The IOA-ThinkIT system for Blizzard Challenge 2019

Ruimin Wang[1,2], *Chunhui Lu*[1,2], *Xiaoyang Hao*[1,2], *Bolin Zhou*[1,2], *Zengqiang Shang*[1,2], *Pengyuan Zhang*[*1,2]

[1]Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, Beijing, P.R.China
[2]University of Chinese Academy of Sciences, Beijing, P.R.China
zhangpengyuan@hccl.ioa.ac.cn

Abstract

This paper presents the IOA-ThinkIT team's text-to-speech system for blizzard challenge 2019. A statistical parametric speech synthesis based system was built with improvements in both front-end text analysis and back-end acoustic modeling. In the front-end, a bidirectional encoder representation from Transformer (BERT) based model was proposed for prosodic boundary prediction. In the back-end, a BLSTM duration model and a multi-speaker acoustic model with speaker code as additional input and variational autoencoder (VAE) residual encoder extension was trained. In acoustic model, speaker code was used to distinguish different speakers while hidden vectors learned from VAE encoder were used to model differences in speech other than speakers and content. Besides, a quantized framework was introduced to model fundamental frequency (F0). Evaluation results showed that though our proposed model (system N) performed not well in MOS and speaker similarity, we got best results on both pingyin (without tone) error rate and pingyin (with tone) error rate among 24 teams.

Index Terms: Blizzard Challenge 2019, statistical parametric speech synthesis, VAE, BERT, quantized F0

1. Introduction

Blizzard Challenge has been held annually since 2005 in order to better understand and compare research techniques in building corpus-based text-to-speech (TTS) systems on the same data. All the participants are asked to take the released speech data, build a synthetic voice and synthesize a prescribed set of test sentences. Then the synthetic sentences from each synthesiser are evaluated through listening tests.

In last fourteen years, there were mainly two mainstream methods used in this challenge, unit selection based waveform concatenation [1, 2, 3] and statistical parametric speech synthesis (SPSS) [4, 5]. Waveform concatenation systems directly use real speech segments from the corpus to concatenate and generate final speech. For SPSS systems, they try to parameterize speech waveform into acoustic features and predict these features by building an acoustic model. Vocoder is used to extract features and reconstruct speech in this process. Comparing these two methods, the advantage of unit selection based systems lies in the speech quality and similarity, which are constrained by the vocoder in SPSS systems, but it demands high quality speech corpus. Recently, some neural vocoders, like WaveNet [6, 7], WaveGlow [8], have been proposed and made up the gap between synthetic speech and natural speech. Meanwhile end-to-end speech synthesis models have also been introduced and generated more natural speech than traditional method.

There was only one task this year. Build a voice from an 8-hour speech data, which was collected from an internet talk show, that was suitable for expressive TTS. All data were from a well-known Chinese male speaker. As these data were not recorded in quiet environments by professional speaker, data quality was not as high as those usually used in TTS systems. Specifically, background noise, speech noise, environmental and channel differences influenced the quality. Considering these factors, we constructed our system based on SPSS and made improvements in both front-end text analysis and back-end acoustic modelling. For front-end, we proposed a BERT [9] based prosodic boundary prediction model. For back-end, except training a phone level duration model and using WORLD [10] as vocoder, a VAE [11] based multi-speaker BLSTM acoustic model was proposed. Besides, a quantized F0 model was adopted.

The rest of the paper is organised as follows. Section 2 introduces our data processing procedure on given data. Section 3 describes the details of our submitted system, followed by the evaluation results in section 4. Finally, the conclusion is given in section 5.

2. Data Preparation

The released data had 480 speech files, each about one minute, and its corresponding transcripts. We first used our grapheme to phoneme (G2P) toolkit generating pinyin according to the original transcripts. As these transcripts were not entirely consistent with the speech, we then manually checked and revised the transcripts and pinyin by listening to the speech. After doing this, all audio files were converted to 16 bits wave files at sampling rate of 16k Hertz and Wiener filtering was used to denoise these speech. Finally, these files were segmented into about 6000 sentences according to the punctuations in the transcripts.

