IMPLEMENTING PROSODIC PHRASING IN CHINESE END-TO-END SPEECH SYNTHESIS

Yanfeng Lu[1], *Minghui Dong*[1], *Ying Chen*[2]

[1]Institute for Infocomm Research, A*STAR, Singapore
[2]Nanjing University of Science and Technology, Nanjing, China
{luyf,mhdong}@i2r.a-star.edu.sg, ychen@njust.edu.cn

ABSTRACT

Text-to-Speech (TTS) systems have been evolving rapidly in recent years. With the great modelling power of deep neural networks, researchers have achieved end-to-end conversion from raw text to speech. It has been shown by various research projects that end-to-end TTS systems are able to generate speech that sounds akin to human voice for English and other languages. However, for languages like Chinese, there are two problems to deal with. Firstly, due to the large character set, a small input set comparable to the English character set is needed for the end-to-end solution. Secondly, there are serious prosodic phrasing mistakes when the end-to-end method is applied to Chinese. In this paper, we will propose a solution for an end-to-end Chinese TTS system on the basis of Tacotron 2 and Wavenet vocoder. We will then add extra contextual information to improve the performance of prosodic phrasing. Our experiments have demonstrated the effectiveness of this proposal.

Index Terms— Chinese speech synthesis, Tacotron 2, Wavenet vocoder, end-to-end TTS, prosodic phrasing

1. INTRODUCTION

Speech synthesis (or TTS) is a process to convert arbitrary texts into speech signals. In the past few decades, the two most dominant types of traditional methods are concatenation methods and statistical parametric methods. The concatenation methods [1][2] rely on a big speech database, and generate target speech signals by concatenating short speech segments. The statistical parametric methods, either HMM-based [3][4] or DNN-based [5][6][7][8], build acoustic models for the language and generate a sequence of acoustic parameters, which are then converted into speech signals by a vocoder. Both methods need a text analysis process to convert textual information into linguistic features, in order to predict the acoustic parameters and generate the speech signals. The text analysis component of a speech synthesis system contains many steps, such as part-of-speech tagging, grapheme-to-phoneme conversion, parsing, prosodic phrasing, etc. The acoustic modeling process needs to predict a set of parameters, such as duration, fundamental frequency and spectrogram. With so many steps in the speech synthesis process, a large amount of resources and effort is required to implement a working system.

In the past few years, thanks to the development of deep neural networks, the end-to-end methods were proposed to generate speech directly from raw text. When the seq2seq model with attention was first introduced for speech synthesis, a rough alignment was needed for the method to work [9]. A limited success was achieved on synthesizing short Chinese sentences. Char2Wav [10] also used encoder-decoder model with attention to predict vocoder parameters from raw text. A neural vocoder was applied to generate speech signals. Later, Tacotron [11] directly predicted raw spectrogram from raw text. The raw spectrogram was then converted into speech with the Griffin-Lim vocoder [12]. Tacotran 2 [13] further improved the speech synthesis process by simplifing the end-to-end process and using Wavenet [14] as its vocoder. Tacotron 2 has shown to generate high quality speech for English speech synthesis.

There were also some efforts on prosody controls for end-to-end methods. In order to generate natural expression, Tacotron was extended to learn a latent space of prosody [15]. The extended system was not only able to generate correct prosody for the text matching the reference signal, but also for the text different from the reference. Another extension to Tacotron was to train global style tokens without explicit labels [16]. The trained embeddings were able to model a large range of acoustic expressiveness. In a different study, an emotional end-to-end system was built by utilizing context vector and residual connection[17]. It was able to generate speech with expected prosody when emotion labels are given. In all the above studies, the control parameters act as a global condition in generating the whole utterance.

Our work of Chinese end-to-end speech synthesis is based on Tacotron 2. We first implemented the method for the synthesis of Chinese Mandarin speech. As Chinese is not an alphabet language, its character set is very large. To make the method work, we had to use pronunciation sequence as input sequence in our implementation. We generated Chinese speech with the implemented system, and found that the prosodic breaks were not properly realized in the generated speech. There are many cases of unsmooth part where syllables are not correctly grouped as word, prosodic word or prosodic phrase. The reason for the failure of prosodic phrasing is largely the fact that the speech database is not big enough to cover the cases of prosodic phrasing. Therefore, we add some extra linguistic information to the end-to-end system so as to achieve high quality natural speech with correct prosodic phrasing. The extra information serves as a local condition to control the prosody.

