# The NTUT's Text-to-Speech System for Blizzard Challenge 2018

*Yuan-Fu Liao*[1], *Ya-Bo Chai*[2] *and Cheng-Hung Tsai*[3]

[1,2]National Taipei University of Technology, Taipei, Taiwan
[3]Institute for Information Industry, Taipei, Taiwan
[1]yfliao@ntut.edu.tw

## Abstract

This paper describes our first Deep Neural Network (DNN)-based speech synthesis system submitted to Blizzard Challenge 2018. The focus of this work is to explore the capacities of DNNs. Therefore, our system is based on latest HMM/DNN-based Speech Synthesis System (HTS) toolkits (ver. 2.3.2). Although, the performance of our system is not good enough on naturalness and similarity, its speech pause and stress scores of audiobook paragraphs were almost above the average. This should be a good starting point for further performance improvement.

**Index Terms**: speech synthesis, deep neural networks, HTS.

## 1. Introduction

This paper describes our DNN-based speech synthesis system submitted to Blizzard Challenge 2018 [1], the open evaluation that benchmarks the performance of different TTS systems with a common speech database. This system is our first DNN-based version. In the development phase, two different toolkits had been adopted and tested including HTS [2] and Merlin [3]. However, since we decided to attend this year's challenge very late, our submission was finished almost in only two weeks and only the HTS-based results were selected and been submitted.

However, since our original aim to participate this challenge is to learn and catch up the main stream DNN-based speech synthesis technologies. We will use our system as a good starting point and, especially, with this precious corpus, to figure out how to build a good TTS for audiobook applications in the future. This experience also helped us a lot on designing and recording a large-scale emotional Mandarin speech corpus for another audiobook related project.

In the following sections, the linguistic features extraction frontend, the voice building backend and finally the official evaluation results will be described in more detail.

## 2. System Configuration

### 2.1. Linguistic Features

The linguistic features and the corresponding question set used for clustering all the context-dependent phones is composed of 5 layers and listed in Table 1. Most of the features were extracted using the festival [4] toolkits.

It is worth noting that for our system, only linguistic features below utterance level were considered. And most of them are number and position-related cues, such as the number and forward and backward positions of phone, syllable, word and paragraph in a syllable, word, phrase and utterance, respectively. In this future, more supra-segmental information should be added. For example, emotional information, types of sentence, descriptive or conversational statements, etc. [5], are all necessary for better audiobook speech synthesis.

Table 1: *Hierarchical structure of linguistic features for DNN-based acoustic modeling.*

| Layer | Question |
|---|---|
| Phone | names and types of current and surrounding phones (5-gram); number and forward and backward position of a phone in a syllable |
| Syllable | number and forward/backward position of a syllable in a word; is it a stressed syllable or not |
| Word | part-of-speech (POS) of current and surrounding words; the number and forward and backward position of a word in a phrase |
| Phrase | number and forward and backward position of a phrase in an utterance |
| Utterance | number and forward and backward position of a paragraph |

### 2.2. Acoustic Features

Although, STRAIGHT [6], [7] is the most popular vocoder for speech synthesis research, its usage is restricted by non-commercial uses only license. Therefore, in this work, traditional 49-order mel-generalized cepstrum (MGC) [8] coefficients and one fundamental frequency, F0, extracted using the Robust Algorithm for Pitch Tracking (RAPT) [9] algorithm, were still selected as the spectral and excitation parameters.

The sampling rate of the Blizzard Challenge 2018 corpus is 44.1 kHz and the speech features were extracted every 5 ms. Moreover, the first and second order derivative features were also generated to form a 150-dimentional feature vector for each speech frame. On the other hand, to improve the quality of the synthesized speech, in the future, open source World [10] or MagPhase [11] vocoders will be adopted instead.

### 2.3. Acoustic Modeling

In this work, DNN-based acoustic models were trained to represent a mapping function from linguistic features to output acoustic features. A trajectory training criterion considering global variance (GV) was also applied. However, the mixture density network (MDN) [5], [12] approach was not yet implemented.

In all the following DNN models, three hidden layers that had 2048 neurons with rectified linear unit (RELU) activation functions plus the dropout and post-filter options were adopted.

But, in the future, recurrent neural networks, especially bi-directional long-short-term memory (LSTM) [13], [14]- or WaveNet [15], [16]-based acoustic models should be evaluated, since they are nowadays the state-of-the-art modeling approaches.