Phone level forced alignments of these sentences were obtained using an automatic speech recognition (ASR) model by Kaldi toolkit[12]. We followed Merlin toolkit[13] to design our system. Each phone was represented by a HTS format full-context label containing features on phone identity, part of speech (POS), prosodic structure and positional information etc. Then these labels were converted to binary and numerical features by a question set. All the features were used as input and normalised by min-max normalisation to the range of [0.01,0.99]. For duration model, the input vectors were 624 dimensions, where 591 dimensions were binary features for categorical linguistic features and 33 dimensions were numerical ones. For acoustic model, 4-dimension features encoding the position of the frame in a phoneme were appended to the input

* Pengyuan Zhang is the corresponding author.

that used in duration prediction.

The speech data was analysed by WORLD with 5 ms frame hop, each of which was represented by a 187-dimension acoustic feature that consists of mel-generalized cepstral (MGC), band aperiodicities (BAP), log F0 (LF0) and their corresponding delta and delta-delta features, besides a binary voiced flag. All the features were normalised by mean and variance normalisation.

3. System Overview

The overview of our system is illustrated in Figure 1. It contains two stages, training and synthesis. In both stages, text analysis including text normalization, word segmentation, POS tagging, prosodic boundary prediction and G2P was firstly conducted to get linguistic features. In training stage, acoustic features were extracted from waveform speech by vocoder, and phone durations were acquired by doing force alignment. Then the extracted linguistic features and acoustic features were used as input and target respectively to train the frame level acoustic model. Similarly, a phone level duration model was also trained in this stage. When it came to synthesis stage, text analysis results of test text were feed into trained duration model and acoustic model sequentially to get acoustic feature sequence. Final speech was then reconstructed by vocoder. The detailed descriptions of each model are as follows.

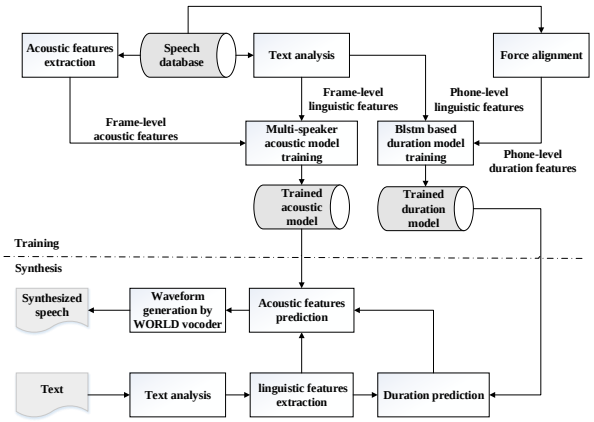

Figure 1: *The overview of our system*

3.1. Prosodic boundary prediction model

In Chinese TTS systems, a hierarchical prosodic structure, including prosodic word (PW), prosodic phrase (PPH), intonational phrase (IPH), is widely employed to distinguish different levels of pauses in sentences. The accuracy of predicting these boundaries largely determines the naturalness and even the intelligibility of synthesized speech.

Inspired by the successful use of BERT in many natural language processing (NLP) tasks, we proposed a BERT based prosodic boundary prediction model to improve prediction accuracy.

As shown in Figure 2, following our previous work[14], we treated hierarchical prosodic boundary prediction as related tasks and predicted them simultaneously by using multi-task learning. In the meantime, word segmentation was set as an auxiliary task to introduce word level information in the model as prosodic boundaries are highly correlated with lexicon words.

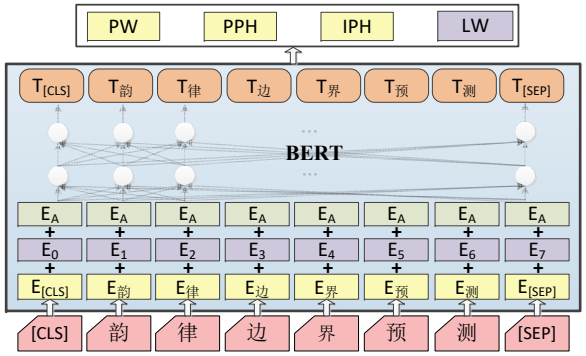

Figure 2: *The architecture of proposed prosodic boundary prediction model*

Specifically, for each task, we added an task-specific output layer to a pre-trained Chinese model[1], then used 8000 sentences with prosodic boundary labels to fine-tune the pre-trained model. The fine-tuned model was used to predict prosodic structure from text on both original transcripts and test sentences after text normalization.

