The I2R-NWPU Text-to-Speech System for Blizzard Challenge 2017

Yanfeng Lu[*], *Zhengchen Zhang*[*], *Chenyu Yang*[*], *Huaiping Ming*[*], *Xiaolian Zhu*[†], *Yuchao Zhang*[†], *Shan Yang*[†], *Dongyan Huang*[*], *Lei Xie*[†], *Minghui Dong*[*]

[*]Human Language Technology Department,
Institute for Infocomm Research, A*STAR, Singapore, 138632
[†]Shaanxi Provincial Key Laboratory of Speech and Image Information Processing,
School of Computer Science, Northwestern Polytechnical University, Xian, China, 710129

Email: lxie@nwpu.edu.cn, mhdong@i2r.a-star.edu.sg

Abstract

We present I2R-NWPU team's entry to Blizzard Challenge 2017 in this paper. Like our previous entry, we still adopt the general deep neural network (DNN) guided unit selection and waveform concatenation method to synthesize the speech. But we make several important improvements to our previous system. Phone duration and frame level acoustic parameters are modelled with long short-term memory (LSTM) recurrent neural network (RNN). But this time we keep the hidden Markov model (HMM) to assist pre-selection. Phone level instead of frame level units are used in the selection and concatenation process. In synthesizing the speech, the Kullback-Leibler Divergence (KLD) between the predicted target and the candidate spectrum HMMs is used to preselect the units. Then the duration and acoustic parameters of the preselected units are predicted with the LSTM-RNN models. The final units are selected with the Viterbi algorithm based on the target and concatenation costs calculated against the predicted trajectory. The listening tests show improvement compared with our previous system.

Index Terms: Blizzard Challenge 2017, Text-To-Speech, LSTM-RNN, HMM, Unit Selection

1. Introduction

The EH1 task of Blizzard Challenge 2017 is essentially the same as the previous year. The participating teams are given several hours of audiobook data of the children's story genre, including speech and text, and then asked to synthesize 6 new audiobooks, 200 news items and 200 semantically unpredictable sentences (SUS). The text given is partially transcribed. The submitted synthesized speech is evaluated with systematic listening tests involving a variety of listeners. Compared with the task of the previous year, the only difference is that more data (the testing audiobooks from the previous year) are provided.

For Blizzard Challenge 2016 we adopted a deep neural network guided trajectory tiling method to build our text-to-speech (TTS) system [1]. Phone duration and acoustic parameters including LSP, F0 and V/UV flag are modelled with deep neural network in the training phase. In the synthesis phase, for each sentence phone duration and frame level acoustic parameters are predicted with the trained models. Then the trajectory tiling method [2] was used to synthesize the target waveform.

For Blizzard Challenge 2017 we make several important adjustments to our system. First, the units are changed from frames to phones. In synthesizing the target waveform, we select candidate phones and concatenate them together. Second, we use KLDs between spectrum HMMs of candidate and target phones to preselect the candidates. Third, in accordance with the unit change the target and concatenation costs are also redefined.

The rest of the paper consists of the following sections. In Section 2 we describe how we process the given data in preparing for the system building. The system building itself is presented in Section 3. The evaluation results are reported and analyzed in Section 4. And finally we summarize our work in Section 5.

2. Data Processing

2.1. Transcription and Alignment

The several audiobooks added to the training dataset this year are not transcribed. So the first thing we do is to transcribe these new audiobooks. The provided audio files are converted to 16K Hertz 16 bits wave files. Then they are split into sentences and the corresponding text is split accordingly. These new data are combined with the training data from last year. The combined training data go through the same alignment process as last year. Our phone alignment is based on a model in our Automatic Speech Recognition (ASR) engine trained with a large

database.

2.2. Full-Context Labels Generation

The phone level full-context features we use this year are the same as last year. They include features on phoneme, syllable, word and syntactic phrase levels. They are to a large extent based on [3]. And we add the syntactic features following [4]. However, this year's full-context labels are generated on the basis of the ASR alignment results. Compared with last year's method there are two differences. First, the phone sequence of a word is based on the recognition result, instead of being arbitrarily picked from the dictionary. Second, silences are also inserted on the basis of the recognition result. In this way, the full-context labels generated match the speech database more closely.

