean opinion scores (MOS) of naturalness is shown in figure 4. Our system achieves an average score of 3. The similarity evaluation measures how close between the synthetic voice and the natural one. As shown in figure 5, the voice we build is moderately close to the natural voice.

Figure 6 presents the result evaluated concerning intelligibility, which is to test how accurate listeners can catch the contents from synthetic speech. Listeners are asked to type in what they heard from synthetic audios during evaluation. The pinyin error rate is 17.8% for our entry. In addition, listeners are also required to evaluate the synthetic results of all participants in various aspects, including pleasantness, stress, intonation and emotion. The overall impression is shown in figure 7, where the synthetic voice we delivered achieves a score of around 30.

5. Conclusions

In this paper, we present our TTS system designed for 2020 Blizzard Challenge. Prosodic annotations are incorporated into the system to improve the naturalness. We adopt the Tacotron-based phoneme-to-spectrogram system to predict the acoustic feature, and the waveform reconstruction is performed by the neural vocoder WaveRNN.

To further improve our system in the future, we need to address issues related to character-to-pinyin conversion: especially the polyphone disambiguation problem and the sandhi problem. Concerning the naturalness, the prosody model we adopted in our entry is able to benefit the performance. However, we can investigate more acoustic variances, e.g., the fundamental frequency (F0), to make the synthetic voice closer to the natural human voice.

[4] https://pypi.org/project/pypinyin/

6. References

[1] A. W. Black and K. Tokuda, "The blizzard challenge-2005: Evaluating corpus-based speech synthesis on common datasets," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[2] H. Shi, X. Zhou, J. Li, L. Xiao, and W. Zengfu, "The IIM System for Blizzard Challenge 2019," in *Blizzard Challenge Workshop*, 2019.

[3] 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*, 2013, pp. 7962–7966.

[4] 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*, 2018, pp. 4779–4783.

[5] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," in *International Conference on Learning Representations*, 2019.

[6] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," in *International Conference on Learning Representations*, 2018.

[7] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "FastSpeech: Fast, Robust and Controllable Text to Speech," in *Advances in Neural Information Processing Systems*, 2019, pp. 3171–3180.

[8] Z. Cai, C. Zhang, Y. Yang, and M. Li, "The DKU Speech Synthesis System for 2019 Blizzard Challenge," in *Blizzard Challenge Workshop*, 2019.

[9] C. Zhang, S. Zhang, and H. Zhong, "A prosodic mandarin text-to-speech system based on tacotron," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2019, pp. 165–169.

[10] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, "Efficient neural audio synthesis," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 2410–2419.

[11] https://github.com/mozillazg/python-pinyin.

[12] C. Lu, P. Zhang, and Y. Yan, "Self-attention based prosodic boundary prediction for chinese speech synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 7035–7039.

[13] Y. Song, S. Shi, J. Li, and H. Zhang, "Directional skip-gram: Explicitly distinguishing left and right context for word embeddings," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 175–180.

[14] "The Carnegie Mellon Pronouncing Dictionary," http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[15] K. Ito, "The LJ Speech Dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.