The DKU Speech Synthesis System for 2019 Blizzard Challenge

Zexin Cai[1]*, Chuxiong Zhang*[1]*, Yaogen Yang*[1]*, Ming Li*[1]

[1]Data Science Research Center, Duke Kunshan University, Kunshan, China

ming.li369@dukekunshan.edu.cn, zc114@duke.edu

Abstract

This paper describes the DKU text-to-speech synthesis system built for the 2019 Blizzard Challenge. The task of this year's challenge is to build a synthetic voice that is similar, expressive and clear as the given data collected from an internet talk show. The DKU speech synthesis system adopts the end-to-end speech synthesis architecture named Tacotron2. First, we analyze the data provided by the organizers and preprocess the data to make it appropriate for text-to-speech synthesis model training. The preprocessing phase includes audio-text aligning, segmentation and manually labeling the pinyin sequences. We pre-train a synthesis model trained with clean Mandarin Chinese speech synthesis dataset and finetune the model using the preprocessed data. In the synthesis phase, we preprocess the texts in evaluation set to obtain the appropriate phoneme sequences for synthesis. After feeding the phoneme sequences into the synthesis system, we use the Griflim algorithm to estimate the phase and convert the output spectrogram to audio. We report our result based on the system performance provided by the organizers.

Index Terms: Tacotron, speech synthesis, transfer learning

1. Introduction

The annual Blizzard Challenge since 2005 aims to explore advance techniques in building corpus-based speech synthesizers on a challenging database. The task 2019-MH1 of Blizzard Challenge 2019 is to deliver an expressive synthetic voice similar to a native Chinese speaker from an internet talk show. Regarding the task, the organizers provide eight hours of speech and the corresponding texts. This is the first year we participate in Blizzard Challenge. We train our synthesizer using the end-to-end neural network framework Tacotron2 [1] under the transfer learning manner.

There are three main approaches for text-to-speech synthesis, which are unit selection, statistical parametric speech synthesis (SPSS) and speech synthesis using end-to-end neural network architecture. The synthesis speech is produced by concatenating natural speech units (segments) when adopting unit selection technique for TTS [2]. It has high quality at waveform. However, the natural variations of speech are lost during automated segmentation of the waveforms, resulting in audible discontinuity and hits in the output [3]. SPSS [4] is based on mathematical approach to generate speech signal and there are many previous Blizzard Challenge partcitipated teams adopted SPSS as their basic framework [5, 6, 7]. SPSS can produce smooth and stable speech. On the other hand, SPSS suffers from distinguishable unnaturalness with buzzy output speech [8]. The TTS synthesis systems using end-to-end neural network architecture reach state-of-the-art performance in recent years [1, 9, 10]. This approach is able to compute the speech directly from graphemes or phonemes. Followed by the Griffin-Lim algorithm [11] or WaveNet [12] vocoder, Tacotron2 [1] is able to yield synthetic speech that approaches the real human speech. However, it is hard to control the speaking style when using the end-to-end speech synthesis architectures. Concerning spontaneous speech, speech synthesis model based on end-to-end framework has better performance comparing to the other two techniques. Therefore, we choose to adopt Tacotron2 for building the text-to-speech synthesis system for task 2019-MH1.

To meet the Blizzard Challenge 2019, our Tacotron2-based DKU speech synthesis system is trained as follows. Firstly, we use our automatic speech recognition (ASR) system to perform force alignment on the audio-text pairs. Base on the likelihood and time boundary information, we remove those audio-text pairs with low likelihood regarding wrong transcription and then segment the texts into sentences. Secondly, we adopt grapheme-to-phoneme technique to obtain the pinyin sequence (phoneme representation in Mandarin Chinese) of each audio-sentence pair. In addition, we fix the pinyin sequence manually to make sure the training pinyin label is correct. Concerning training strategy, we pre-train the Tacotron2 model using a clean Mandarin Chinese database of approximately 9.6 hours. Then the segmental audio-pinyin sequence pairs are fed into the pre-trained Tacotron2 model for fine-tuning. In the synthesis phase, we use a rule-based text normalization method to convert the non-natural language tokens in test set to natural language tokens. Then we break the input text into short sentences and convert the sentences into pinyin sequences. Besides, we use a rule-based method according to the sandhi problem to correct the tone annotations. The predicted linear spectrograms is generated by feeding the pinyin sequences into the Tacotron2 model. Griflim algorithm is used as the vocoder for converting the spectrogram to audio in our approach. Finally the audio sentence-level segmental speech are concatenated into a long audio matching the texts in the provided evaluation set.

