# Submission from vivo for Blizzard Challenge 2019

*Yang Wang, Mingyue Wu, Zhiming Xu, Baoqin Luo, Hao Liang, Binbin Chen, Xiaoxin Chen*

vivo AI Lab, Shenzhen, P.R. China

yang.wang.rj@vivo.com

## Abstract

This paper presents the vivo speech synthesis system for Blizzard Challenge 2019. The task is to build an expressive speech synthesis system on an 8-hour corpus of a well-known Chinese talk-show character. Our system is based on Tacotron with several minor improvements, which are more clear speech energy normalization, outlier removal of problematic shorter utterances, and special phone modelling for explicit long silences, audible breath sounds, and mouth-click sounds. Evaluation results showed that the proposed system is somewhat successful.

**Index Terms**: Blizzard Challenge 2019, speech synthesis, end-to-end, Tacotron, special phone modelling

## 1. Introduction

Blizzard Challenge (BC) was organized annually since 2005 to evaluate and compare corpus based speech synthesis techniques on common datasets [1]. In the past few years, the most popular approaches are Statistical Parametric Speech Synthesis (SPSS), unit selection based waveform concatenation, and hybrid approach concatenating waveforms typically guided by Hidden Markov Model (HMM) or Deep Neural Network (DNN). High-level comparison for methods implemented in submitted systems of past few years is presented in Table 1 of summary papers [2, 3].

Unit selection systems and hybrid systems could take advantage of natural speech segments directly, thus produce better naturalness most of the time, e.g., the best system last year is a hybrid system [4]. An interesting visualization for this trend is depicted in Figure 2 in summary paper for BC 2017 [2]. One disadvantage of these systems is its less flexibility: it demands laborious expert fine-tuning on a specific speech corpus for a specific language, thus it requires non-trivial work to develop a synthesis system on a new speech corpus or for a new language. On the contrary, SPSS converts waveform into some parametric form, and trains an acoustic model to predict acoustic features. A relatively independent vocoder, such as WORLD [5], is required to reconstruct speech waveform from predicted acoustic features. Despite that speech quality is restricted by vocoder, SPSS systems are more flexible. In recent years there is clear research paradigm shift from HMM based SPSS [6] to end-to-end DNN based SPSS, which makes SPSS more attractive since it requires much less domain knowledge, and it can be trained in a unified, end-to-end way, which is expected to lead to better performance. Thus many variants of end-to-end systems emerged quite recently, such as Tacotron [7] and its families, ParaNet [8], MelNet [9].

HMM based SPSS is motivated by physically and mathematically straightforward model, thus corresponds well to our intuition, and is typically easy to explain. On the contrary, DNN based SPSS is inspired by the extraordinary advancement of DNN research in the past years [10], but this advancement only spreads its great power for SPSS since 2015 or so. DNN based SPSS tells us many success stories by showing its unexpectedly good performance over that of HMM based SPSS, despite it is typically not easy to explain persuasively and intuitively why it is successful.

Our submitted system is based on Tacotron [7], a breakthrough of end-to-end, DNN based SPSS recently. Several minor and successful improvements that we made are:

- More clear speech energy normalization solved energy abnormality problem in synthesized speech.
- Removing outliers in automatically segmented utterances greatly alleviated trailing silence abnormality, and increased model accuracy a little.
- Modelling special phones improved speech prosody significantly.

The rest of this paper is organized as follows. Section 2 describes how the system was built as well as our improvements. Section 3 presents the evaluation results. We conclude this paper in Section 4.

## 2. System building

This section describes how we preprocessed training corpus released by BC organizing committee, how we converted Chinese characters to corresponding Pinyin sequence, and what improvements we tried based on which open source implementation of end-to-end synthesis system Tacotron.

### 2.1. Corpus preprocessing

The released corpus has 8-hour speech data from an internet talk show speaker. Informal checking showed that the provided text transcription does not match the actual speech content accurately enough. Besides, each utterance is about 60 seconds, thus we had to segment these long utterances to shorter ones to better fit them to our model training process. We did not use any external corpora.

#### *2.1.1. Text cleaning*

Initial rough check showed a 5-10% word error rate of the provided text transcription, thus we organized annotators to correct these mis-typed Chinese characters.

#### *2.1.2. Utterance segmentation*

The released utterances are about one minute each. This was regarded to be significantly longer than our expectation. Initial model training only produced non-intelligible speech, which was presumably caused bad alignment due to too long utterances.