3.2. Duration model

Duration model used phone-level linguistic features to predict frame numbers of each phone. We used phoneme boundary information got by force alignment as target. The model was composed of a 512 units full connected (FC) layer following by two bidirectional long short-term memory (BLSTM) layers with 512 units in each direction. Mean square error (MSE) criterion with Adam[15] optimizer was used in the training process.

3.3. Acoustic model

Traditional acoustic model directly predicts acoustic parameters using linguistic features as the input. However, the data provided this year was collected from an Internet talk show, the content was repetitive and phoneme coverage was limited. As extra data from other speakers was allowed according to the challenge rules, we adopted a multi-speaker based acoustic model in our system. A 7-hour corpus from another male speaker was additionally used.

At first, we mainly focused on adaption training [16, 17, 18]. Following [18], we built a linear networks based speaker adaptation model. However, experiments showed that this method learned more speech channel differences than that of speakers due to the poor speech quality.

The second system based on speaker labels was then attempted. A simple idea was to encode the speaker information to one-hot vectors, and concatenate these vectors to original linguistic features to distinguish different speakers. In preliminary experiments, we built our acoustic model using the same structure as the duration model. But there was severe over-smoothing effect in synthetic speech. We believed that part of the reason was due to the large channel differences in training data. To address this issue, we introduced VAE residual encoder extension to the acoustic model, inspired by its success in speech style control and transfer in end-to-end structure[19].

[1] https://github.com/google-research/bert

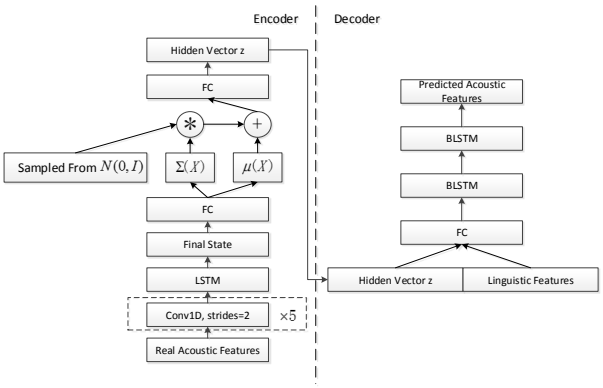

Figure 3: *Acoustic model based on variational autoencoder*

The structure of the whole acoustic model was given in Figure 3. To synthesize speech in a clean channel, we chose a high quality voice to compute the hidden vector by VAE encoder. However, it made the synthetic speech all the same rhythm. To enrich the tone and rhythm of synthetic speech, we built an independent model to predict F0.

3.3.1. Acoustic model based on VAE

As shown in Fig 3, our proposed VAE based acoustic model was composed of two components, including an encoder and a decoder. The encoder aimed to convert the real acoustic features into sampled hidden vectors z. Meanwhile, the decoder used hidden vectors to restore acoustic features under the condition of linguistic features.

Specifically, real acoustic features x are fed into the encoder to predict the posterior $q(z|x)$ during training. Given the true posterior $p(z|x)$, we can estimate the data distribution $p(x)$ as:

$$\log p(x) = \mathcal{L}(x,z) + D(q(z|x)\|p(z|x)) \approx \mathcal{L}(x,z) \tag{1}$$

where $\mathcal{L}(x,z)$ is Evidence Lower Bound (ELOB).

$$\mathcal{L}(x,z) = \mathbb{E}_{q(z|x)}(\log p(x|z)) - D(q(z|x)\|p(z)) \tag{2}$$