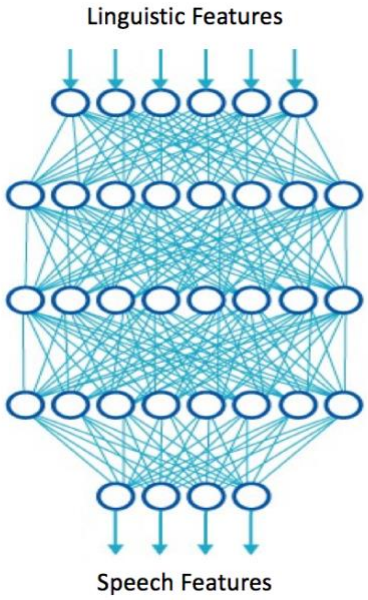


Figure 1: *DNN-based Spectral and excitation model.*

## 3. Training Procedure

The voice building steps of our system are showed in Fig. 2. Here the deterministic annealing expectation and maximization (DAEM) training algorithm was utilized to initialize the HMM-based acoustic modeling. The numbers of iterations for DAEM were experimentally set to 20.

For both the DNN-based approaches, Nvidia K80 GPUs and Tensorflow ver. 1.7 were applied to train the DNNs. By this setting, it usually requires about 2~3 days to complete the whole training procedures.

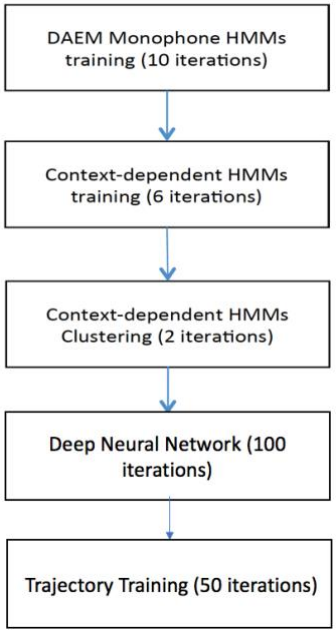


Figure 2: The flowchart of the voice building procedure using HTS version 2.3.2.

## 4. Parallel Processing

Since, the training procedure is somehow very time-consuming, some scripts and commands in HTS 2.3.2 recipe were modified in order to be executed in parallel using multiple CPUs or GPUs including (1) data preparation and (2) model training.

### 4.1. Data Preparation

First of the all, the data preparation script "data/Makefile.in" in HTS 2.3.2 was rewritten to allow the make command to execute many recipes simultaneously using the "-j" option (followed Alexis's work [17]).

This parallel "Makefile" recipe will extract all required data in about one and half hour on a 10-core/20-thread Intel Xeon E5-2650@2.30GHz machine. This modification is especially useful while running the STRAIGHT vocoder using MATLAB.

```
...
all: analysis labels
analysis: features stats
labels: lab gen mlf list scp question
...

f0/%.f0 ap/%.ap sp/%.sp: raw/%.raw
...
mgc/%.mgc lf0/%.lf0 bap/%.bap: raw/%.raw
...
cmp/%.cmp: mgc/%.mgc lf0/%.lf0 bap/%.bap
...
ffo/%.ffo: cmp/%.cmp
...

labels/full/%.lab: txt/%.txt
...
labels/mono/%.lab: labels/full/%.lab
...
labels/gen/%.lab: txt_gen/%.txt
...
...
```

Figure 3: *The major modification of the parallel "Makefile" recipe for data preparation.*

### 4.2. HMMM Model Training

Secondary, several HTS commands were run in parallel using a queuing system including (1) HCopy for feature extraction, (2) HHEd for triphone clustering, (3) HERest for HMM model training, HVite and HSMMAlign for recognition and forced-alignment.

These steps were done by writing several Perl scripts to submit parallel jobs to a Sun Grid Engine (SGE) queueing system. In our case, 5 workstations with in total 100 CPU cores were utilized to:

- Divide the list of data files
- Send the divided jobs to the processors
- Check for job completion
- Combine the results

The detail explanation of the integration of the Perl script and queuing mechanism could be found in [18].

On the other hand, for HHEd procedure, we took the advantage of the "JM" command of HHEd to modify the training scripts to run triphone state clustering in parallel (using Perl's multi-threading features [19] and multi-core CPUs environment). By this way, the computation time of HHEd procedure could be greatly reduced to about 1/5 (for 5-state phone models).

### 4.3. GPU Settings

The most computation intensive procedure in our system is on the final DNN training steps using the Tensorflow [20] toolkits. In Tensorflow's default settins, if no loading balance is programed by hand, all GPUs in a machine will be occupied but only one GPU will be really invoked.

The Tensorflow framework does support multiple and arbitrary CPUs and GPUs combination, but how to efficiently distribute the computation loading is difficult (at least for us). Therefore, in this work, a python module [21] was written to atomically assign only one free GPU to each experiment job by setting the "CUDA_VISIBLE_DEVICES" environment variable to the most suitable one. In this way, different GPU could be utilized by different job at the same time.