The rest of the paper is organized as follows: Section 2 describes our baseline Chinese end-to-end TTS system. It's essentially an adaptation of Tacotron 2. In Section 3 we propose our text enhancement methods, to improve the prosodic phrasing of the speech generated from the TTS system. Section 4 presents the experiments we conducted on our several versions of Chinese TTS systems and especially the comparison of the effectiveness of different approaches. At last, we conclude our study in Section 5.

2. BASELINE END-TO-END CHINESE TTS SYSTEM

Firstly, we build an end-to-end Chinese TTS system, based on which we further try to resolve the prosodic phrasing issue.

2.1. Network Architecture

Our Chinese TTS system employs the general architecture of the original Tacotron 2 [13], except that we first convert the text input into a sequence of phones with tones.

As Figure 1 shows, the phone sequence of the input text is then passed to an encoder. The encoder (color coded with light blue) consists of a phone embedding layer, 3 convolution layers and a bi-directional LSTM layer. Generally the encoder encodes the phone sequence with hidden feature representation. The hidden features then go through a location sensitive attention module to generate attention context vector.

The decoder (color coded with pink) is an autoregressive recurrent neural network. It consists of a 2-layer pre-net, 2 LSTM layers, a linear projection layer and a 5-convolution-layer post-net. Mel spectrogram features are generated from the decoder frame by frame autoregressively. The attention context vector is concatenated at the LSTM layer of the decoder.

The generated mel spectrogram is finally input to a Wavenet vocoder [18] to produce the waveform samples. The Wavenet vocoder is speaker-dependent and trained from the same data as mel spectrogram prediction network.

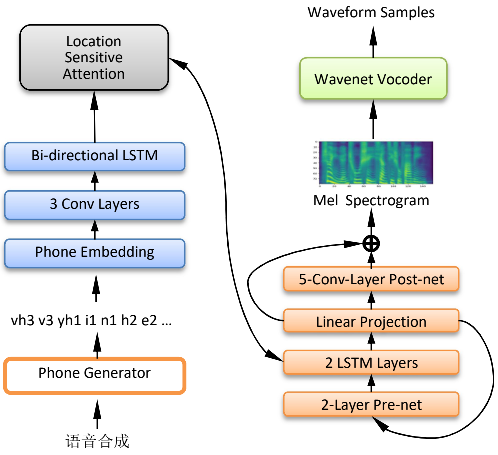

Fig. 1: End-to-end Chinese TTS architecture

2.2. Special Handling of Chinese Text

The original Tacotron 2 was designed for English. It can be easily applied to other languages which use Latin alphabet, as in those cases the input would be the same sequence of Latin characters. We have also successfully built a Malay end-to-end TTS system, with minimal modification of the original architecture.

Chinese has a very different story. Chinese characters are ideograms symbolizing the idea instead of indicating the sound. Compared with the phonetics, the semantics of a language is much more complicated. As a result, there are thousands of Chinese characters. It is not practical to directly use the original Chinese characters as the input character set in the end-to-end system.

In order to reduce the size of the input Chinese character set, a straightforward solution is to use the Pinyin (pronunciation representation) of Chinese characters. Unlike Chinese characters, Pinyin is a much smaller set. There are only about 400 initial and final combinations in Pinyin. However, this is still a much bigger set compared with the Latin character set. And when we combine the tones, the Pinyin set will increase 4-5 folds.

To further shrink the input character set size, we adopt the phone set. Our Chinese phone set contains 43 phones. Combined with the 5 Mandarin tones, our input character set size is 215. In our architecture, the phone sequence corresponding to the input Chinese text is the equivalent of the input English text of the original Tacotron 2. The size of 215 is much closer to the Latin character set size. Moving from Pinyin to phone set is a trivial task in terms of implementation, because there is a direct mapping between them.

By using phone set, we have achieved high quality synthesized speech. However, comparing different Chinese input sets is not the focus of this paper. Here we just use this Chinese TTS system as a baseline, to explore the possible improvements on prosodic phrasing.

3. TEXT ENHANCEMENT WITH EXTRA INFORMATION

3.1. Motivation

Although the original Tacotron 2 was good at rendering high quality speech, we quickly identified some issues in the baseline system. One major issue concerned prosodic phrasing.