In eq.2, $p(x|z)$ is conditional distribution modeled by decoder. The first term is the reconstruction loss from decoder, and the second term is regularization term from encoder. Generally, we regard the prior $p(z)$ and $q(z|x)$ as Gaussian distributions. Given $p(z) \sim \mathcal{N}(0,I)$, then the loss of encoder is calculated by:

$$\mathcal{L}_{enc} = \sum_{j=1}^{J}\{\Sigma_j^2(x) + \mu_j^2(x) - \log\Sigma_j^2(x) - 1\} \tag{3}$$

where J means the dimensions of Gaussian distribution, $q(z|x) \sim \mathcal{N}(\mu(x), \Sigma^2(x)I)$, $\mu_j(x)$ and $\Sigma_j(x)$ the j-th value of $\mu(x)$ and $\Sigma(x)$ respectively. Therefore, the encoder predicts $\mu(x)$ and $\Sigma(x)$, totally $2J$ dimensions. To ensure the gradient can be computed, the reparameterization trick is applied to sample z:

$$z = u(x) + \Sigma(x) \odot \xi \quad \xi \sim \mathcal{N}(0,I) \tag{4}$$

The structure of decoder is similar to traditional acoustic model except adding hidden vectors z as extra inputs. Therefore, the loss of decoder is simply MSE loss. Because a quantized F0 model is built independently, the binary voiced flag and the F0 value are not predicted in the decoder. Finally, the decoder loss and total loss are written as:

$$\mathcal{L}_{dec} = \sum_{j=1}^{D}(\tilde{x}_j - x_j)^2 \tag{5}$$

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{enc} + \mathcal{L}_{dec} \tag{6}$$

where α gradually increases from 0 along with the training steps to prevent $\mathcal{L}_{enc}$ from decaying to 0.

In the synthesis stage, hidden vectors were extracted from a certain speech file in training data which has less noise and clearer channel. Since the VAE learned the speech style at the same time, the F0 counters of all the synthesized speech were similar. In order to ensure the diversity of the synthesized speech style, we additionally used a quantized F0 model to predict F0. In the training stage, we used the true F0 value as extra input of the acoustic model. In the synthesis stage, the predicted F0 value was used as input.

The detailed model structure was as follows. Encoder used 5 1D convolution layers with stride of 2 to shorten the speech sequence. Then an LSTM layer was used to get temporal information, the total sentence was represented by the final state. Following that, the mean and variance of hidden vector was computed by an FC layer. The hidden vector was sampled by reparameterization trick, and was fed into an FC layer. After that we got a sentence level hidden vector and repeated it to frame level to feed into the decoder. The decoder structure was the same as the duration model.

3.3.2. Quantized F0 model

F0 is an important feature to depict the tone and prosody of speech. Traditional acoustic model predicts F0 together with the other acoustic features. However, using MSE loss as the optimization function may not suitable in F0 prediction because it's an one-to-many prediction problem. For instance, we may speak a same sentence twice with different rhythm. For the acoustic model, the inputs are the same but the targets are different. Once MSE loss is adopted, we will learn an averaged F0 value. Simply, providing 'wo', which means 'I' in English, appears twice with different F0 values, F_1 and F_2, then $\frac{F_1+F_2}{2}$ is the best value to minimum the MSE loss. Obviously, this value never appears among the data, and will cause over-smoothing phenomenon.

Motivated by [20], we built a quantized F0 model. The minimum value F_{min} and maximum value F_{max} of raw F0 data were counted firstly. Then the F0 value was encoded as a one-hot vector $o_t = [o_{t,0}, o_{t,1}, ..., o_{t,N}]$. If the frame was unvoiced, $o_{t,0} = 1$; otherwise, $o_{t,0} = 0$, the other dimensions were all zeros except one dimension is 1 according to F0 value. We adopted the similar hierarchical softmax layer in [20], where the probability of each quantization channel can be calculated as:

$$P(o_{t,j}|h_t) = \begin{cases} \dfrac{e^{h_{t,0}}}{e^{h_{t,0}}+1} & if \quad j=0 \\ \dfrac{1}{e^{h_{t,0}}+1}\dfrac{e^{h_{t,j}}}{\sum_{i=1}^{N}e^{h_{t,i}}} & if \quad j\neq 0 \end{cases} \tag{7}$$

where $h_{t,j}$ is the j-th output of F0 model. KLD is used as the loss funtion, which is written as:

$$\mathcal{L}_{KLD}^{F0} = \sum_{j=0}^{N} o_{t,j}\log P(o_{t,j}|h_t) \tag{8}$$

Because the unvoiced frames were far more than the frames with a certain F0, a threshold was set to determine whether the frame was an unvoiced frame. We used the same threshold as [20]. Only if $\frac{e^{h_{t,0}}}{e^{h_{t,0}+1}} > 0.5$, $o_{t,0} = 1$, otherwise, the other channels with the maximum probability was set to 1.