2.3. Sentence Features Generation

Due to the large variance of prosody in the audiobooks, it's necessary to differentiate the types of sentences. We planned to include this in our entry to Blizzard Challenge 2016, but didn't manage to achieve that in time. This time we do put it in. Specifically, for each sentence or part of sentence we consider two kinds of properties: the relation to the quotation marks and the ending punctuation. For the former we define 4 possible values: 1 - within quotation; 2 - adjacent to quotation; 3 - far away from quotation; 0 - undefined. Value 2 covers parts of sentences such as [he said], [Tom asked], which are adjacent to quotation. We assume different values correspond to different prosodic styles. The ending punctuation also correlates with prosody. Questions and statements are obviously read differently. Besides these two features we include the sentence length in terms of word count as well. So we have totally 3 sentence features.

These sentence features are first computed on the word level. We consider the previous and next sentences together with the current one. We also include the forward and backward positions of a word in the sentence it belongs to. Therefore, for each word we have an 11 dimensional feature vector: 2 dimensions for word position, 3 sentence features each for the previous, current and next sentences. After these word level features are computed they are distributed to the phones which the word contains. So finally we have some extra features corresponding to the phone level full-context features.

We build systems with and without the sentence features described above and find that sentence features have significant positive effect.

2.4. Speech Data Filtering

In order to remove the over expressive audio files, we clean the dataset according to the phone duration and the F0 features. Some utterances containing mimetic words, such as shout and cry, might be exaggerated. The over expressive utterances usually have abnormal durations which are much longer or shorter than the average duration of phone. We cleaned these utterances out of the dataset, including 147 utterances, about 2.06%. The second method is to choose utterances according to the fundamental frequency (F0). In this case, the utterances with excessive low or high frequency are removed, which constitute a subset of 174 utterances, about 2.44% of the dataset. The removed utterances might be singing, onomatopoeia and over stressed utterances.

3. System Building

The TTS system we submitted this year still belongs to the category of DNN-guided unit selection. Generally phone duration and acoustic parameters are modelled with deep neural networks using the provided dataset in the training phase. In the synthesis phase, phone duration and acoustic parameters are first predicted, and then they are used to guide the target unit selection. Finally the selected units are concatenated to make the target waveform.

The architecture of the system is depicted in Figure 1. The top part is already discussed in Section 2. For HMM training we use the standard HMM-based speech synthesis system (HTS) package [5]. The spectrum parameters we use are mel-generalized cepstral (MGC). The text analysis in the synthesis phase is a little different from that in the training phase. In the training phase the phone sequence of a word is based on the ASR alignment. This is impossible in the synthesis phase. Instead, the phone sequence of a word is determined by the first instance in the dictionary. And breaks are inserted in the middle of a sentence on the basis of a set of rules applied to the syntactic tree, built with the ZPar package [6]. In unit pre-selection we pick 100 candidate units which have the smallest weighted KLDs from the target unit. Next we discuss the remaining major components in turn.

3.1. Phone Duration and Acoustic Parameter Modelling with LSTM-RNN

LSTM-RNN, introduced in 1997 [7], has been proved to be a very effective sequence modelling tool. A bidirectional RNN [8] captures the context more effectively. These two types of RNNs later were combined together to construct the bidirectional LSTM-RNN, or BLSTM-RNN for short.

A hybrid of DNN and BLSTM-RNN is built for our phone duration prediction and acoustic modelling. There