Thus, we automatically segmented utterance into shorter ones by utilizing three empirical thresholds, which are duration

of tentative shorter utterance, duration of silence or pause, and speech energy in each frame. Durations of shorter, segmented utterances demonstrated a distribution in Figure 1, which was in accordance with our experience on other speech synthesis corpora. In short, the 480 utterances in corpus were segmented to 4700 shorter ones. This simple segmentation method was acceptable but not ideal, and we eliminated most of resulted problematic shorter utterances, as described in detail in Section 2.4.2.

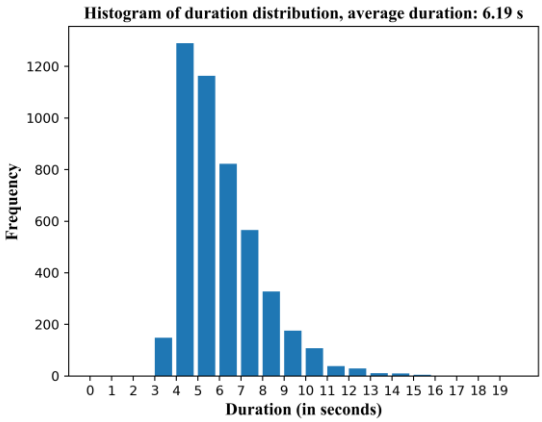


Figure 1: *Duration distribution of segmented shorter utterances.*

Utterance segmentation solved the trainable problem, after which we firstly produced intelligible speech in continuous endeavor of system development.

### *2.1.3. Punctuation normalization*

Several types of punctuation marks exist in text transcription. We retained only full stop, question mark, and comma, and each type of all other punctuation marks was replaced by one of these three marks accordingly. The detailed rule is shown in Table 1. This normalization step was motivated by the importance and frequency in text transcription, and slightly simplified symbol table in our system.

Table 1: *Punctuation normalization.*

| Before normalization | After normalization |
|---|---|
| Comma, full stop, question | As it is for each |
| Exclamation mark | Full stop |
| Colon, semicolon | Comma |

### *2.1.4. Pronunciation of English words*

Overall speaking, English words are rare in this corpus. There are about thirty English words occurring about 100 times. Very few of them occur more than several times, such as “App”, while most of them occur only once, such as “NASA” or “PPT”. Considering this rareness, it is suspicious that English words could be modelled intelligibly if we naively use a simple mixed Chinese English system.

A workaround was employed to alleviate this data sparsity problem: we simply chose the most similar Chinese Pinyin to mimic pronunciation of that English word. An incomplete table demonstrating this is shown in Table 2. Note specifically that word “App” has two different pronunciations depending on its actual speech content, and all other words shown in this table has only one pronunciation.

One unexpected consequence for us is that BC evaluation text includes 40 new English words not occurring in training text, thus we had to specify similarly Pinyin sequence for these out of vocabulary words. The extra work slightly hindered the goal that speech wave should be synthesized completely automatically.

Table 2: *English word pronunciation mimicking using Chinese Pinyin.*

| English words | Pinyin mimicking |
|---|---|
| App | ei1 pi1 pi5<br>ai4 po5 |
| OK | ou1 kei4 |
| Apple | ai1 pou5 |
| A | ei1 |
| vivo | vei1 vou5 |
| PPT | pi1 pi1 ti4 |
| Uber | wu1 ber5 |

Experiment results showed that this mimicking mechanism produced non-ideal but acceptable speech quality for this task.

## 2.2. Text frontend for character to Pinyin conversion

Although end-to-end approaches claim that they make traditional text frontend module unnecessary, this is true for phonetic language, such as English, French, and Italian. With regards to ideogram-based language Chinese, we assumed that it is still necessary to convert Chinese character to its Pinyin form. Thus, we utilized open source tool pypinyin (https://pypi.org/project/pypinyin) to convert Chinese character sequence to Pinyin sequence. Since the conversion accuracy is not good enough, annotators helped to correct predicted Pinyin sequence further.

Two special phenomena widely exist in Pinyin sequence, which affects naturalness significantly. The first is tone sandhi, and the most frequent subclass of tone-sandhi is 3-3 tone sandhi, e.g., Pinyin sequence “li3 xiang3” should be pronounced as “li2 xiang3” for Chinese word “理想”. We used an internal premature module to accomplish this goal. There are also other less frequent subclasses of tone sandhi, which were not handled in our system when writing this paper.

