The IMU speech synthesis entry for Blizzard Challenge 2019

Rui Liu, Jingdong Li, Feilong Bao and Guanglai Gao

College of Computer Science, Inner Mongolia University,
Inner Mongolia Key Laboratory of Mongolian Information Processing Technology,
Hohhot 010021, China

liurui_imu@163.com

Abstract

This paper decribes the IMU speech synthesis entry for Blizzard Challenge 2019, where the task was to build a voice from Mandarin audio data. Our system is a typical end-to-end speech synthesis system. The acoustic parameters is modeled by using "Tacotron" model, and the vocoder is using Griffin-Lim algorithm. In the synthesis stage, the task is divided into the following parts: 1) segment long sentence into short sentences by comma; 2) predict interjection labels of each words in short sentences; 3) predict prosodic break labels of each words in short sentences; 4) generate corresponding synthesis speech for each short sentences which enriched by prosodic break labels and interjections; 5) concatenate short sentences into an entire long sentence.

The Blizzard Challenge listening test results show that the proposed system achieves unsatisfactory performance. The problems in the system are also discussed in this paper.

Index Terms: Blizzard Challenge 2019, end-to-end, Tacotron, prosodic phrase break, interjections

1. Introduction

This paper introduces the system submitted to Blizzard Challenge 2019 by Institute for Inner Mongolia University and Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, Hohhot, China. The name of our team is "IMU" and this is our first entry to Blizzard Challenge.

The task of Blizzard Challenge 2019 is to build a voice corpus in Mandarin that would merge up for 8 hours or so. This speech corpus is a collection of the voice of a well-known Chinese man named Zhenyu Luo.

However, since these speech files were not recorded by a professional studio and its non-ideal speech quality, several challenges emerge.

A testing set of sentences are also provided. All participates are asked to submit the corresponding speech files generated by their own model. A large scale subjective evaluation will be conducted.

Recent work on neural text-to-speech (TTS) can be split into two camps. In one camp, statistical parametric methods with deep neural network architectures etc. are used [1, 2]. In the other camp, end-to-end models are used [3, 4, 5]. The recent proposed end-to-end TTS architectures (like Tacotron [3]) can be trained on <text,audio> pairs, eliminating the need for complex sub-systems that needs to be developed and trained separately. In our system, we used the end-to-end Tacotron model [3] to construct experiment architecture.

Besides speech waveform generation, text analysis is another important component of a TTS system. Mandarin is the given language of this year's challenge. In order to retain the expression style of the speakers as much as possible, we used the text data in the training data to model the prosodic phrase break and interjections.

Since our system didn't achieve satisfactory evaluations, we will try to analyze the defects or problems in it. We hope that the problems we faced and solutions we found may provide some useful information for other studies. No external training data was used in our system.

This paper is structured as follows: section 2 describes the data sets and pre-processing steps used for building the voice, while section 3 details the end-to-end speech synthesis techniques, along with the prosodic phrase break prediction and interjection prediction models. The results from the Blizzard Challenge listening tests will be discussed in section 4, with concluding remarks in section 5.

2. Data Processing

2.1. Overview of the data

The data corpus released for the Blizzard Challenge 2019 consists of Mandarin talkshows, which all read by the same male speaker in talkshow style. This database has approximately contains 480 utterances with the total duration of 8 hours. The sampling rate is 24kHz. All files in this database are in mp3 format.

Most of these utterances in the speech files are very rich in emotion with a large number of interjection words. They also include many phrase breaks that are unique to the speaker. The testing transcriptions given by the organizer are picked up from news, long facts, English words, abbreviation, numbers, letters, chinese poetry, etc..