The rest of this paper is organized as follows. Section 2 describes the data cleaning method. Section 3 presents the details of end-to-end neural network framework called Tacotron. The text analyse method in evaluation phase and subjective results are shown in Section 4. Finally, the conclusions is given in Section 5.

2. Data Processing

2.1. Raw Data

The provided dataset of 2019 Blizzard Challenge contains 480 audio-text pairs. The audios are compressed in MP3 format. The duration of every audio is 60 seconds. They are spontaneous speech of one male speaker talking about all kinds of ideas and thoughts on modern society. Each audio has a nicely organized transcript that is not exactly match the content in the corresponding audio since the transcript was refined by removing some speaking characteristics in spontaneous speech, like word repetitions, filled pauses, modal particles, etc.

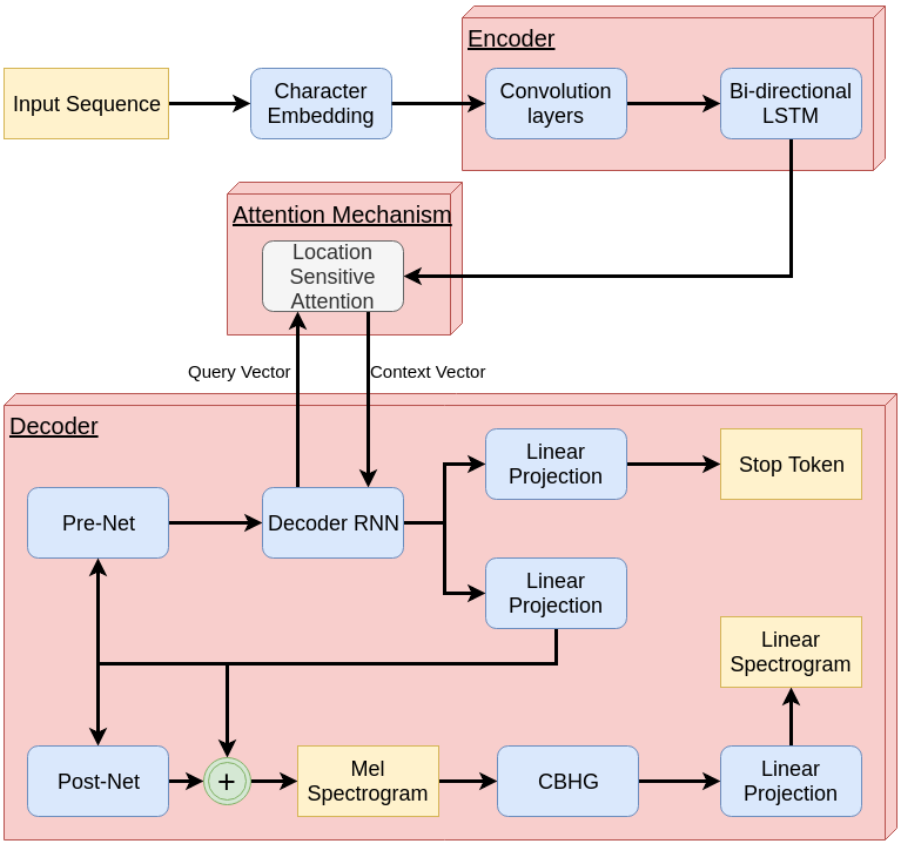

Figure 1: *Architecture of Tacotron*

2.2. Align and Segment

We convert the audio from MP3 format to waveform audio file format sampling in 16kHz, which is more appropriate for acoustic feature extraction. Then we perform force alignment using our automatic speech recognition (ASR) system between the text transcripts and audios to obtain the time-boundary information. 18 utterances is removed since the alignment log likelihood is extremely low and we found out that those transcripts do not match the content in the corresponding audios at all. The remaining 462 utterances are segmented into short audio-sentence pairs base on the time-boundary information and the punctuations in transcript. Furthermore, we remove those segments with duration longer than 12.5 seconds. At last, the total duration of the processed data set is 6.42 hours. The transcript labeled with Mandarin characters are converted into pinyin(phoneme) sequence using pypinyin toolkit [13]. This will be introduced in section 2.3. Finally, the pinyin sequence of the selected utterances, especially the tone annotations, are fixed by native speakers to match the speech content in the audio.

2.3. Text Processing for Mandarin Chinese

We could not intuitively get the pronunciation of the Chineses text in written form unless we convert it into phoneme sequences. The phoneme sequence is more suitable and stable for Mandarin Chinese TTS system. The mapping and alignment between the phoneme sequence and acoustic feature sequence are more reliable. Therefore, we employ the phoneme sequence as the input of our system. The Mandarin Chinese phoneme, called pinyin, can be divided into three parts: consonant, vowel and tone respectively. For example, the phoneme ”shang1” can be separated into “sh”, “ang” and “1” where “sh” is the consonant, “ang” is the vowel and “1” denotes the tone. Usually, we need blank spaces to perform disambiguation between characters when converting the text into the pinyin sequence. For example, the corresponding phoneme sequence of the transcript “今天天气不错” (It is a nice weather today) is “jin1 tian1 tian1 qi4 bu2 cuo4”.

3. System Description