The second is tone softening and lightening occurring at the end of prosodic words from time to time, e.g., /hai2 zi3/ is generally pronounced as /hai2 zi5/ for word “孩子”. This phenomenon is a little tricky, and were not handled in our system either.

After character to Pinyin conversion, text line in training corpus looked like follows. Only Pinyin sequence was actually used for training, while character sequence is listed here for completeness and for ease of debugging when developing synthesis system.

> 100001_001|就是怎样演讲嘛。那罗永浩当然是这个方面的高手。|jiu4 shi5 zen3 yang4 yan2 jiang3 ma5. na4 luo2 yong3 hao4 dang1 ran2 shi4 zhei4 ge5 fang1 mian4 de5 gao1 shou3.

## 2.3. Tacotron system

Tacotron [7] is a seq2seq model [11, 12] with attention mechanism [13, 14], which includes an encoder, an attention based decoder, and a postprocessing network. From a high-level viewpoint, Tacotron maps sequence of characters to

sequence of spectral frames, which is further converted to waveform using a vocoder.

We give a brief description of Tacotron architecture below, but readers are suggested to refer to original paper for more detailed description [7]. Encoder in Tacotron aims at extracting robust sequential representations of text, which mainly consists of a pre-net and a CBHG module, where CBHG module further consists of 1-D convolutional filter bank, highway network [15], and bi-directional Gated Recurrent Network (GRU) [16]. CBHG module transforms pre-net's output to encoder's hidden representation, which is later decoded by decoder. The decoder is a stack of GRUs with residual connections [17]. Content-based or location-based attention could be used in decoder. Decoder predicts mel-scale spectrogram, and anther CBHG module is used as postprocessing step to correct prediction error for each individual frame. Lastly, Griffin-Lim algorithm [18] is used to convert predicted spectrogram to waveform.

After investigation of several open source implementations, we started from Mozilla TTS Tacotron, available from https://github.com/mozilla/TTS. This version of implementation has several improvements over original Tacotron paper, such as an explicit stop-net to determine when decoder should stop decoding. Our earlier experiments verified that this implementation produces good speech quality on several internal speech corpora. Hence it was chosen as our baseline system.

### 2.4. Our improvements on Tacotron system

We made several trials trying to further improve speech quality, some of which are successful, and thus summarized below.

#### *2.4.1. Speech energy normalization*

At the early phase of system development, a small portion of synthesized syllables was abnormal probably due to problematic energy control. After short time Fourier transformation in speech feature extraction, the energy at a frequency point of a speech frame was converted into dB scale, denoted as $x_{dB}$. Then the energy value was normalized as:

$$x_{dB\_normalized} = \frac{x_{dB} - ref_{dB} - min_{dB}}{-min_{dB}}, \quad (1)$$

and then clipped with lower and upper bounds as

$$x_{dB\_normalized\_clipped} = clip\left(x_{dB\_normalized}, 0, 1\right), \quad (2)$$

It can be seen in Eq. (1) that $-min_{dB}$ was used as normalization range, where $min_{dB}$ defaulted to −100 dB. However, this is counter-intuitive since a range is typically defined by a pair of lower bound and upper bound, but there is only a single lower bound $-min_{dB}$ here. This forced normalization range to coincide with minus of the minimum dB level $min_{dB}$. Besides, the $ref_{dB}$ should be fixed theoretically, since it is "reference level db, theoretically 20db is the sound of air" as written in comments of source code.

We thought this is obscure, thus modified Eq. (1) to Eq. (3):

$$x_{dB\_normalized} = \frac{x_{dB} - min_{dB}}{max_{dB} - min_{dB}} \quad (3)$$

Then we had a pair of lower bound and upper bound, which could be tuned independently, and we did not use $ref_{dB}$ anymore. The clipping Eq. (2) was used as usual. A grid search on $min_{dB}$ and $max_{dB}$ was then carried out to find the best combination for these two hyperparameters, i.e., $min_{dB} = -65, max_{dB} = 35$.

Experiment results showed that the energy abnormality in synthesized speech did not appear anymore after normalizing in this way.

#### *2.4.2. Outlier removal*

During system development, we noted that a small portion of synthesized utterance terminated abruptly with a few trailing words not pronounced at all. We assumed that this links to the fact of non-ideal utterance segmentation.