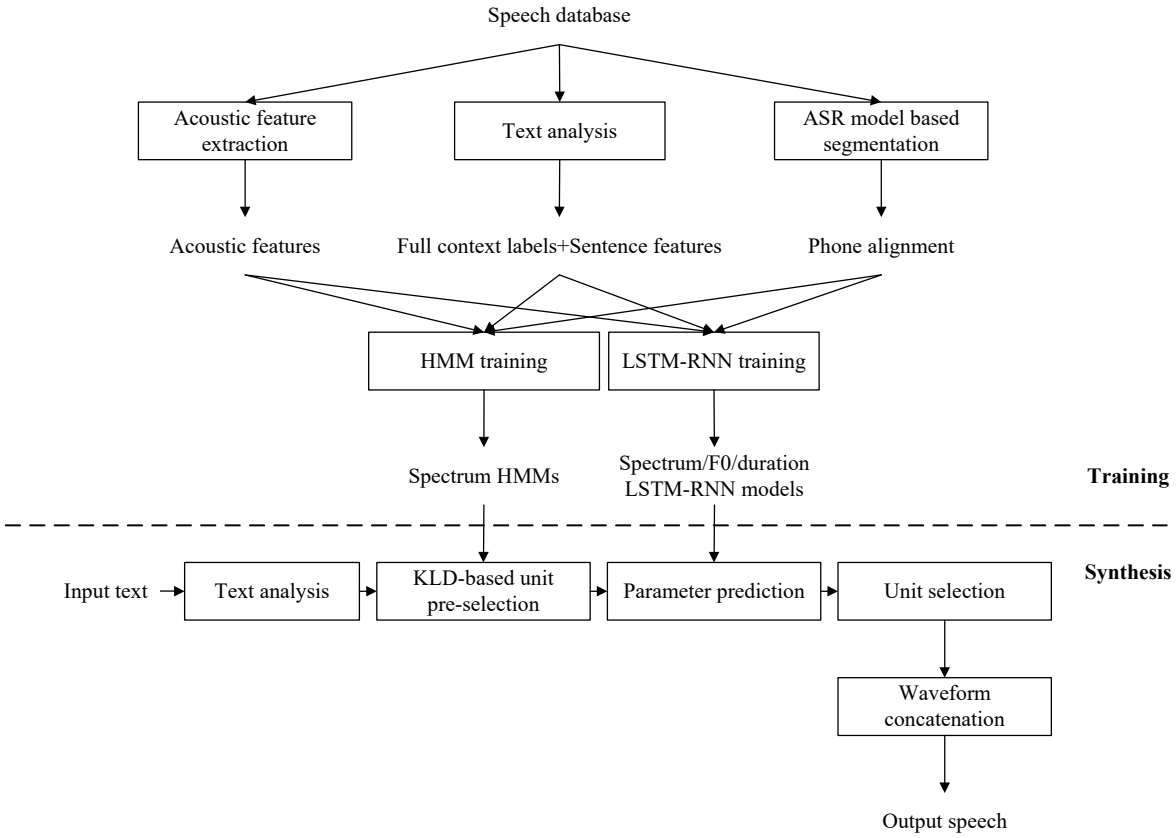

Figure 1: System Architecture.

are 6 hidden layers, where the bottom 3 hidden layers are feed-forward structure with 1024 nodes per layer, while the top 3 hidden layers are LSTM-RNN structure (512 forward nodes and 512 backwards nodes).

Input feature vectors for duration prediction are generated from the full-context labels we get from Section 2 and time-aligned frame-by-frame with the output features. The categorical features like phoneme IDs, POS types, and phrase types are converted into binary features. The positions of phonemes, syllables and words are numerical features. There are totally 797 dimensions in the input vectors, where 695 dimensions are binary features for categorical linguistic contexts, 51 dimensions are numerical linguistic contexts, and 51 dimensions are sentence-level features.

The input feature vectors for frame level acoustic modelling are almost the same as those for duration prediction, except 4 dimensions containing frame information are added. Of these, 3 dimensions are for coarse coded position of the current frame and 1 dimension for duration of the current segment. The output of the acoustic modelling network is a vector of 187 dimensions, which consists of MCCs, BAPs, log F0 and their delta and delta-delta features, plus a voiced/unvoiced flag.

We use the Merlin package [9] and the Computational Network Toolkit (CNTK) [10] to extract the acoustic parameters and train the deep neural networks.

3.2. Unit Selection

In selecting the optimal units for a target sentence, we follow the standard dynamic programming-based search according to target and concatenation costs [11].

