A Prosodic Mandarin Text-to-Speech System Based on Tacotron

Chuxiong Zhang*, Sheng Zhang† and Haibing Zhong†
* Data Science Research Center, Duke Kunshan University, China
† Jiangsu Jinling Science and Technology Group Limited
E-mail: cz134@duke.edu

***Abstract*—The Tacotron performs well in English speech synthesis and successfully aligns two arbitrary sequences from different domain in an automatic way. However, to introduce Tacotron into Mandarin Chinese Text-to-Speech (TTS), a prosody system is needed for generating more natural speech. This paper proposes a practical method to involve the prosodic annotation into Tacotron training for Mandarin Chinese synthesis system. A prosody model predicting the prosodic boundaries from the given text serves as the front-end system in our approach, followed by a Tacotron synthesis system trained with well-labeled TTS database containing the prosodic annotations. Under subjective evaluation in terms of the prosody, results show that the synthesis system performs better by adding the prosodic system as the front-end system for Tacotron.**

I. Introduction

The conversion from normal language text to speech is called speech synthesis, which also known as text-to-speech (TTS). The traditional concatenation synthesis simply concatenating pieces of pre-recorded speech units was the state-of-the-art for many years [1], [2]. The speech generated under this approach is considered intelligibility. However, limited to the natural variations in speech and inflexible automatic technique, the output speech could have audible glitches. Another primary technology for TTS is Statistical Parametric Speech Synthesis (SPSS) [3], [4], [5], [6], [7], [8], [9]. Unlike the concatenation synthesis, the SPSS no longer needs a large database for speech units searching, and have a higher speed in synthesizing a speech. The Merlin toolkit is developed using Deep Neural Network for acoustic modeling method in SPSS [10], [11]. We need a vocoder as the back-end synthesizer converting the estimated parameters into speech signals for SPSS [12], [13], [14]. In recent years, the TTS approaches using end-to-end neural network architecture has dominated the field [15], [16], [17]. The Tacotron [15] sequence-to-sequence model successfully aligns two arbitrary sequences (one is the character sequence, the other is the acoustic representation sequence) in different length, making it possible to compute the speech output directly from graphemes or phonemes. Followed by the Griffin-Lim algorithm [18] or WaveNet [19], Tacotron is able to yield natural speech that approaches the real human speech.

The characteristics of Mandarin Chinese, however, challenge the TTS systems in several ways comparing to English. 1) The text in Mandarin Chinese do not have explicit separator between words; 2) The characters in a given text need to be converted into Latin alphabet representation which is more suitable for TTS system; 3) The homograph problem and off-beating problem is common in Mandarin Chinese. The prosody in Mandarin Chinese mostly refers to the duration of pauses between words or sentences. We lose the prosody information, which can obtain from the context, as we convert the Chinese character into phonemes or Latin alphabet in the Chinese TTS system. Thus the synthetic speech sounds monotonous and less human-like comparing to the English TTS system. While there are approaches under SPSS using Hidden Markov Model (HMM) [20] or extracting the prosody parameters as input to address the prosody problem, two approaches regarding the English end-to-end TTS framework are proposed [21], [22]. However, the two approaches address the prosody problem in a guiding manner that needs a reference utterance or style token as reference code.

This paper proposes a practical method using prosodic annotation to retain the prosody information for end-to-end Mandarin Chinese TTS. Specifically, in the training phase, a prosody labeling network and a Tacotron model are trained. We adopt a sequence-to-sequence neural network for the prosody labeling network to predict the prosodic boundaries for a given text including pauses between words, pauses between phrases and pauses between sentences. The Tacotron is trained with a well-labeled database containing prosodic annotations. In the synthesis phase, a given text would convert to a phoneme sequence with prosodic annotation after feeding to the prosody labeling network. Then we use the prosody-related Tacotron model to synthesize output speech regarding the phoneme sequence. Besides, the subjective evaluation is performed on our proposed method comparing to the baseline Tacotron Mandarin Chinese TTS system. Results show that our proposed method is able to synthesize rhythmed speech with natural prosody.