As explained in section 2.1.2, utterances were segmented automatically based on several empirical thresholds. Such kind of segmentation was not ideal, since about 5% of all shorter utterances terminated in middle part of a syllable, which made trailing silence model in each utterance difficult to align. Besides, a few utterances had significantly different recording background compared with majority of utterances, and a few segmented shorter utterances contained unusual content like laughter while speaking. We suspected that all these exceptions should be difficult to learn appropriately for an end-to-end neural network, thus we eliminated all these exceptions that annotators reported, resulting in about 6% elimination in terms of number of segmented shorter utterances.

One ironic thing for us is whether it should be considered as outlier when dealing with poems. Since poems are rare in training corpus, and poems have significantly different prosody compared with majority of normal, spontaneous speech, we eliminated poems as outliers during system development. However, the evaluation text has quite a large portion of poems, thus we had to make poems re-enter training corpus, and re-trained the system in a hurry.

Experiment results showed that trailing silence abnormality in synthesized speech was greatly alleviated, and speech intelligibility was improved a little as well, which is assumed to be benefited by better alignment due to cleaner training data.

#### *2.4.3. Special phone modelling*

The training corpus is expressive and spontaneous. Careful listening showed that there are plenty of audible breath sounds, less frequent mouth-click sounds, and other rare vocal phenomena. Thus, it was presumed that our system could be improved if we model these sounds as special "phones" explicitly. Since these special phones apparently occupy speech frames, if we do not model these special phones explicitly, corresponding speech frames must be aligned with neighboring common phones, and these common phones would be obscured in duration characteristic and intelligibility.

We added special phone annotations as specified in Table 3. Annotators reported that the first three, - long silences, audible breaths, and mouth-clicks -, could be annotated reliably, but the last special phone, gasp, was difficult to distinguish with audible breath. Thus, we simply merged gasp phone into much more frequent audible breath phone. After adding special phone, a text line in training corpus looked like below. Please note the added special phone, long silence "-" and audible breath "h", as indicated in blue color.

100001_001|就是怎样演讲嘛。那罗永浩当然是这个方面的高手。|- jiu4 shi5 zen3 yang4 yan2 jiang3

ma5. **h** na4 luo2 yong3 hao4 dang1 ran2 shi4 zhei4 ge5 fang1 mian4 de5 gao1 shou3.

Table 3: Special phone modelling

| Special phone | Symbol | description |
|---|---|---|
| Long silence | - | At sentence boundary, high frequency |
| Audible breath phone | h | Notable breath, high frequency |
| Mouth-click phone | bl | When mouth open, less frequent |
| Gasp phone | s | Gasp only once, rare |

At synthesis stage, we naively added an audible breath phone to the beginning and sentence boundaries of each text line to be synthesized.

Experiment results showed that speech prosody improved notably, and the speech intelligibility improved as well, as we expected.

# 3. Evaluation

Listening test results of our system for Blizzard Challenge 2019 is presented in this section. 25 systems, including 1 benchmark and 24 submitted systems, as well as natural speech were systematically evaluated. Identifiers for natural speech, the benchmark system and our system are:

- A: Natural speech
- B: Benchmark Merlin
- J: Our system

## 3.1. Naturalness evaluation

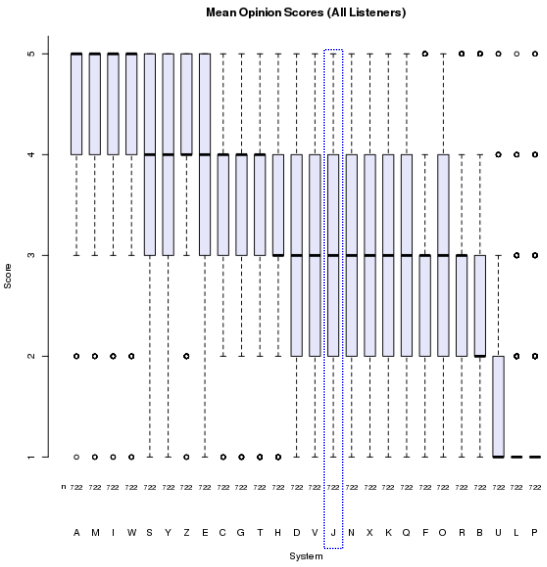


Figure 2: *Boxplot of naturalness scores of each submitted systems for all listeners.*

Boxplot of naturalness evaluation results of all systems is shown in Figure 2 for comparison, where our system is highlighted in dashed rectangular. It is obvious that we get middle level score.