The Tacotron2 system consists of two components, (1) a recurrent sequence-to-sequence feature prediction network with an attention module which predicts a sequence of linear spectrogram frames from a phoneme sequence, and (2) the vocoder module using Griffin-Lim algorithm to convert the spectrogram frames to audio. The detail of the first component will be presented in the following subsections.

3.1. Acoustic Feature Representation

In our approach, we choose the linear-frequency spectrograms as a high-level acoustic representation to bridge the two components. Short Time Fourier Transform [14] (STFT) is used to analyze the time-domain audio signal in the frequency domain. The linear-frequency spectrograms can embody more acoustic

information yet it is much harder for neural networks to imitate such acoustic representation. It is more appropriate to introduce a low-level acoustic representation as a temporary output. The mel-frequency spectrogram is considered as a good acoustic feature with lower dimensionality. The mel-frequency spectrum is obtained by applying a non-linear mel scale transform on the frequency axis of STFT. Then a particular number of mel-filters are designed to summarize the frequency content according to the human auditory system.

3.2. Model Architecture

Tacotron is designed as an attention-based sequence-to-sequence model involving three modules: the encoder, decoder and attention mechanism. The architecture of Tacotron system is shown in Figure 1.

3.2.1. encoder

The encoder is composed of several convolutional layers followed by bi-directional long short-term memory (BLSTM) [15] layers to obtain the interactive and long-term correlations between elements in the sequence. This structure produces hidden representation from character or phoneme embedding sequence. The hidden representation is then fed into the attention mechanism to obtain a fix-length context vector. The input phonemes are represented using a learned 512-dimensional character embedding. Some necessary batch-normalization and dropout layers are also involved to prevent it from over-fitting.

3.2.2. attention mechanism

The attention mechanism used in encoder-decoder architecture is to allow the decoder to refer to different parts of the source encoding sequence at each decoding step. This mechanism has widely been used in many sequential machine learning fields including Speech Recognition [16], Machine Translation [17], etc. However, unlike machine translation, speech synthesis is a streamlining task and the attention is undoubtedly moving forward as time-related decoding step increases. In this case, a location-sensitive attention [18] is applied to the output of the encoder. The location-sensitive attention extends the additive attention mechanism [19] to employ cumulative attention weights from previous decoder time step as an additional feature. In other words, this mechanism enables the model to move forward consistently through the input sequence. Additionally, it mitigates potential failure cases like repeated subsequences or ignored subsequences. The attention mechanism in Tacotron summarizes the full sequence from the encoder as a context vector consumed by the decoder to predict the acoustic feature representation at each decoder time step.