Phrasing is a general linguistic phenomenon, but it is more complex in Chinese than other languages. Unlike Western languages, there is no explicit word breaks between Chinese characters in a sentence. Sometimes adjacent characters may be grouped differently to form different words. Which way of grouping makes sense depends upon the context. Besides this word grouping, there are also higher level groupings [19][20]. Words are grouped into prosodic words. Prosodic words are grouped into prosodic phrases.

Phrasing influences the rhythm and smoothness of speech. Through informal evaluation we found that the baseline system performed well in short sentences and sentences from the same dataset as the training data. But we saw phrasing issues happen much more frequently when we tried to synthesize novel sentences with greater length. This indicates that the baseline system can model phrasing to some extent, but has difficulty in generalizing in this respect.

This result is still caused by the high complexity of Chinese phrasing. When humans perform phrasing in Chinese, they rely on syntactic and semantic information to a large extent. While syntactic and semantic information is involved, more data are needed for modelling. However, it is not practical to build very large speech database to cover all the text combinations. In contrast, text data are much easier to acquire, and phrasing model could be built out of text data alone.

Therefore, we propose a text enhancing method on top of the baseline end-to-end TTS architecture. The method keeps the modelling power of the existing architecture, but tries to leverage previous phrasing models and larger text database at the same time. Our goal is to resolve the phrasing issue significantly while keeping the high quality of synthesized speech.

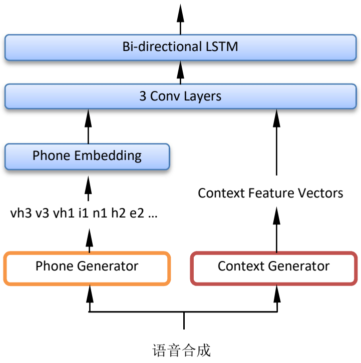

Fig. 2: Architectural adaptation

3.2. Architectural Adaptation

To add in the text enhancement functionality, we need to modify the existing architecture. The extra linguistic features have to be added close to the text input. Once the hidden feature representation is generated in the encoder, we had better not meddle with it. Otherwise the architecture would be modified significantly. That could in turn lower its performance.

As Figure 2 shows, we propose to combine the extra linguistic context features with the phone embedding vector. The embedding is supposed to capture certain features of the character, so it makes sense to concatenate the extra context feature vector with the embedding vector directly. The concatenated vector then goes through 3 convolution layers. And all the rest of the existing architecture keeps intact. In this way we add the extra linguistic information to the existing network.

3.3. Feature Choices

In our Chinese TTS architecture, phone embedding is used as the input to the encoder network. At first we tried to leverage the full context label of the phone, as was used in the HMM-based TTS system [3]. The full context label contains various context features of the phone. After one-hot expansion we got a vector with 336 dimensions. But with such a long vector we could not generate intelligible speech after training the adapted network. This probably had a big impact on the phone embedding. Hereafter we have tried the following features.

3.3.1. Prosodic Word Context

The prosodic word context has a direct influence on the text phrasing and speech rhythm and break. We specifically include the following features in the context:

- position of the current phone in the character
- position of the current character in the word
- position of the current character in the prosodic word
- position of the current word in the prosodic word
- character boundary type (eg. word boundary, prosodic word boundary, etc.)

3.3.2. Whole Sentence Context

Beside the prosodic word context, we also want to find out if larger context helps. The whole sentence context contains all the above prosodic word context features, plus the following:

- position of the current character in the sentence
- position of the current word in the sentence

3.3.3. N-Gram Context

The above context involves non-trivial text analysis and labeled corpora. In comparison, the N-gram context information can be easily calculated from unlabled text corpus. Uni-gram and bi-gram frequencies can be computed and stored in tables beforehand, and then retrieved at run time.

The N-gram context is based on the mutual information of bi-gram. In previous work researchers used mutual information for Chinese word segmentation [21][22]. Mutual information is an indication of the strength of association between the two characters in the bi-gram . The stronger the association, the more likely the bi-gram belongs to the same word. The mutual information of a certain bi-gram can be computed according to the following equations:

$$MI(x, y) = log_2(\frac{p(xy)}{p(x)p(y)}) \tag{1}$$

and

$$log_2(p(g)) = log_2(\frac{freq(g)}{N}) = log_2(freq(g)) - log_2(N), \tag{2}$$

where x and y are the two characters in the bi-gram; $p(.)$ and $freq(.)$ are the probability and frequency of an N-gram, respectively; and N is the total number of a particular type of N-grams in the dataset.