## 3.2. Similarity evaluation

Boxplot of similarity evaluation results of all systems is shown in Figure 3 for comparison. Again, we get middle level score.

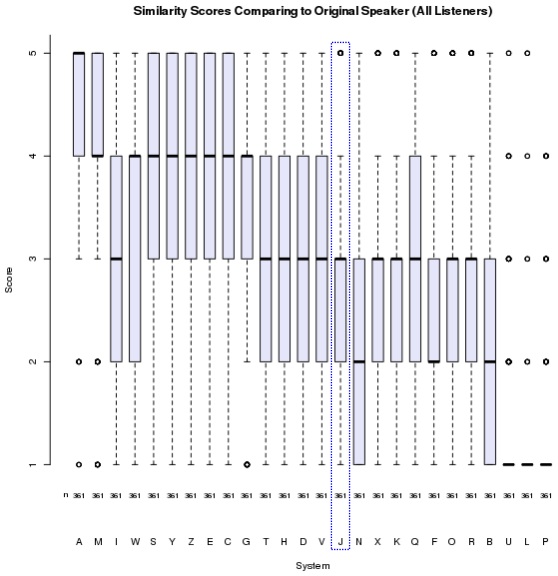


Figure 3: *Boxplot of similarity scores of each submitted systems for all listeners.*

## 3.3. Discussion

Two kinds of deficiency were noted in our system. Firstly, a few synthesized long sentences terminate unexpectedly and abruptly, indicating that occasionally the current stop-net module did not work properly. Thus, the stop-net module should be improved.

Secondly, in synthesis phase we segmented text line from paragraph level to shorter sentences, synthesized each shorter sentence independently, and concatenated synthesized speech to compose longer speech wave restored at paragraph level. This divide-and-conquer mechanism incurred noticeable speech background switching. In other words, expressive consistency across consecutive sentences within a paragraph was not well maintained. Thus, other synthesis mechanism being able to handle a whole paragraph directly, is worth to explore in future.

# 4. Conclusions

We submitted an end-to-end system based on Tacotron. Several minor improvements were tried to improve speech quality further, including more clear speech energy normalization, outlier removal of problematically segmented shorter utterances, and special phone modelling. Evaluation results in Blizzard Challenge 2019 demonstrated that these trials are somewhat successful.

# 5. References


[1] A. W. Black and K. Tokuda, "The Blizzard Challenge-2005: Evaluating corpus-based speech synthesis on common datasets," in *Interspeech*, 2005, pp. 77-80.

[2] S. King, L. Wihlborg, and W. Guo, "The Blizzard Challenge 2017," in *Blizzard Challenge workshop*, 2017.

[3] S. King, J. Crumlish, A. Martin, and L. Wihlborg, "The Blizzard Challenge 2018," in *Blizzard Challenge workshop*, 2018.

[4] Y. Jiang, X. Zhou, C. Ding, Y.-j. Hu, Z.-H. Ling, and L.-R. Dai, "The USTC System for Blizzard Challenge 2018," in *Blizzard Challenge workshop*, 2018.

[5] 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*, vol. 99, no. 7, pp. 1877-1884, 2016.

[6] H. Zen, K. Tokuda, and A. W. Black, "Statistical Parametric Speech Synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039-1064, 2009.

[7] Y. Wang *et al.*, "Tacotron: Towards End-to-End Speech Synthesis," in *Interspeech*, 2017, pp. 4006-4009.

[8] K. Peng, W. Ping, Z. Song, and K. Zhao, "Parallel Neural Text-to-Speech," *arXiv e-prints*. [Online]. Available: https://arxiv.org/abs/1905.08459

[9] S. Vasquez and M. Lewis, "MelNet: A Generative Model for Audio in the Frequency Domain," *arXiv e-prints*. [Online]. Available: https://arxiv.org/abs/1906.01083

[10] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org/.

[11] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," in *Advances in neural information processing systems*, 2014, pp. 3104-3112.

[12] K. Cho *et al.*, "Learning Phrase Representations using RNN Encoder–Decoder," *arXiv preprint arXiv:1406.1078*, 2014.

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *arXiv preprint arXiv:1409.0473*, 2014.

[14] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a Foreign Language," in *Advances in neural information processing systems*, 2015, pp. 2773-2781.

[15] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[16] 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.

[17] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[18] D. Griffin and J. Lim, "Signal Estimation from Modified Short-Time Fourier Transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236-243, 1984.