The rest of this paper is organized as follows. Section 2 briefly describes the Tacotron baseline system. Section 3 presents our extended TTS system with prosodic annotation. The experimental details and results are discussed in Section 4. Section 5 presents the conclusions and future work.

II. Baseline TTS System

The baseline system consists of two components, (1) a recurrent sequence-to-sequence feature prediction network named Tacotron with an attention module which predicts a sequence of linear spectrogram frames from a phoneme

sequence, and (2) the Griffin-Lim algorithm which generates time-domain waveform samples conditioned on the predicted linear spectrogram frames. In the following subsection, we will focus on the first component.

A. Text Processing for Mandarin Chinese

We could not intuitively obtain 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 a Mandarin Chinese TTS system, because 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 Tacotron. The Chinese phoneme, which is also 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"; the "sh" is the consonant in "shang1", the "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 phoneme sequence. For example, if the utterance is "今天天气不错" (It is a nice weather today), the corresponding phoneme sequence for Tacotron input would be "jin1 tian1 tian1 qi4 bu2 cuo4".

B. Acoustic Feature Representation

In this paper, we choose a high-level acoustic representation, the linear-frequency spectrograms, to bridge the two components. Short Time Fourier Transform [23] (STFT) is used to analyze the time-domain audio signal in terms of the frequency domain. High-level representation can embody more acoustic information, but it is much harder to imitate for networks. Hence it is appropriate to introduce a low-level acoustic representation as a temporary output. A mel-frequency spectrogram is considered as a low-level acoustic representation of the short-term power spectrum of a sound. Firstly 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. The phrase information is discarded during acoustic feature representation extraction and is estimated using the Griffin-Lim algorithm when generating the audio samples.

C. Model Architecture

Tacotron is designed as an attention-based sequence-to-sequence model involving three modules, which are the encoder, decoder and attention mechanism. Figure 1 shows the model architecture of Tacotron.

1) Encoder: The encoder is composed of several convolutional layers followed by bi-directional long short-term memory (BLSTM) [24] layers to obtain the interactive and long-term correlations between elements in the sequence. This structure produces hidden representation from character embedding sequence. The hidden representation then feed into the attention mechanism to get a fix-length context

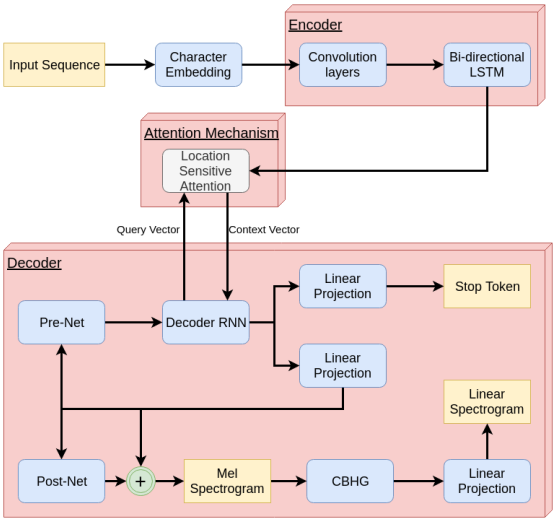

Fig. 1. Architecture of Tacotron

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.

2) Attention mechanism: The attention mechanism used in encoder-decoder architecture is believed to allow the decoder to refer to different parts of the source sequence at each decoder step. It has recently been a trend in deep learning including Speech Recognition [25], Machine Translation [26], etc. However, unlike machine translation, speech synthesis is a streamlining task and the attention is undoubtedly moving forward as time step increases. In this case, a location-sensitive attention [27], which extends the additive attention mechanism [28] to employ cumulative attention weights from previous decoder time step as an additional feature, is applied to the output of the encoder. This mechanism enables the model to move forward consistently through the input sequence and mitigates potential failure cases where some subsequences are repeated or ignored by the decoder. Besides, the attention mechanism summarizes the full sequence from the encoder for the decoder to predict the acoustic feature representation at each decoder time step.