2.2. Preprocessing

Although the quality of these recordings of the training data is standardization certain. the quality of the talkshows may not be ideal for parametric speech synthesis. Moreover, Some sentences in the training data are very long, which puts a burden on the model training. Thus, preprocessing is conducted on the talkshows data as follows:

2.2.1. Segment and Alignment

The provided training data contained mismatches between speech data and text. These mismatches were caused by misreading of a text or phrase or words that do not exist in the text. This will negatively impacts the mapping of parameters of TTS. As is known to all, it is preferable to use a text of fully matched speech data for the training corpus. To overcome this problem, we investigated an automatic construction of a training corpus from talkshows using a speech recognizer. It is expensive to manually obtain a large amount of alignmented texts. Therefore, a speech recognizer is used to recognized text

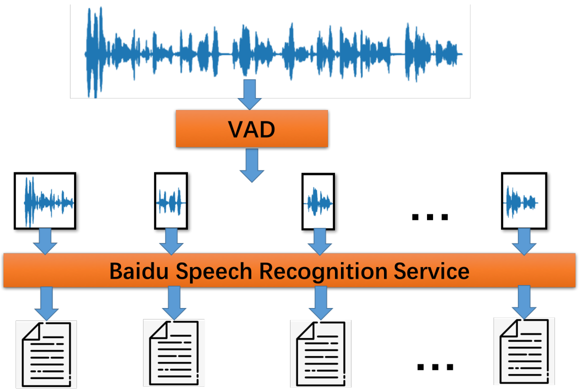

Figure 1: *Segment long sentences and alignment text and its speech by using speech recognizer.*

Table 1: *Examples of true content from a speech file, its text transcription from training data and its recognized text.*

true content	真是凡事都有方法啊
text transcription	真是凡事都有方法
recognized text	真是凡事都有方法啊

of speech data.

Figure 2 shows an overview of the alignment method. First, we segment long speech to some short speech files leverage the voice activity detection tools[1]; Second, all short speech files were recognized to their corresponding text transcriptions with the help of Baidu online speech recognition service[2]. Table.1 shows examples of true content from a speech file, its text transcription from training data and its recognized text. In the examples of Table.1, the words “啊” , which does not exit in a book text, is recorded in a speech data.

In addition, some of the speech files in the training data are too low to hear clearly, so we deleted these files. Through the above pre-processing operation, the size of our training data becomes 8630 utterances with the total duration of 7.68 hours.

2.2.2. *Text Normalization*

The alignmented text data were manually checked. First, because of some error feedbacks by using speech recognizer, the text was manually checked to matched the speech content. Second, due to the mistakes in the voice activity detection, some incomplete or incoherent words in the speech waveform was annotated in the text.

Furthermore, in order to facilitate model training, we replaced other punctuation marks with commas, and converted the digits, English words or letters, etc. in the text into Chinese word representations through regular expressions.

3. System Description

The training framework and synthesis module of our system is shown in Figure.2. We will introduce this model in order.

3.1. Training phase

In training phase, we added some extra prosody information to the end-to-end system, according to the prosodic phrase break labels, so as to achieve high quality natural speech with correct prosodic phrasing. The extra information serves as a local condition to control the prosody.

As shown in the top part of Figure.2. We mapped the break label for each words into a one-hot vector which called as “prosody embeddings (PE)”. The character[3] embeddings (CE) were concatenated with PE and sent to Tacotron model.

Due to the fact that the PE and CE have different time resolutions, below we need to upsample the PE to be the same as the length of the character-level sequence. We first make N copies of PE, where the number of copies is equal to the number of characters in a word. Then we concatenate the PE and original character-level embeddings across all characters in the word into a joint vector which used as the new character-level representation for the Tacotron model.

3.2. Synthesis phase

Before synthesis, the given long sentences are segmented into some short sentences. To synthesize speech, each of the short sentences is enriched by “Interjections prediction” model and “Phrase break prediction” model. In the process of text analysis, we find that there are many interjections, which reflect most of the emotional information in speech, in the data. In order to retain the speaker’s emotional information better, we make interjection prediction for the text to be synthesized. Moveover, phrase break plays an important role in both naturalness and intelligibility of speech [6, 7, 8, 9]. It breaks long utterances into meaningful units of information and makes the speech more understandable. Therefore, identifying the break boundaries of prosodic phrases from the given text is crucial in speech synthesis.

As the bottom half of Figure.2 shows, as a complement of the main Tacotron model, the “Interjections prediction” model and “Phrase break prediction” model are to learn prosody features and essential interjections, belonging to the particular speaker, based on input text.