In our final system, we used multi-task learning to predict quantized F0 and the other acoustic features. In fact, experiments showed predicting F0 separately will lead to mismatch with the mel-generalized cepstral. The structure of F0 prediction model was the same as duration model too.

4. Results

4.1. Internal evaluation

Three acoustic models were compared in our experiment:

1. **Baseline**: BLSTM based acoustic model.
2. **BLSTM-V**: Multi-speaker acoustic model based on VAE.
3. **BLSTM-QV**: Multi-speaker quantized acoustic model based on VAE.

The structure of **BLSTM-V** had been shown in section 3.3. The filter numbers of 1-dimensional convolution layers were all 128. The dimension of Gaussian distribution was 8. Then a vector sampled from the predicted Gaussian distribution was fed into a fully connected network. Finally, the dimension of hidden vector was 128. For **BLSTM-QV**, frame level acoustic features were analyzed to compute the range of F0 value. The minimum value d_{min} and the maximum value d_{max} of raw duration data were 58 Hz and 571Hz. The result was shown in Table 1.

Table 1: *RMSE of different duration models*

model	MCD(dB)	RMSE(Hz)	CORR	UV(%)
Baseline	7.553	42.117	0.603	17.229
BLSTM-V	6.779	41.108	0.611	17.036
BLSTM-QV	6.788	39.978	0.647	15.710

The result showed that using multi-speaker training and VAE extension decreased mel-cesptral distortion (MCD). A reasonable explanation was that different people had similar pronunciation characteristics and extra data expanded the quantity of different phones. Although VAE extension used true acoustic features as input, it showed the upper bound of the acoustic model, and made the channel and rhythm of synthesized speech closer to real speech. Quantized F0 model decreased F0 RMSE and U/V error percentage, which was associated with speech tone and prosody. We used **BLSTM-QV** acoustic model for our submitted system.

4.2. Evaluation results

The listening test results in Blizzard Challenge 2019 were presented below. Including Merlin baseline, 24 submitted systems and natural speech were evaluated. The identifiers for the benchmark and our system are:

- A: natural speech
- B: merlin baseline
- N: our system

Each audio was evaluated over Mean Opinion Score (MOS), similarity, Pinyin Error Rate (PER) and Pinyin Tone Error Rate (PTER). The MOS and similarity results were based on all the listeners' responses, including paid listeners at Edinburgh, volunteers and experts. The PER and PTER were mainly based on paid listeners' responses.

4.2.1. MOS test

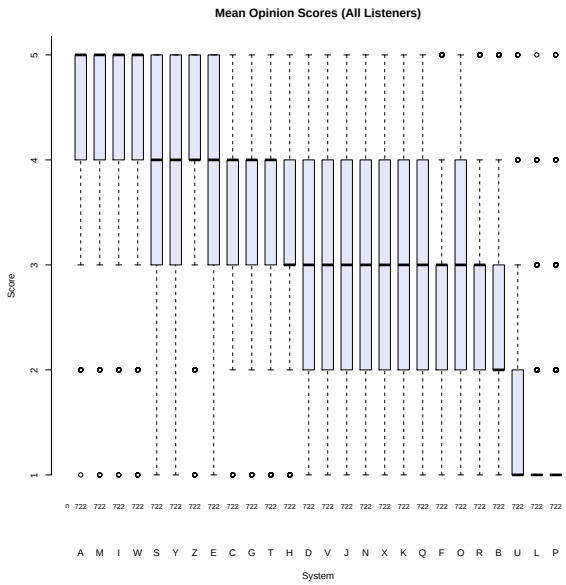

Figure 4: *Boxplot of MOS comparing to original speaker*

Fig 4 illustrated the boxplot of evaluation results of all systems on MOS. Apart from the nature speech, our proposed system ranks 14th over participants with the other two teams. Our system used the pipeline SPSS structure, couldn't synthesize voice as natural as end-to-end model. And we only used a little extra speech data, which constrained our system's performance.