3.2.1. Target and Concatenation Costs

With the predicted acoustic trajectory as the reference, the target cost of a candidate unit is straightforward to define. It's basically the acoustic distance between the candidate unit and the corresponding target unit. Specifically, it's defined as follows:

$$C_t = w_{dur}\,|d_t - d_c| + w_{f0}D_{f0} + w_{ap}D_{ap} + w_{sp}D_{sp}, \quad (1)$$

where C_t is the target cost, d_t and d_c are the target and candidate phone durations, D_{f0}, D_{ap} and D_{sp} are the F0, aperiodicity and spectrum distances, and w_{dur}, w_{f0}, w_{sp} and w_{sp} are the corresponding weights.

In most cases the durations of the target and candidate phones are different. To handle this issue we do a simple linear alignment of the frames. We prefer a simple alignment to minimize the computation cost. On the aligned frames, D_{f0}, D_{ap} and D_{sp} are defined as the mean absolute log F0 difference, absolute aperiodicity difference and Euclidean distance between the spectral parameter vectors.

In calculating the concatenation cost we consider two major factors: the continuity of candidate units in the speech database and the boundary distance between two adjacent candidate units. The first is meant to favor bigger chunk of natural speech. And the second is for making the synthesized speech as smooth as possible. So the concatenation cost is defined as follows:

$$C_c = w_{frag}C_{frag} + w_{bound}D_{bound}, \qquad (2)$$

where C_c is the concatenation cost, C_{frag} is the fragmentation cost, D_{bound} is the boundary distance, and w_{frag} and w_{bound} are the corresponding weights.

All the weights are manually tuned. To facilitate the weight tuning, some distances are normalized to bring them to the same scale.

3.2.2. Dynamic Programming-based Search

With the target costs of single candidate units and the concatenation costs between adjacent candidate units, Viterbi search is used to find the candidate unit path that has the least accumulated joint costs. To do a complete search in the whole unit path space is very computation costly. Therefore, pruning has to be applied in order to make the search computationally manageable. One kind of pruning is candidate unit pre-selection. This is based on the combined KLDs between the predicted target HMMs and the candidate HMMs. Target costs are only computed for pre-selected units. Another kind of pruning is applied to the candidate unit paths. In Viterbi search the candidate paths are constructed unit by unit. At each step we only keep a certain number of optimal candidate paths before we move on to the next unit. Both the number of pre-selected candidate units and the number of candidate paths kept are tunable parameters of the system.

3.3. Waveform Concatenation

To further improve the smoothness of the synthesized speech, we also use some techniques in waveform concatenation. Naturally adjacent units are concatenated directly to keep naturalness. Other units are concatenated with triangular cross-fading method. The cross-fading point is optimally selected according to certain measure. The measure is the Euclidean distance between the overlapping segments from the two units to be concatenated. To do this extension has to be made when we take the waveforms of the units from the database.

4. Subjective Evaluation Results

The submitted synthesized speech files go through comprehensive listening tests. The listening tests include four major parts. Part 1 consists of two multi-dimensional tests of the book paragraphs. The tested dimensions are overall impression, pleasantness, speech pauses, stress, intonation, emotion and listening effort. Part 2 contains two naturalness tests of the book sentences. Part 3 is a similarity test of the book sentences. The listeners are requested to judge how similar the synthesized speech is to the provided speech. This is a restriction on using extra speech data. Part 4 consists of two intelligibility tests of the SUS speech. The sentences tested in this part are semantically unpredictable, so it's difficult to guess a word through the surrounding words. The listeners are requested to write down the words after single listening of a sentence. Three types of listeners are involved in the tests: paid listeners, online volunteers and speech experts.

In reporting the test results we include the overall impression, naturalness, similarity and intelligibility as evaluated by all types of listeners. We compare our results of this year with those of last year. We also compare our results with other sytems. All the figures show results of all the systems. System A, B, C and D are for references. They are not submitted by the participating teams. A is natural speech. B is the unit selection system benchmark. C is the HTS system benchmark. And D is the DNN system benchmark. Our system is labelled "K" this year.