3.2.1. *Interjections prediction model*

We regard interjection prediction as a sequence labeling task. According to the statistical results of interjections in training data, we select the top eight interjections as prediction targets: 啊, 呐, 吧, 呀, 嘛, 哎, 啦, 呢. The statistical results of these interjections in the training data are shown in the Table.2.

We use the Bidirectional Long Short-Term Memory (BiLSTM) model to predict the interjection for each words. In this module, each words in an input sentence is mapped to a sequence of word embeddings (WE_1,...,WE_t) by pretrain embeddings. Given the input vector sequence (WE_1,...,WE_t), the forward LSTM reads it from left to right, but the backward LSTM reads it in a reverse order.

$$\overrightarrow{h_t} = LSTM(WE_t, \overrightarrow{h_{t-1}}) \quad , \overleftarrow{h_t} = LSTM(WE_t, \overleftarrow{h_{t-1}}) \quad (1)$$

Finally, we use a softmax layer to produce interjection labels. The softmax calculates a normalised probability distribution over all the possible labels of each word:

[1] https://webrtc.org/

[2] https://ai.baidu.com/tech/speech/asr

[3] The “character” mentioned in this paper is pinyin, and we use open source tools, pypinyin(https://pypi.org/project/pypinyin/), to convert the chinese word into its pinyin sequence.

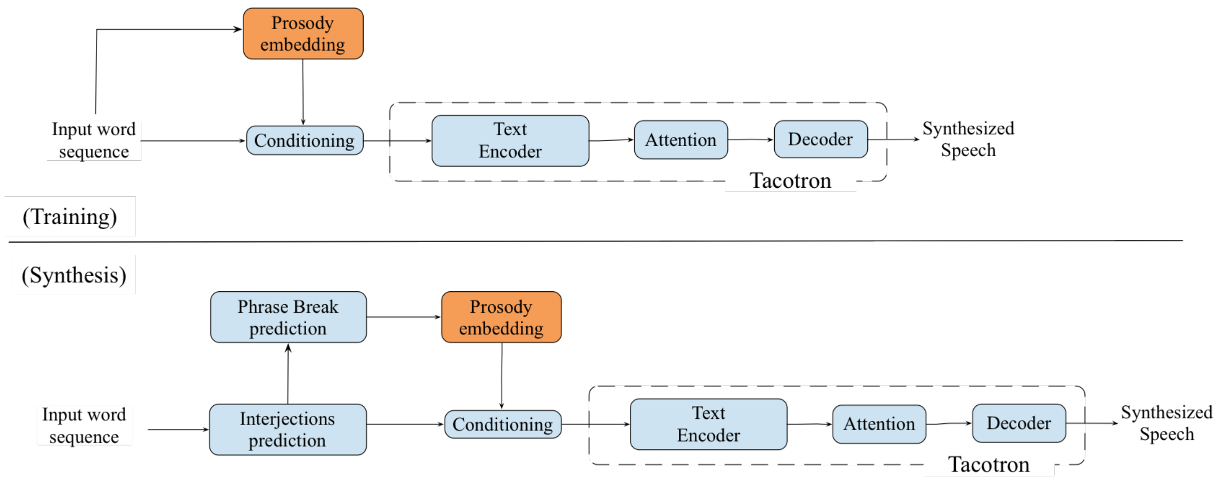

Figure 2: *Training and sythesis phase of IMU speech synthesis system. The prosody embedding is connected to the input of the encoder.*

Table 2: *The statistical results of the top eight interjections in the training data*

啊	3876
呐	328
吧	180
呀	712
嘛	674
哎	726
啦	146
呢	1486

$$P(y_t = k|d_t) = \frac{e^{W_{o,k} d_t}}{\sum_{\widetilde{k} \epsilon K} e^{W_{o,\widetilde{k}} d_t}} \tag{2}$$

where $P(y_t = k|d_t)$ is the probability of the label of the t-th word (y_t) being k, K is the set of all possible labels, and $W_{o,k}$ is the k-th row of output weight matrix W_o.

After the interjection prediction, the original non-interjection text embedded with the predicted interjection can be even richer in emotion. This can make the model get better synthesized speech.