3) Decoder: The decoder is an autoregressive recurrent neural network. At each time step, it predicts a mel-spectrogram frame given the previous output and a context vector generated from the attention mechanism. A pre-net layer consisting of two fully connected layers is applied to the previous output, followed by a stack of uni-directional LSTM to maintain the long-term dependencies. The concatenation of context vector and LSTM output is then passed through two separately projection layers to get the mel-spectrogram prediction and the stop token prediction respectively. A post-net consisting of several convolution layers is applied to the mel-spectrogram prediction to perform residual reconstruction. After feeding the residual mel-spectrogram prediction

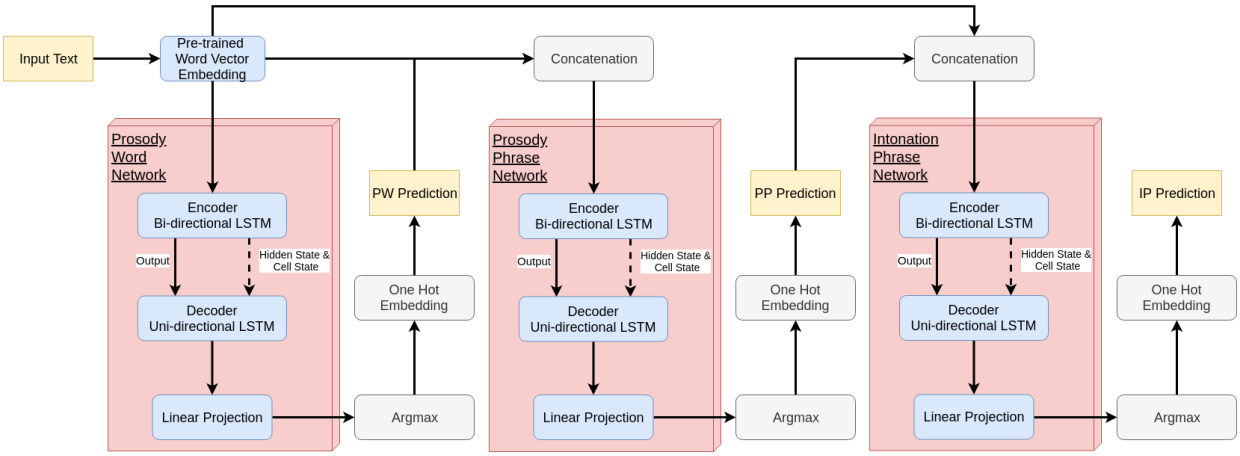

Fig. 2. Architecture of prosody model

into a complicated network named CBHG, which consists of convolution layers, fully connected layers and GRU [29] recurrent layers, a projection layer is used to generate the linear-spectrogram prediction.

III. Extended TTS System with Prosody

The Tacotron model can produce a sequence of linear-spectrogram predictions based on the given phoneme sequence. However, when it is adopted in Mandarin Chinese TTS, Tacotron could not learn any prosody information from the input unless the prosodic annotation is provided. For example, given that "/" represents a short pause within a Mandarin utterance, for an identical sentence "请明天下午到我办公室" (please come to my office tomorrow afternoon), there are plenty ways in reading when it is converted into phoneme sequence "qing3 ming2 tian1 xia4 wu3 dao4 wo3 ban4 gong1 shi4", such as "qing3 ming2 tian1 xia4 wu3 / dao4 wo3 ban4 gong1 shi4", "qing3 ming2 tian1 / xia4 wu3 dao4 wo3 ban4 / gong1 shi4" and etc. Because the phoneme "gong1 shi4" in the later one is the same as another word "公式" (means formula). The former one is more natural compared to the later one because it would not cause ambiguities for comprehension. Therefore, we adopt a prosody labeling system to provide the prosody annotation before the text-to-phoneme procedure.

The prosodic annotation contains three kinds of pause:

- **PW**, a short pause between words where the former word is pronounced with stress.
- **PP**, a medium pause between words or phrases where the tone of the former phrase is pronounced like the end of an utterance but actually not.
- **IP**, the long pause between sentence.

We are not able to distinguish the difference between these three types correctly since the pauses are too short, usually less than 0.2 second. But we can sense the pause and transition in rhythm within a sentence. These prosodic boundaries help disambiguate the comprehension of a sentence, which is significant in the Mandarin Chinese TTS system.