Figure 2 shows the boxplot of the overall impression MOS of audiobook paragraphs given by all listeners for all the systems. Compared with our score last year there is some improvement in the overall impression MOS this year. We also see improvements in some other participating teams. So our ranking is about the same. However, now we surpass system D and is on a par with system B, the best synthesized reference system, in this respect, as the significant difference matrix shows. Figure 3 shows the boxplot of the naturalness MOS of all the systems given by all listeners. We also make improvement in this respect. Figure 4 shows the boxplot of the similarity MOS of all the systems given by all listeners. Here we see more significant improvement. These improvements to a large extent result from our changing the units from frames to phones. Figure 5 shows the word error rate of all the systems based on the evaluation of all listeners. There is an increase in the

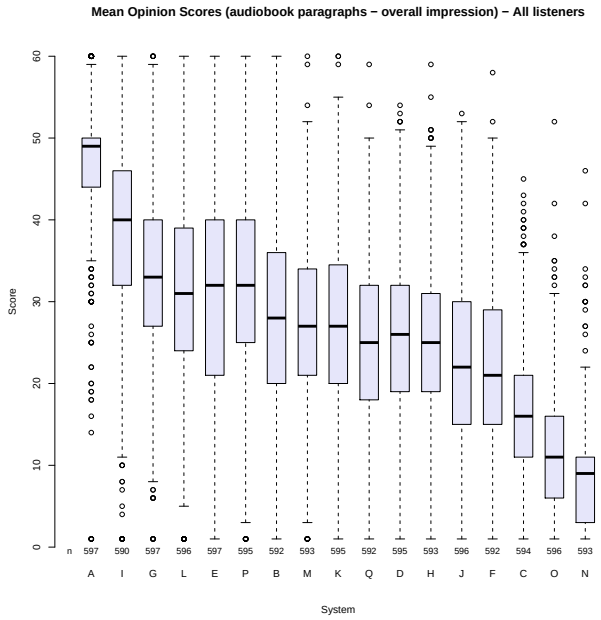

Figure 2: Boxplot of overall impression MOS of audiobook paragraphs.

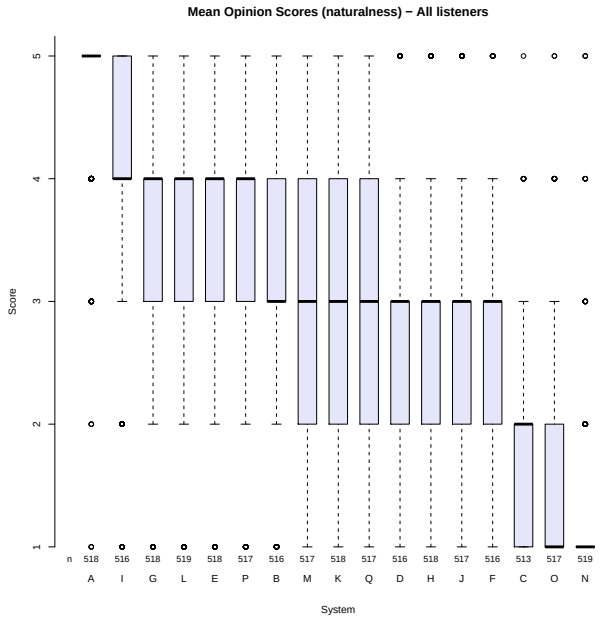

Figure 3: Boxplot of naturalness MOS.

absolute value of WER compared with our result of last year. But all the systems experience the same increase. Factoring out this system shift, there is no degradation of our system in intelligibility from last year.