4.2.2. similarity test

The similarity results of all systems were presented in the Fig 5. Our system performed not well on similarity overall. Because our system and Merlin baseline shared the same vocoder which constrained the synthesis quality and speech detail. And our VAE based acoustic model used a hidden vector computed from a clean speech, which had a different channel from most of the other speech. We think the listeners would consider channel information when they did similarity test.

4.2.3. PER and PTER test

The boxplot of PER and PTER results were presented in Fig 6 and Fig 7. Our system achieved the lowest PER and PTER among all the submitted system which meant that our synthetic speech was most intelligent. We believed it was benefit from accurate front-end processing including text normalization, G2P conversion, prosodic boundary prediction. Another reason was that the pipeline SPSS acoustic model had higher robustness than the end-to-end model and concatenate-based model. The VAE extension decreased the noise of synthetic speech and made the synthetic speech sound more clearly. Besides, quan-

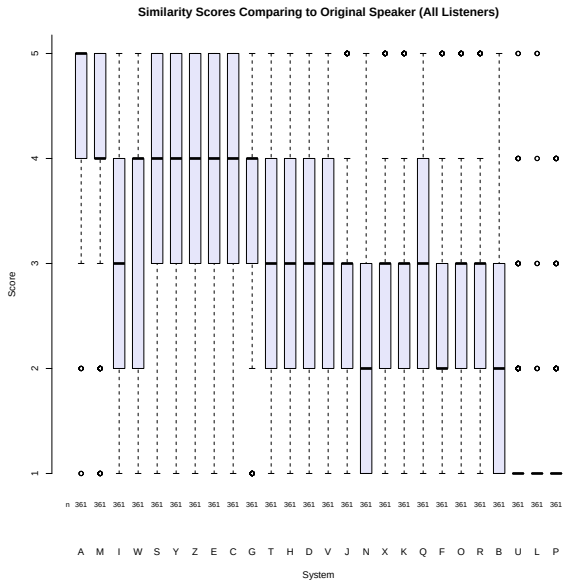

Figure 5: *Boxplot of similarity scores comparing to original speaker*

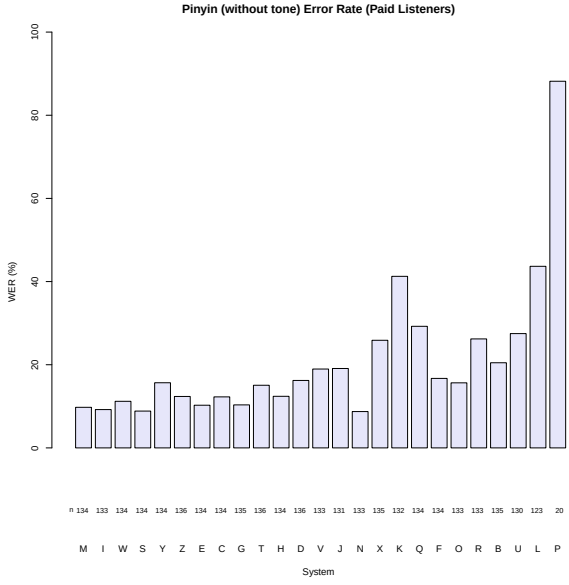

Figure 6: *pinyin (without tone) error rate*

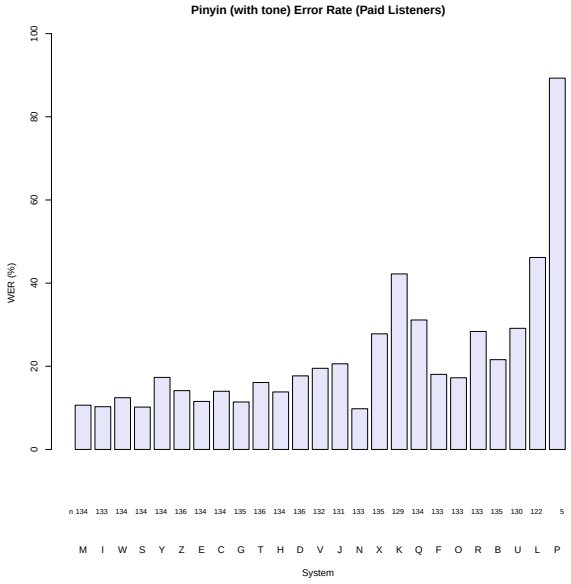

Figure 7: *pinyin (with tone) error rate*

tized F0 model fits the F0 contour better which conveys the tone in an utterance more accurately.