A. Prosody Model

We adopt a prosody neural network architecture to predict prosodic boundaries for a given text. We believe the prosodic annotation is learnable from text since it is determined by the context and word composition of a sentence.

1) Input: Each Chinese character is converted to a unique integer when applying the neural network approaches. The texts in a batch are padding to the max length of all the texts with a symbol that does not appear in the Chinese text, which makes the data more suitable to feed in the neural network without affecting the performance.

2) Prosodic annotation symbol: In our method, we use symbol "@" for **PW** annotation, "%" for **PP** annotation and "&" for **IP** annotation respectively. For a given sentence "今天天气不错" (Today's weather is good), it would become "今天@天气%不错&" after the prosodic annotation prediction.

3) Architecture: The main architecture is shown in Figure 2. We use the prosody prediction approach illustrated in [30] for prosodic annotation labeling. Different to the original method, we use a pre-trained word-to-vector lookup table [31] for text embedding instead of a learnable embedding layer. The word-to-vector table is more suitable in this case regarding its coverage, freshness and accuracy on representing Chinese words.

The prosody model could be divided into three modules according to different annotations. Each module is a binary classification network predicting if a prosodic annotation needs to appear in each time step. For each time step, there is only one prosodic annotation could appear. Therefore the prosody labeling priority follows **IP** > **PP** > **PW**. The input of the annotation prediction module with higher priority is correlated with the module with lower priority through concatenation.

To obtain the long-term dependencies and hidden-level characteristic feature of the Chinese characters in a sentence, we apply a bi-directional LSTM layer as the encoder for every annotation module. Then the encoder output is passed through a decoder containing a single uni-directional LSTM

layer. Finally, a linear projection layer is adopted for binary classification.

B. Tacotron with Prosodic Annotation

To introduce the prosody in Tacotron, we train the Tacotron model with a well-labeled dataset with prosodic annotation. In the synthesis phase, we label a given sentence with prosodic annotation, then convert the text into a phoneme sequence for the Tacotron model to predict the relevant spectrum. This model can imitate the prosodic pause inside the synthetic speech signal according to the input phoneme sequence with prosodic annotations.

IV. Experiments

A. Database

We train our systems on the BZSYP database [32] consisting of 10,000 audio samples. The speaker is a Chinese female in age 20s. The sample rate of the audio is 48 kHz. All the texts, phoneme sequence and prosodic boundaries are well-labeled. We downsample the audio to 16 kHz for training. The duration of each audio is around 4-6 seconds and sum up to 10.38 hours. The phoneme sequence and texts have carefully been rectified with less than 2% and 0.2% error rate as described authoritatively. 5% of the BZSYP database is set aside for testing. We use the Pypinyin [33] Python package to convert the Mandarin Chinese characters into phoneme sequence.

B. Tacotron Model Setup

We use librosa [34] Python package to perform Short Time Fourier Transform [23] (STFT) to extract the acoustic parameter from audio. We extract the normalized energy of STFT as acoustic features which our model aims to predict. 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 samples under 16 kHz, the window size and hop size is set to 1024 and 256 respectively. The minimum frequency in extracting acoustic feature is 95 to help taking off the noise. All other network hyper-parameters remain unchanged to the default value. Two Tacotron models are trained in our work. The first one is trained without prosody annotations, while the other system is trained with prosody annotations and use the phoneme sequence converted from the text that obtains prosodic labeling from the prosody model.

TABLE I
Accuracy for three types of prosodic boundaries

Prosody type	PW	PP	IP
Accuarcy	92.16%	93.27%	99.18%

C. Prosody Model Setup

We pad all the sentence to a fixed length 59, same as the decoder time steps size. The embedding dimension of a word is 200. The number of the hidden unit is 128 in LSTM layers for both the encoder and decoder. The dropout rate is 0.5 and the regulation factor is 0.8. We train our model with Adam optimizer with an initial learning rate of 0.01 and 0.85 decay rate. The prosody model is trained with the text data from BZSYP database.