In our N-gram context we include the mutual information of the two bi-grams to which the current character belongs. In the context vector we just include the log frequencies of the 3 uni-grams and the 2 bi-grams contained in the 3 adjacent characters centered at the current character, letting the network handle the coefficients involved.

4. EXPERIMENTS

In the experiments, we want to evaluate the effectiveness of the above three text enhancing approaches in improving the rhythm, break and smoothness of the synthesized speech.

4.1. System Building

In building our baseline and enhanced systems, we leverage two online implementation packages: the Wavenet vocoder package [23] and the Tacotron 2 package [24]. Both packages have to be adapted to handle our own data. We also add the enhancement functionality to the Tacotron 2 package. In addition, we use our own text analyzer to generate full context labels from Chinese text. Scripts for N-gram frequency calculation and other necessary data transformation are written particularly for this study.

4.1.1. Data Preprocessing

The main dataset used in our study contains 22,779 sentences, about 31 hours of transcribed Chinese female speech. We reserve 300 sentences for testing purpose. The original recordings are converted into 16 kHz, 16 bit wave files.

The conversion from Chinese text to phone sequence is done beforehand. We use our text analyzer to generate full context labels from the text. Phone sequences and context features are generated out of the full context labels in turn. Categorical features are one-hot expanded. Finally, our prosodic word context feature vector has 24 dimensions, whereas the whole sentence context feature vector has 30 dimensions.

Before Wavenet vocoder and Tactron 2 model training, the database files are preprocessed. The major part of the preprocessing is to extract mel-spectrogram features. We unify the feature extraction method to use LWS [25], but we keep the original feature normalization method. A trivial transformation is needed in passing the output of Tacotron 2 to Wavenet vocoder.

Beside the main dataset, a much larger text dataset is used to calculate the N-gram frequency tables. This dataset contains about 30GB data crawled from the Internet. The frequency tables are accessed in generating the N-gram context feature vectors. The vectors have 5 dimensions.

4.1.2. Model Training

Wavenet vocoder and Tacotron 2 are trained separately. Through experiments we find that Wavenet with 8 layers is enough to generate high quality speech. Compared to the original network structure, this can increase the training speed 4 times and synthesis speed 10 times. To get satisfactory speech, we normally let the training run about 2,000K steps.

The training of Tacotron 2 is much faster than Wavenet vocoder. Training up to 100K steps is sufficient to generate high quality speech. Usually the alignment becomes good at 20K steps. The enhanced network just takes a little bit longer to train than the baseline version. On two Nvidia TITAN V GPUs, training of Tacotron 2 with our dataset can be completed within one day.

4.2. Method Evaluation

4.2.1. Evaluation Data

We first tested the baseline Chinese TTS system on the reserved sentences from the major dataset. After an informal evaluation on the generated utterances, we found that about 20% of the sentences had prosody phrasing issues. To have a cost effective evaluation, we decided to go beyond the major dataset to include more sentenses with prosodic phrasing issues into the evaluation data.

Therefore, 30 sentences were taken from news and non-fiction articles on the Internet. The only selection criterion was to choose long sentences because long sentences are more challenging in rhythm, break and smoothness. All the sentences contain 17-35 characters. Among them, about 60% utterances generated from the baseline system have prosodic phrasing issues. These sentences can help us evaluate the target systems more efficiently and cost effectively.

The testing sentences went through the synthesis process to generate the target speech. The synthesis process included full context label generation, phone sequence and context feature vector extraction, mel-spectrogram prediction and finally wave generation.

4.2.2. Evaluation Experiment

Corresponding to the 4 systems we built, the following methods were tested in this study.

- Baseline method: no context information is used.
- Ngram method: unigram and bigram spanning 3 characters with the current character in the middle.
- PWord method: context including prosodic word information.
- FullSen method: context including both prosodic word information and sentence level information.

We estimated that the performance of the methods would be in ascending order. Therefore three pairwise comparisons (FullSen vs. PWord, PWord vs. Ngram, Ngram vs. Baseline) were conducted to verify our hypothesis.