3.2.3. decoder

The decoder is an autoregressive recurrent neural network. At each decoding time step, it predicts a mel-spectrogram frame given the previous decoder output and a context vector generated from the attention mechanism. Tacotron use two fully connected layers as the pre-net applied to the previous output and followed by a stack of uni-directional LSTM to maintain the long-term dependencies. The context vector and LSTM output is concatenated and passed through two separately projection layers for mel-spectrogram prediction and stop token prediction respectively. A post-net consisting of several convolution layers is applied to perform residual reconstruction of mel-spectrogram prediction. Finally, the linear-spectrogram prediction is generated by feeding the residual mel-spectrogram prediction into a complicated network named CBHG and a linear projection layer. The CBHG network consists of convolution layers, fully connected layers and GRU [20] recurrent layers.

3.3. Model Setup

We use librosa [21] Python package to perform Short Time Fourier Transform [14] (STFT) on the audio to obtain the acoustic parameters. Our model aims to predict the normalized energy of STFT. The number of mel-spectrogram channels is set to 80 while the dimensionality of the linear channels is 1024. Since our model is trained with audio sampling in 16 kHz, the window size and hop size is set to 800 and 200 respectively. The minimum frequency in extracting acoustic feature is 55Hz. All other network hyper-parameters remain unchanged. We add the tone annotation (represented by Arabic number 1 to 5) in the embedding character dictionary to allow our model to mimic the speaking style produced by different tone.

3.4. Model Training Procedure

We pre-train our system using the BZSYP database [22]. The speaker is a Chinese female in age around 20. The sample rate of the audio is 48 kHz. All the texts, phoneme sequence and prosodic boundaries are well-labeled in this database. We downsample the audio to 16 kHz for training a matching model for adaptation. The BZSYP database has 10.38 hours of audio data. The phoneme sequence and texts have carefully been rectified with less than 2% and 0.2% error rate as described authoritatively. The training steps for the pre-trained model is 200000.

Then we use the processed data (as described in Section 2) to further fine-tuning the model. In this case, the training steps for adaptation is set to 185000.

4. Synthesizer and Results

4.1. Text Analysis

There are 2546 texts in the evaluation data provided by organizers. The evaluation set contains the transcripts of talks of the same speaker, transcripts of the daily spoken speech, transcripts in written language, poems, etc. However, the text is too rough for our synthesis system. A front-end processing module is used to obtain the correct pinyin sequences of the provided text for our synthesizer.

4.1.1. text normalization

There are non-natural language tokens in the given texts. We use a rule-based method to convert those tokens into natural language tokens. Besides, this method could eliminate the ambiguity caused by non-character Chinese strings of various formats. Thus the system can synthesize the correct pronunciations of these non-natural tokens. For example, a rule-based text normalization method includes such thing as identifying and expanding abbreviations, recognizing and analyzing expressions such as dates, fractions, and amounts of money, and so on. [14]

4.1.2. text segment and grapheme-to-phoneme conversion

We segment the text into sentences with a length of fewer than 30 characters. Then Pypinyin[13] is used to convert the character sequence into pinyin sequence. Generally, the tone sandhi problem in Mandarin strongly affects the performance of the synthesis system and results in unnatural synthetic speech[23].