3.2.2. Phrase break prediction model

We extend the original Tacotron model by adding a "***phrase break prediction***" model which takes the word sequence from given text as input, and computes the corresponding break label by using the BiLSTM network. In this model, each words in an input sentence is mapped to a sequence of word embeddings (WE_1,...,WE_t) by a look-up table, which is passed through a BiLSTM. Finally, to produce phrase break label, we use a softmax layer also.

Then the PE and CE (from interjections enriched text) are concatenated, by using upsample method as described in Section 3.1, to create a new character-level encoder input. We assume that this improved input sequence carries more realistic prosody and emotional information.

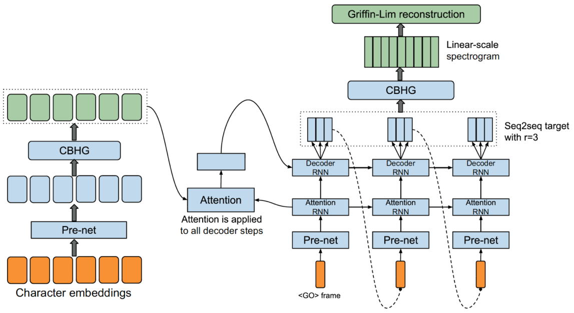

Figure 3: *Detailed network architecture of Tacotron model.*

3.2.3. Tacotron model

For the proposed model, the main task is to generate the speech spectrograms given the input character-level features based on the original Tacotron model.

The Tacotron model is a complicate neural network architecture, as shown in Figure.3. It contains a multi-stage encoder-decoder based on the combination of convolutional neural network (CNN) and recurrent neural network (RNN). The character embeddings of raw text is fed into an encoder which generates attention features. Then the generated features fed in every step of the decoder before generating spectrograms. At last, the generated spectrograms are converted to waveform by the Griffin-Lim method [10]. Our implementation is forked from the TensorFlow implementation from Keith Ito[4], which is faithful to the original Tacotron paper. We first convert the input text to character sequence. Then the character inputs are converted into one-hot vectors. The one-hot vectors then turn into character embeddings and are fed to a pre-net which is a multilayer perceptron. The output is then fed into the CBHG which stands for Convolutional Bank + Highway network + gated recurrent unit (GRU). The output from CBHG is the final encoder representation used by the attention module. Then the generated attention features are fed in a RNN-based decoder. In every step, the decoder generate few frames of spectrograms. The number of output frames is controlled by

[4] https://github.com/keithito/tacotron

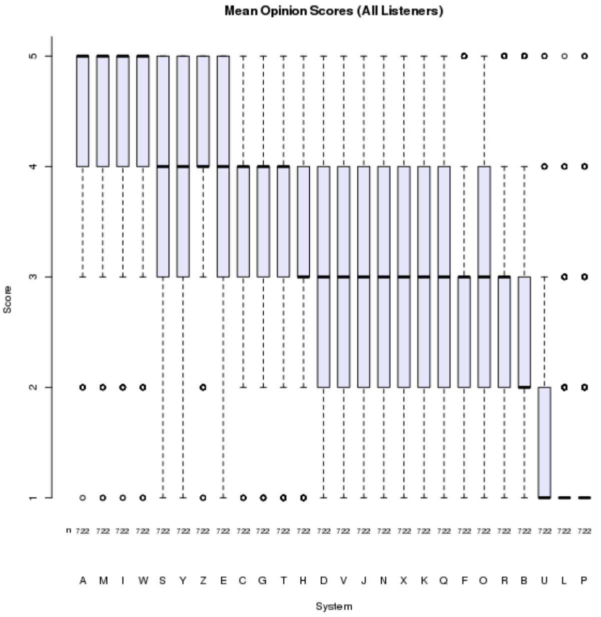

Figure 4: *Naturalness ratings. System O is the proposed system, A is the natural speech, and B is the Merlin benchmark.*