5. Conclusions

In this paper, we introduced our system which was used in Blizzard Challenge 2019, including a prosodic boundary prediction model, a quantized F0 model and a multi-speaker acoustic model based on VAE. Evaluation results showed that our proposed model performed best on both PER and PTER among 24 teams, which showed the accuracy of our front-end analysis module and the robustness of our acoustic model. Because we could obtain only a little extra male speech, and the pipeline SPSS structure constrained the quality of synthetic speech, our system performed ordinarily on naturalness. Besides, the World vocoder we used may not produce the same speech quality as neural vocoder like WaveNet, and the VAE extension made the channel of synthetic speech different from most ground-truth speech, our system didn't perform well on similarity. In future work, we will mainly focus on putting prosody and F0 information into End-to-End model.

6. Acknowledgements

This work is partially supported by the National Natural Science Foundation of China (Nos.11590773,11590770), the Pre-research Project for Equipment of General System (No.JZX2017-0994/Y306).

7. References

[1] Z.-H. Ling, L. Qin, H. Lu, Y. Gao, and L.-R. Dai, "The ustc and iflytek speech synthesis systems for blizzard challenge 2007," in *The Blizzard Challenge 2007 Workshop*, 2007.

[2] J. Tao, Y. Zheng, Z. Wen, Y. Li, and B. Liu, "Blstm guided unit selection synthesis system for blizzard challenge 2016," in *The Blizzard Challenge 2016 Workshop*, 2016.

[3] Y. Jiang, X. Zhou, C. Ding, Y. jun Hu, Z.-H. Ling, and L.-R. Dai,

"The ustc system for blizzard challenge 2018," in *The Blizzard Challenge 2018 Workshop*, 2018.

[4] Y.-J. Hu, C. Ding, L.-J. Liu, Z.-H. Ling, and L.-R. Dai, "The ustc system for blizzard challenge 2017," in *The Blizzard Challenge 2017 Workshop*, 2017.

[5] K. Liu, F. Meng, Y. Song, B. Fan, W. Duan, and W. Chen, "The sogou speech synthesis system for blizzard challenge 2018," in *The Blizzard Challenge 2018 Workshop*, 2018.

[6] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," in *SSW*, 2016.

[7] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel wavenet: Fast high-fidelity speech synthesis," *ArXiv*, vol. abs/1711.10433, 2017.

[8] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP*, 2019, pp. 3617–3621.

[9] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019, pp. 4171–4186.

[10] M. Morise, F. Yokomori, and K. Ozawa, "World: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions*, vol. 99-D, pp. 1877–1884, 2016.

[11] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2014.

[12] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[13] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system." in *SSW*, 2016, pp. 202–207.

[14] C. Lu, P. Zhang, and Y. Yan, "Self-attention based prosodic boundary prediction for chinese speech synthesis," in *ICASSP*, 2019, pp. 7035–7039.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Multi-speaker modeling and speaker adaptation for dnn-based tts synthesis," in *ICASSP*, 2015, pp. 4475–4479.

[17] S. Pascual and A. Bonafonte, "Multi-output rnn-lstm for multiple speaker speech synthesis and adaptation," *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 2325–2329, 2016.

[18] Z. Huang, H. Lu, M. Lei, and Z. Yan, "Linear networks based speaker adaptation for speech synthesis," in *ICASSP*, 2018, pp. 5319–5323.

[19] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling, "Learning latent representations for style control and transfer in end-to-end speech synthesis," in *ICASSP*, 2018, pp. 6945–6949.

[20] X. Wang, S. Takaki, and J. Yamagishi, "Autoregressive neural f0 model for statistical parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 1406–1419, 2018.