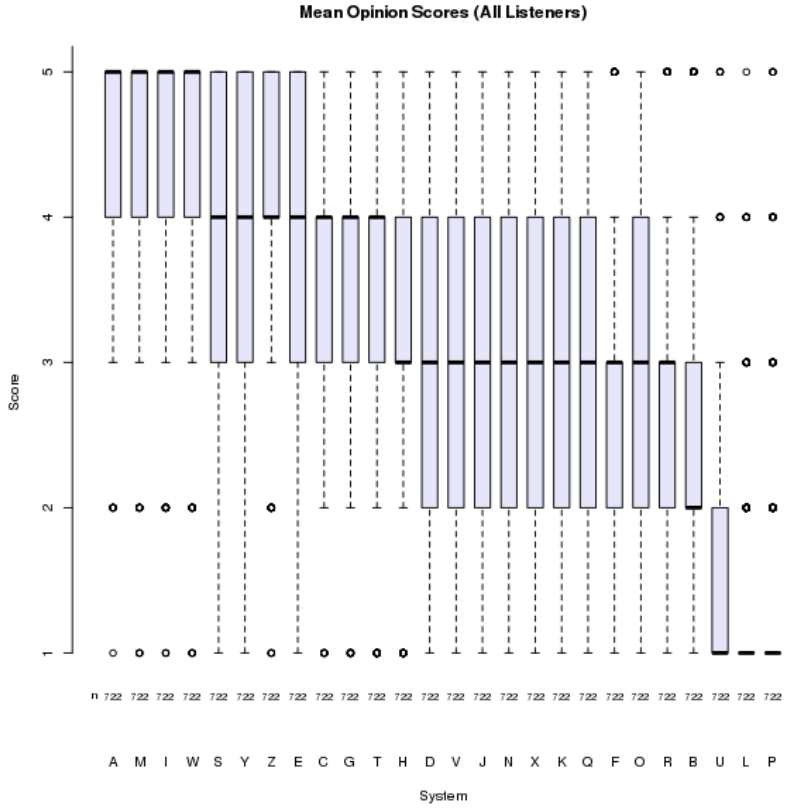

Figure 2: *Mean Opinion Scores on naturalness (all listeners)*

There are five phonologically distinctive tones in Mandarin, represented by tone 1 to 5. The sandhi problem can be described with rules including the tone of the former pinyin become tone 2 when two consecutive pinyins with tone 3 pronounce. In order to deal with the problem of contextual variable pronunciations, we develop a rule-based algorithm to estimate the accurate tone according to the sandhi rule in Mandarin Chinese. After synthesizing the speech of all sentences, the short audios in sentence-level are reconstructed to long audios in text-level matching the provided evaluation text set, and short silence is added between sentences concerning the punctuation mark.

4.2. Subjective Evaluation Results

The evaluation is conducted by paid listeners, online volunteers and speech experts. Our system is labeled as "D" in the challenge. Whereas system "A" is the natural speech and system "B" is the synthetic voice built by merlin[24]. System "C" to "Z" are all participated teams in 2019 Blizzard Challenge.

The subjective evaluation is conducted according to three aspects:

- **Naturalness**, this evaluates how natural that the synthetic speech sound comparing to the natural speech. Listeners evaluate the given speech in scale 1-5, the higher the value, the better the naturalness.
- **Similarity**, this evaluates how much the synthetic voice sounds like the target voice. Listeners evaluate the given synthetic voice in scale 1-5.
- **Intelligibility**, this evaluates how much can people understand the meaning of the synthetic speech. In this task, listeners are asked to write down what they hear from the speech. And the intelligibility is evaluated by the pinyin error rate, with and without tone respectively.

Figure 2 presents the overall MOS concerning the naturalness of all systems. Figure 4 shows the pinyin error rate without tone evaluated by paid listeners of all systems. Figure 5 shows the pinyin error rate with tone evaluated by paid listeners of all systems. And figure 3 shows the overall MOS concerning the similarity of all systems. As shown in the figures, our system performs moderately among all participated teams.

5. Conclusions

In this paper, we present our DKU synthesis system based on Tacotron end-to-end neural network architecture. Our system is trained under the pre-training manner for 2019 Blizzard Challenge. The subjective evaluation of our team is in the medium level of all participated systems.