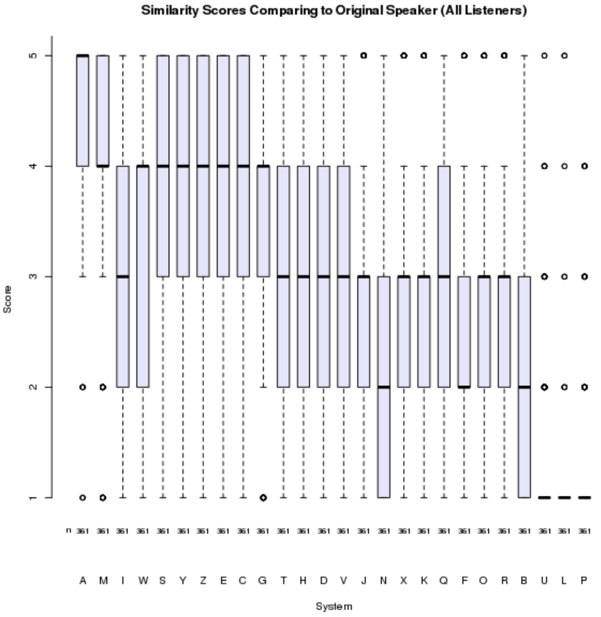

Figure 5: *Speaker similarity ratings. System O is the proposed system, A is the natural speech, and B is the Merlin benchmark.*

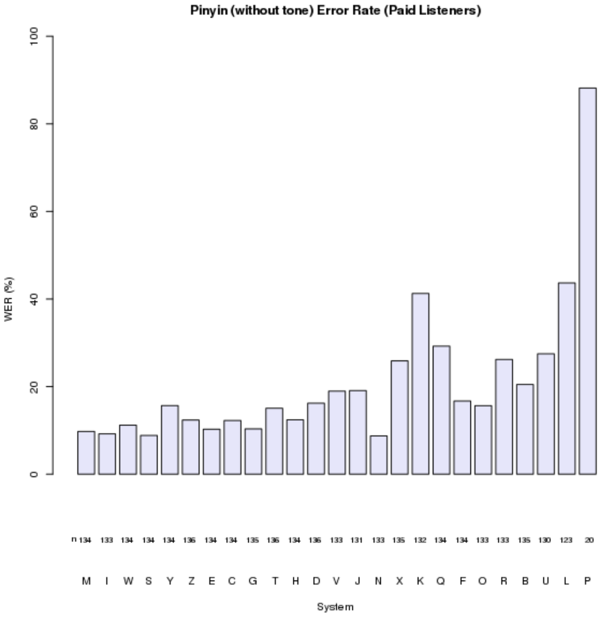

Figure 6: *Pinyin (without tone) error rate. System O is the proposed system, A is the natural speech, and B is the Merlin benchmark.*

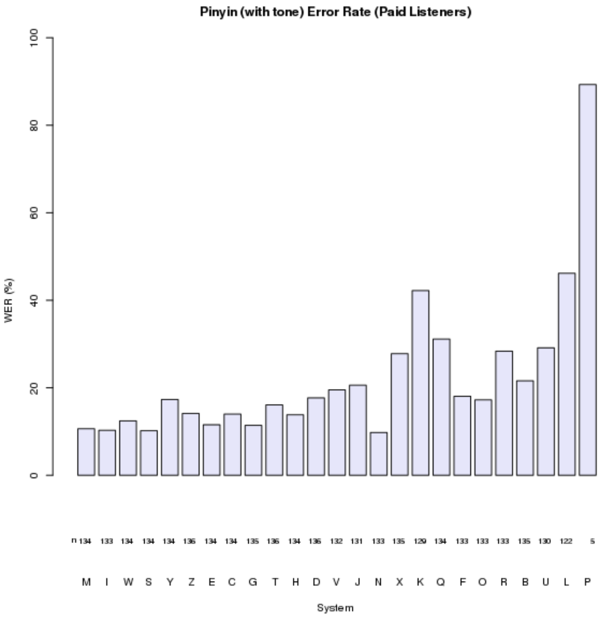

Figure 7: *Pinyin (with tone) error rate. System O is the proposed system, A is the natural speech, and B is the Merlin benchmark.*

a hyperparameter reduction factor (r in Figure.3). At last, the audio is reconstructed by the Griffin-Lim method.