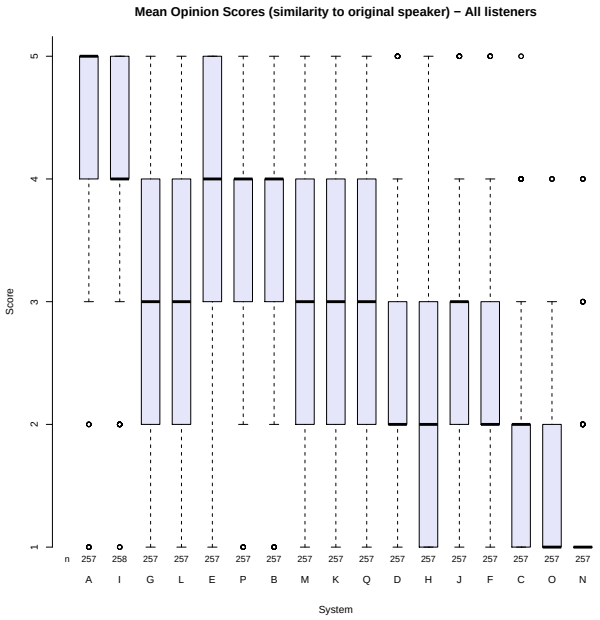

Figure 4: Boxplot of similarity MOS.

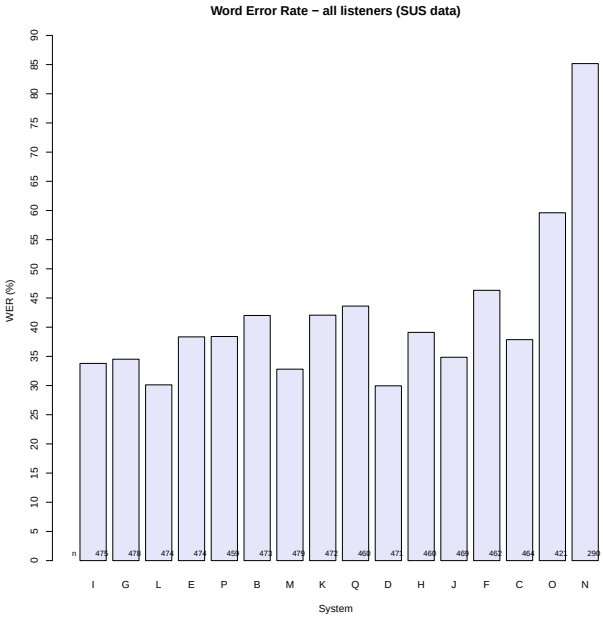

Figure 5: WER in the SUS testing.

5. Conclusion

In this paper we present our entry system to Blizzard Challenge 2017. Again we adopt a general deep neural network guided unit selection and waveform concatenation architecture as last year. But we make several important adjustments to our system this year. They include changing the units from frames to phones, incorporating sentence features, spectrum HMM KLD

based unit pre-selection and new ways of calculating the target and concatenation costs. Subjective evaluation results demonstrate that we've made significant progress since last year. But compared with other better systems we could improve our system further in the future.

6. References

[1] Zhengchen Zhang, Mei Li, Yuchao Zhang, Weini Zhang, Yang Liu, Shan Yang, and Yanfeng Lu, "The i2r-nwpu-ntu text-to-speech system at blizzard challenge 2016," .

[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, "An example of context-dependent label format for hmm-based speech synthesis in english," *The HTS CMUARCTIC demo*, vol. 133, 2006.

[4] Yansuo Yu, Fengyun Zhu, Xiangang Li, Yi Liu, Jun Zou, Yuning Yang, Guilin Yang, Ziye Fan, and Xihong Wu, "Overview of shrc-ginkgo speech synthesis system for blizzard challenge 2013," in *Blizzard Challenge Workshop*, 2013, vol. 2013.

[5] 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.

[6] Yue Zhang and Stephen Clark, "Syntactic processing using the generalized perceptron and beam search," *Computational linguistics*, vol. 37, no. 1, pp. 105–151, 2011.

[7] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] Mike Schuster and Kuldip K Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[9] Zhizheng Wu, Oliver Watts, and Simon King, "Merlin: An open source neural network speech synthesis system," *Proc. SSW, Sunnyvale, USA*, 2016.

[10] Dong Yu, Adam Eversole, Mike Seltzer, Kaisheng Yao, Zhiheng Huang, Brian Guenter, Oleksii Kuchaiev, Yu Zhang, Frank Seide, Huaming Wang, et al., "An introduction to computational networks and the computational network toolkit," *Microsoft Technical Report MSR-TR-2014–112*, 2014.

[11] 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.