However, we could improve our system in the following ways. The front-end module that converts the text with non-natural symbols to correct phoneme sequence should be improved. This module includes the text normalization module, grapheme-to-phoneme conversion module and the sandhi process module. Especially in grapheme-to-phoneme conversion module, we do not address the homograph problem called polyphone disambiguation in Mandarin Chinese. We did develop a polyphone disambiguation system for Blizzard Challenge. But the performance of it is not good since there exists mismatch

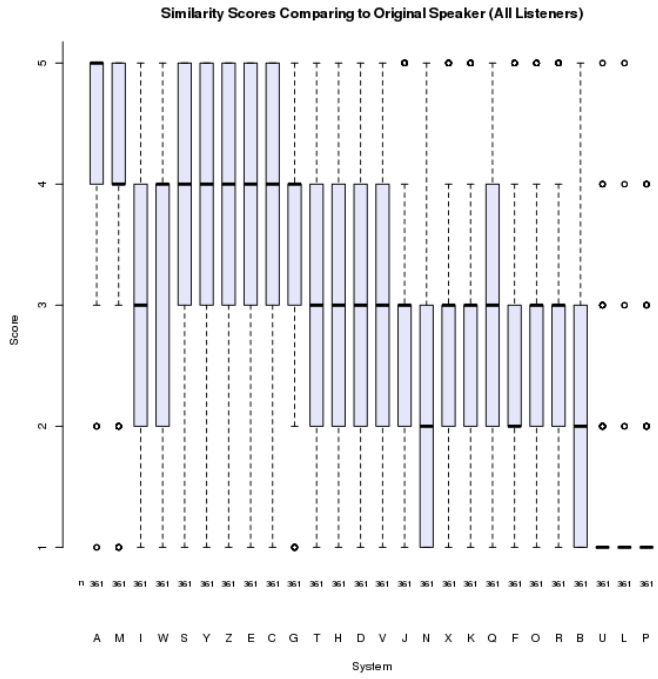

Figure 3: *Mean Opinion Scores on similarity (all listeners)*

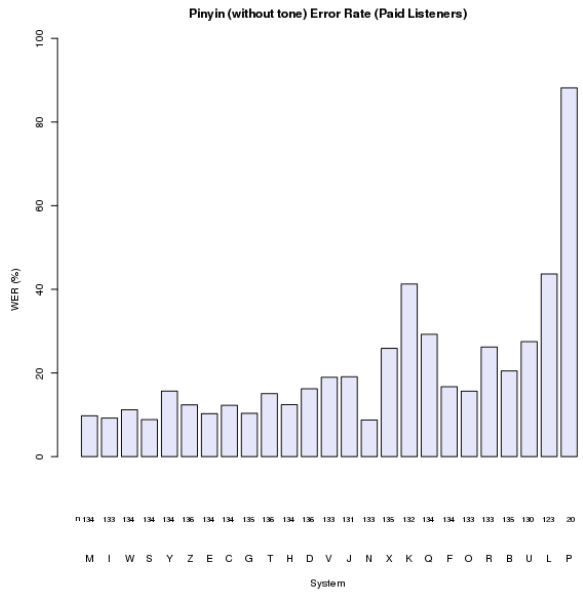

Figure 4: *Pinyin error rate without tone (paid listeners)*

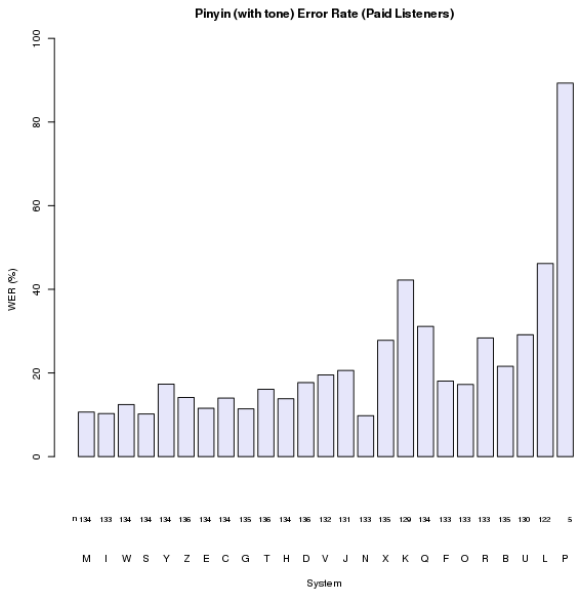

Figure 5: *Pinyin error rate with tone (paid listeners)*

of data materials between the training data of the polyphone disambiguation system and the evaluation text set of the 2019 BLizzard Challenge.