3.2.4. Model setup

The prosody phrase break for training data was labelled by expert annotators by both listening the utterances and reading the transcriptions. In the end of the process each word is labeled as “B” or “NB” depending upon the location of the breaks introduced by the speaker. We set the dimensions of the *PE* to 2. For interjection model, we removed the interjections from the training data firstly, then treated them as separate labels for the previous word in the model training. We set the dimensions of the interjection one-hot vector to 9.

For the above two models, the input tokens fed into BiLSTM network are transformed into embedding features by a look-up table. For Mandarin, we choose “Tencent AI Lab Embedding Corpus for Chinese Words and Phrases” [11]. This corpus provides 200-dimension vector representations, a.k.a. embeddings, for over 8 million Chinese words and phrases, which are pre-trained on large-scale high-quality data.

The LSTM layer size was set to 200 in both direction for all experiments. The size of hidden layer d is 50. We set learning rate as 1.0 and batch size as 64. All parameters were optimised by using AdaDelta algorithm. The training/validation/test split is 8:1:1. At every epoch, we calculate the performance on the training set. We stop training if the effect does not increase seven epoches. The best model on training stage was then used for evaluation on the test set. Finally, the accaury of our interjection model and phrase break model were achieved 97.24%, 90.02% respectively. For Tacotron model, we use Adam optimizer with learning rate decay to train the model and the network were trained 200k steps.

4. Results and Analysis

In this section, we will discuss the evaluation results in detail. Our designated system identification letter is “O”. System A is the natural speech. System B is the Merlin benchmark system and others are participants systems. The subjects who are involoved in the listening test are paid isteners and online volunteers and so on. There are a total of 2546 sentences in the 2019 testing set. The evaluation results of all systems are shown in Figures.4, 5, 6 and 7.

Figure.4 shows the naturalness ratings presented as box plots, where the central solid bar marks the median, the shaded box represents the quartiles, and the extended dashed lines show 1.5 times the quartile range. The most relevant comparisons can be made with the other known synthesis systems, namely system B, which is the Merlin benchmark system. As you can see from the results, our system ranks 19th out of all submitted systems except natural speech. The results show that our proposed system O outperforms the Merlin benchmark B. There is a big gap between our system and the champion system M, and it is also different from other systems in a statistical sense. As far as I know, the champion system M use neural vocoder, and we only use simple Griffin-Lim algorithm, so such obvious gap can be expected.

Speaker similarity scores are presented in Figure.5 with similar box plots. The results show that several systems have shown comparable results to our system, such as J, X, K, F, R, and the proposed system works a bit better of speaker similarity to the Merlin benchmark, having lower similarity than most other systems like M, S, Y, Z, E, C. Four possible reasons may have lead to the relatively low similarity score. First, in the interjection prediction model, we only select a part of interjection (top eight) as the prediction target, missing the rest of the interjections. Moreover, the existing interjection prediction model also has a certain prediction error rate; second, in the prosodic prediction model, we did not address the prosodic hierarchy in detail. Similarly, the existing prosodic prediction also has prediction errors; Third, we believe that the failure to extract the speaker’s characteristics is an important reason for the relatively low similarity score. Fourth, long sentence speeches are generated by concatenating some short sentence, which ignored the importance of pauses between sentences.

As shown in Figure.6 and Figure.7, the pinyin (without tone) error rate and pinyin (with tone) error rate of our system is about 18% on the intelligibility test, ranking 12th in all submitted systems. We found that there were some digits, English words or letters and chinese ancient poetry in the 2019 testing data. Ancient poetry contains a large number of polyphonic words, and we only use open source tools, pypinyin, to convert words to its pinyin (with tone) sequence, not making any extra special treatment on their pronunciation. In addition, English letters are not processed separately. It was one of the factors that degrading the performance in intelligibility.

To sum up, we summarize the problems in the system as follows:

1) The interjection model and prosodic prediction model are too simple and have a lot of room for improvement.