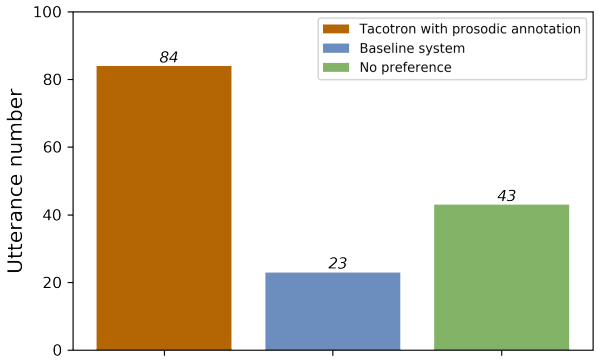

Fig. 3. Preference between two systems

D. Results

As shown in table I. The accuracy of three types of annotation is high enough for prosody prediction. As for subjective evaluation, we asked 10 native Chinese speakers to choose their preference on speech utterances that synthesized from the baseline Tacotron model and the method we proposed according to the prosody and naturalness. Each speaker needs to make their choice on 15 different sentences that randomly selected from the testing set. Results in Figure 3 show that our proposed method outperforms the baseline with 84 choices on the Tacotron with prosodic annotation while 23 on the baseline system.

V. Conclusions and Future Work

This paper provides a method to introduce the prosodic annotation into Tacotron model for generating rhythmed and natural Chinese speech. To address the ambiguity of comprehension in Mandarin Chinese TTS system, a prosody model is trained to predict prosodic boundaries between characters for given text. The subjective evaluation conducted on native speaker shows that our proposed method outperforms the baseline system trained without prosodic annotation. In the future, we will combine the two network architecture, prosody model and Tacotron model, together. Design a prosodic feature extractor that could obtain prosody information automatically from text, where the prosodic feature helps adding pauses between syllables in the synthetic speech to make it more human-like.

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 (ICASSP)*. IEEE, 1996, pp. 373–376.

[2] A. W. Black and P. A. Taylor, "Automatically clustering similar units for unit selection in speech synthesis." pp. 601–604, 1997.
[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 (ICASSP)*, 2013, pp. 7962–7966.
[4] 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)*, 2014, pp. 3829–3833.
[5] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4460–4464.
[6] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks," in *2014 Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014, pp. 1964–1968.
[7] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4470–4474.
[8] Z. Wu and S. King, "Investigating gated recurrent networks for speech synthesis," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5140–5144.
[9] H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak, "Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices," in *2016 Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 2273–2277.
[10] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system." in *2016 ISCA Speech Synthesis Workshop (SSW)*, 2016, pp. 202–207.
[11] S. Ronanki, O. Watts, and S. King, "A hierarchical encoder-decoder model for statistical parametric speech synthesis." in *2017 Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 1133–1137.
[12] 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*, no. 7, pp. 1877–1884, 2016.
[13] H. Kawahara and M. Morise, "Technical foundations of tandem-straight, a speech analysis, modification and synthesis framework," *Sadhana*, no. 5, pp. 713–727, 2011.
[14] F. Espic, C. Valentini-Botinhao, and S. King, "Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis." in *2017 Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 1383–1387.
[15] 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)*, 2018, pp. 4779–4783.
[16] 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.
[17] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," 2017.
[18] N. Perraudin, P. Balazs, and P. L. Søndergaard, "A fast griffin-lim algorithm," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.
[19] 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 *2016 ISCA Speech Synthesis Workshop (SSW)*, 2016, p. 125.
[20] T. Yoshimura, "Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for hmm-based text-to-speech systems," *PhD diss, Nagoya Institute of Technology*, pp. 2347–2350, 2002.
[21] 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 preprint arXiv:1803.09047*, 2018.
[22] 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 preprint arXiv:1803.09017*, 2018.
[23] 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.
[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[25] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.
[26] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.
[27] 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.
[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.
[29] 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.
[30] https://github.com/BoragoCode/AttentionBasedProsodyPrediction.
[31] https://ai.tencent.com/ailab/nlp/embedding.html.
[32] http://www.data-baker.com/open_source.html.
[33] https://github.com/mozillazg/python-pinyin.
[34] 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 *Proceedings of the 14th Python in Science Conference*, 2015, pp. 18–25.