Twenty-six native Chinese speakers participated in the perceptual evaluation experiment. 3 AX discrimination tasks were conducted to compare 30 pairs of sentences generated from four synthesis systems by Praat ExperimentMFC [26]. Each pair of the sentences was counter-balanced with AB and BA sequences and auditorily presented in a random order. The participants were requested to choose the better sentence of the two they have heard in terms of rhythm, break and smoothness. The number of the preferred utterances for each system is recorded as the preference score. The experiment was carried out in a sound-attenuated booth, using a Lenovo Thinkpad P40 Yoga laptop computer and Sony MDR-7506 sound monitor headphones.

4.2.3. Evaluation Result

The perceptual evaluation experiment generated three sets of data to compare the 30 pairs of sentences synthesized with different approaches. Paired-samples t-tests were run to test the difference of every two approaches.

The paired test for system preference is shown in Figure 3. The results show that the FullSen method performed better than the PWord method [t(25) = 3.952, p = 0.001], the PWord method performed better than the Ngram method [t(25) = 8.244, $p < 0.001$], and the Ngram method performed better than the baseline method [t(25) = 3.797, p = 0.001]. The results of perceptual evaluation experiment support our prediction of the method comparison.

We also examined the preference of all listeners to each sentence. In the above linstening test, each pair of the utterances was listened 52 times by 26 listeners. If a method is preferred in more than half of the tests, we shall consider the method as a preferred one. Figure 4 shows the statistics of the preference for all 30 sentences. The results show a clear preference of the listeners in terms of the number of sentences. We further examined the generated utterances from the baseline system and found that there were about 18 sentences with prosodic phrasing issues. After the FullSen method was applied, the number of problematic sentences was reduced to 8, showing a great improvement.

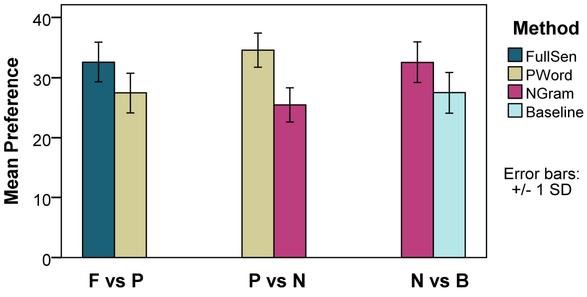

Fig. 3: Paired-samples t-test for the methods

In summary, the three proposed context representations help improve prosodic phrasing. Among them, the FullSen method is the most effective one. Some synthesized sample voices of the different methods are available online[1].

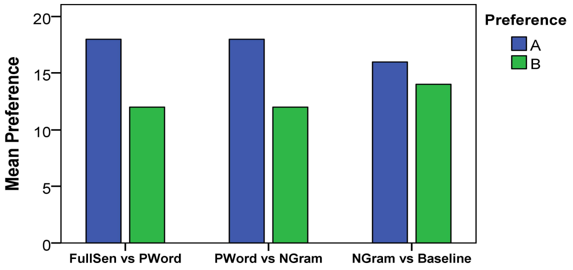

Fig. 4: System preference analysis

5. CONCLUSION

In this study, we built an end-to-end Chinese TTS system, on the basis of Tacotron 2 and Wavenet vocoder. We used phones with tones as our input set, and achieved synthesized speech with high quality. To fix the prosodic phrasing issues, we integrated some traditional contextual information into the end-to-end TTS system. We proposed and evaluated three text enhancement approaches to improve the prosodic phrasing of the generated speech. Our experiments have demonstrated that the whole sentence context performs the best in improving prosodic phrasing.

In this solution, we keep most of the original features of the end-to-end system, but make it more controllable for Chinese prosodic phrasing. This general approach incorporates previous research in the latest end-to-end TTS system and overcomes some of its limitations. This text enhancement method may also be applied to other properties than phrasing.

6. ACKNOWLEDGEMENT

This study is partially supported by the National Science Foundation of China, approval number 61573187, granted to the third author.

[1] http://www.colips.org/mhdong/publication/end2endtts

7. REFERENCES

[1] Andrew J Hunt and Alan W Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*. IEEE, 1996, vol. 1, pp. 373–376.