2) The conversion of Chinese words to pinyin only uses open source tools, and some other special cases have no extra processing like digits, English words or letters and chinese ancient poetry, etc.

3) In order to retain the speaker’s style and emotion information, we only conduct some simple analysis at the textual level, without considering the speaker’s acoustic characteristics [12, 13, 14], which is a fatal shortcoming.

4) The final result is obtained by concatenating short sentences together to form long sentences. In the process of concatenating, there must be discontinuity at the connection point, which also has a negative impact on speech quality.

5) Last but not least, our system use Griffin-Lim algorithm as vocoder to convert acoustic feature to waveform. We did not use the popular neural network vocoder to model the speech waveform. Considering of the successful application of WaveNet [5] in TTS, Wavenet-based vocoder was widely used in TTS fileds. By conditioning WaveNet on acoustic features, a WaveNet based neural vocoder is implemented. It can learn the relationship between acoustic features and waveform samples automatically. This neural vocoder is used to replace the conventional vocoder so that waveforms can be generated directly. The quality of synthetic speeches is supposed to be further improved.

5. Conclusions

We introduce the IMU speech synthesis system for Blizzard Challenge 2019. The results of listening test for our system are not good, but we have found many interesting problems that we should have attacked. According to the subjective evaluation results, there is still much room for improvement on our method.

6. Acknowledgements

This research was supports by the China national natural science foundation (No.61563040, No.61773224), Inner Mongolian nature science foundation (No.2016ZD06).

7. References

[1] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *ICASSP 2013 – 38th IEEE International Conference on Acoustics, Speech and Signal Processing, May 26-31, Vancouver, BC, Canada, Proceedings*, 2013, pp. 7962–7966.

[2] Z. Ling, S. Kang, H. Zen, A. Senior, M. Schuster, X. Qian, H. M. Meng, and L. 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.

[3] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous., "Tacotron: Towards end-to-end speech synthesis," in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association, August 20-24 , Stockholm, Sweden, Proceedings*, 2017, pp. 4006–4010.

[4] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio, "Char2wav: End-to-end speech synthesis," 2007.

[5] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Z. Yu, Y. Wang, and R. J. Skerry-Ryan, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," 2017.

[6] Y. Zheng, Y. Li, Z. Wen, X. Ding, and J. Tao, "Improving prosodic boundaries prediction for mandarin speech synthesis by using enhanced embedding feature and model fusion approach," in *INTERSPEECH 2016 – 17th Annual Conference of the International Speech Communication Association, September 8-12, San Francisco, CA, USA, Proceedings*, 2016, pp. 3201–3205.

[7] Y. Zheng, J. Tao, Z. Wen, and Y. Li, "Blstm-crf based end-to-end prosodic boundary prediction with context sensitive embeddings in a text-to-speech front-end," in *INTERSPEECH 2018 – 19th Annual Conference of the International Speech Communication Association, September 2-6, Hyderabad, India, Proceedings*, 2018, pp. 3062–3066.

[8] R. Liu, F. Bao, G. Gao, and W. Wang, "A lstm approach with sub-word embeddings for mongolian phrase break prediction," in *COLING 2018 – 27th International Conference on Computational Linguistics, August 20-26, Santa Fe, New Mexico, USA, Proceedings*, 2018, pp. 2448–2455.

[9] ——, "Improving mongolian phrase break prediction by using syllable and morphological embeddings with bilstm model," in *INTERSPEECH 2018 – 19th Annual Conference of the International Speech Communication Association, September 2-6, Hyderabad, India, Proceedings*, 2018, pp. 57–61.

[10] G. D and L. J. S, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics Speech & Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[11] Y. Song, S. Shi, J. Li, and H. Zhang, "Directional skip-gram: Explicitly distinguishing left and right context for word embeddings," in *NAACL 2018 – 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics, June 1-6, New Orleans, USA, Proceedings*, 2018, pp. 175–180.

[12] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, and I. L. Moreno, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," 2018.

[13] D. Stanton, Y. Wang, and R. Skerry-Ryan, "Predicting expressive speaking style from text in end-to-end speech synthesis," *arXiv preprint arXiv:1808.01410*, 2018.

[14] 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.