Besides, the training steps for finetuning the pre-trained model is not enough for 2019 Blizzard Challenge because of the variability of the training speech concerning speaking speed, prosody and so on. A prosody estimation module followed by the implicit prosody parameters extraction module could also improve our system refer to the similarity and naturalness. Finally, a better vocoder, like Wavenet[12] or Waveglow[25] could be used to convert the mel-frequency spectrogram to speech with better quality.

6. References

[1] 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 *ICASSP*, 2018, pp. 4779–4783.

[2] A. J. Hunt and A. W. Black, "Unit Selection in A Concatenative Speech Synthesis System Using A Large Speech Database," in *ICASSP*, 1996, pp. 373–376 vol. 1.

[3] H. Mullah, "A Comparative Study of Different Text-to-Speech Synthesis Techniques," *International Journal of Scientific & Engineering Research*, vol. 6, pp. 287–292, 07 2015.

[4] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech Parameter Generation Algorithms for HMM-based Speech Synthesis," in *ICASSP*, vol. 3. IEEE, 2000, pp. 1315–1318.

[5] S. Rallabandi, P. Baljekar, P. Wu, E. Spiliopoulou, and A. W. Black, "Submission From CMU for Blizzard Challenge 2018."

[6] L.-J. Liu, C. Ding, Y. Jiang, M. Zhou, and S. Wei, "The IFLYTEK System for Blizzard Challenge 2017," in *The Blizzard Challenge 2017 Workshop, Stockholm*, 2017.

[7] Z.-H. Ling, Y.-J. Wu, Y.-P. Wang, L. Qin, and R.-H. Wang, "USTC System for Blizzard Challenge 2006 an Improved HMM-based Speech Synthesis Method," in *Blizzard Challenge Workshop*, vol. 4, 2006.

[8] K. Kuligowska, P. Kisielewicz, and A. Włodarz, "Speech Synthesis Systems: Disadvantages and Limitations," *International Journal of Engineering and Technology(UAE)*, vol. 7, pp. 234–239, 01 2018.

[9] 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," *CoRR*, vol. abs/1710.07654, 2017.

[10] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end Speech Synthesis," 2017.

[11] N. Perraudin, P. Balazs, and P. L. Søndergaard, "A Fast Griffin-lim Algorithm," in *WASPAA*, 2013, pp. 1–4.

[12] 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," in *ISCA SSW*, 2016, p. 125.

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

[14] S. Nawab and T. Q. and, "Signal Reconstruction from Short-time Fourier Transform Magnitude," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, no. 4, pp. 986–998, 1983.

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition," in *ICASSP*, 2016, pp. 4960–4964.

[17] M.-T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," *arXiv preprint arXiv:1508.04025*, 2015.

[18] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based Models for Speech Recognition," in *Advances in Neural Information Processing Systems*, 2015, pp. 577–585.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv preprint arXiv:1409.0473*, 2014.

[20] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[21] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and Music Signal Analysis in Python," in *SciPy*, 2015, pp. 18–25.

[22] http://www.data-baker.com/open_source.html.

[23] J. Zhang and Y. Lai, "Testing The Role of Phonetic Knowledge in Mandarin Tone Sandhi," *Phonology*, vol. 27, no. 1, p. 153–201, 2010.

[24] Z. Wu, O. Watts, and S. King, "Merlin: An Open Source Neural Network Speech Synthesis System." in *SSW*, 2016, pp. 202–207.

[25] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A Flow-based Generative Network for Speech Synthesis," in *ICASSP*. IEEE, 2019, pp. 3617–3621.