[2] Yao Qian, Frank K Soong, and Zhi-Jie Yan, "A unified trajectory tiling approach to high quality speech rendering," *IEEE transactions on audio, speech, and language processing*, vol. 21, no. 2, pp. 280–290, 2013.

[3] Heiga Zen, Takashi Nose, Junichi Yamagishi, Shinji Sako, Takashi Masuko, Alan W Black, and Keiichi Tokuda, "The hmm-based speech synthesis system (hts) version 2.0.," in *SSW*, 2007, pp. 294–299.

[4] Heiga Zen, Keiichi Tokuda, and Alan W Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[5] Zhen-Hua Ling, Li Deng, and Dong Yu, "Modeling spectral envelopes using restricted boltzmann machines and deep belief networks for statistical parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2129–2139, 2013.

[6] Shiyin Kang, Xiaojun Qian, and Helen Meng, "Multi-distribution deep belief network for speech synthesis," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8012–8016.

[7] Heiga Zen, Andrew Senior, and Mike Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 7962–7966.

[8] Zhen-Hua Ling, Shi-Yin Kang, Heiga Zen, Andrew Senior, Mike Schuster, Xiao-Jun Qian, Helen M Meng, and Li Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 35–52, 2015.

[9] Wenfu Wang, Shuang Xu, and Bo Xu, "First step towards end-to-end parametric tts synthesis: Generating spectral parameters with neural attention," in *Proceedings Interspeech*, 2016, pp. 2243–2247.

[10] Jose Sotelo, Soroush Mehri, Kundan Kumar, Joao Felipe Santos, Kyle Kastner, Aaron Courville, and Yoshua Bengio, "Char2wav: End-to-end speech synthesis," 2017.

[11] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[12] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[13] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj 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.

[14] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio.," in *SSW*, 2016, p. 125.

[15] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. J. Weiss, R. Clark, and R. A. Saurous, "Towards End-to-End Prosody Transfer for Expressive Speech Synthesis with Tacotron," *ArXiv e-prints*, Mar. 2018.

[16] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous, "Style Tokens: Unsupervised Style Modeling, Control and Transfer in End-to-End Speech Synthesis," *ArXiv e-prints*, Mar. 2018.

[17] Y. Lee, A. Rabiee, and S.-Y. Lee, "Emotional End-to-End Neural Speech Synthesizer," *ArXiv e-prints*, Nov. 2017.

[18] Akira Tamamori, Tomoki Hayashi, Kazuhiro Kobayashi, Kazuya Takeda, and Tomoki Toda, "Speaker-dependent wavenet vocoder," in *Proc. Interspeech*, 2017, vol. 2017, pp. 1118–1122.

[19] Zhengchen Zhang, Fuxiang Wu, Minghui Dong, and Fugen Zhou, "Mandarin prosodic word prediction using dependency relationships," in *International Conference on Asian Language Processing (IALP)*. IEEE, 2015, pp. 173–176.

[20] Zhengchen Zhang, Fuxiang Wu, Chenyu Yang, Minghui Dong, and Fugen Zhou, "Mandarin prosodic phrase prediction based on syntactic trees," in *9th ISCA Speech Synthesis Workshop*, 2016, pp. 160–165.

[21] Richard Sproat and Chilin Shih, "A statistical method for finding word boundaries in chinese text," *Computer Processing of Chinese and Oriental Languages*, vol. 4, no. 4, pp. 336–351, 1990.

[22] Ling-Xiang Tang, Shlomo Geva, Yue Xu, and Andrew Trotman, "Word segmentation for chinese wikipedia using n-gram mutual information," in *Proc. 14th Australasian Document Computing Symposium*, Dec. 2009.

[23] Ryuichi Yamamoto, "Wavenet vocoder," `https://github.com/r9y9/wavenet_vocoder`, 2017 (accessed May 8, 2018).

[24] Rayhane Mamah, "Deepmind's tacotron-2 tensorflow implementation," `https://github.com/Rayhane-mamah/Tacotron-2`, 2018 (accessed June 29, 2018).

[25] Jonathan Le Roux, Hirokazu Kameoka, Nobutaka Ono, and Shigeki Sagayama, "Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency," in *Proc. International Conference on Digital Audio Effects (DAFx)*, Sept. 2010, pp. 397–403.

[26] Paulus Petrus Gerardus Boersma et al., "Praat, a system for doing phonetics by computer," *Glot international*, vol. 5, 2002.