## 5. Official Evaluation Results

Since, the time we decided to attend this year's challenge very late, our submission was done in only two weeks. So, our system mainly followed the setting of the given demo recipe and almost no finetunes had been done. Moreover, the STRAIGHT vocoder was not applied due to some technical issues.

It is also worth noting that in this year challenge there is only one single task 2018-EH1: UK English Children's Audiobooks, and in all the following content, the results of our system are represented as letter "N".

### 5.1. Cons

First of all, the overall performance of our system is, of course, not good enough. Fig. 4 and 5 show the official evaluation results on naturalness and similarity reported by all listeners.

These results may come from that our system didn't utilize band aperiodicity-like features or the high-quality STRAIGHT vocoder. Since, the STRAIGHT vocoder usually could improve the MOS by absolute one score according to previous studies. However, it may also show that we don't yet fully understand the HTS and Merlin toolkits. Some system polishing works have to be done in the future.

### 5.2. Pros

It is interesting that our system's performance on the scores of speech pause and stress of audiobook paragraphs is almost above the average among all submissions. Fig. 6 and 7 show the official evaluation results on speech pause and stress of audiobook paragraphs reported by all listeners.

These results are quite surprised, since we had done nothing special to improve the qualities of speech pause and stress of audiobook paragraphs. Our system in fact just synthesized speech sentence-by-sentence and then directly concatenated all segments in a paragraph, chapter or the whole story into one speech waveform file using the sox [22] sound processing toolkits. Sometime, there were even no silence segment between two utterances.

Another interesting result is that our system's performance on speech intelligence was not too bad. Fig. 8 shows the official evaluation results, word error rate (WER in %), on speech intelligence reported by all listeners. This may be not too surprised. Although, the sound of our system is strange or unpleasant but its pronunciation is indeed clear enough and is not difficult to be understood.

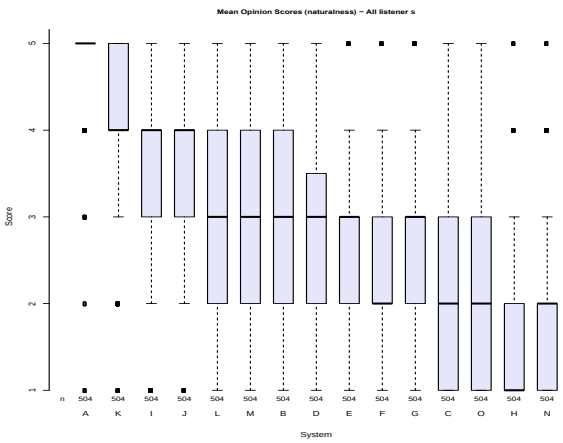


Figure 4: *Naturalness evaluation results (in MOS) reported by all listeners on 2018-EH1 task.*

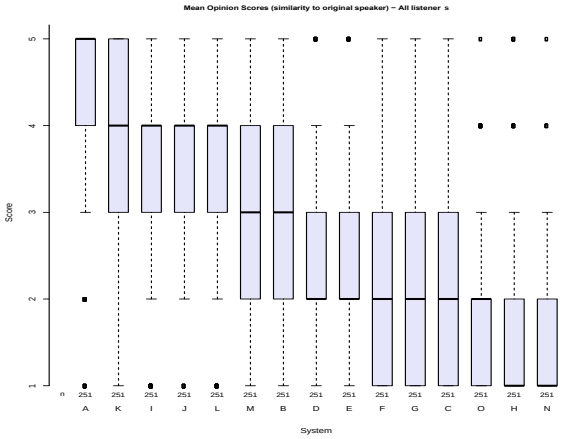


Figure 5: *Similarity evaluation results (in MOS) reported by all listeners on 2018-EH1 task.*

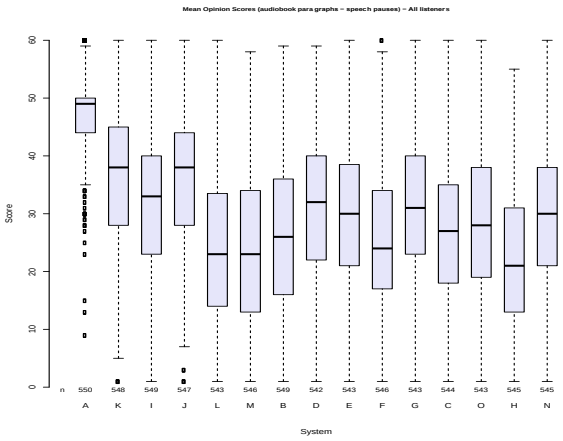


Figure 6: *Speech pauses of audiobook paragraphs evaluation results (in MOS) reported by all listeners on 2018-EH1 task.*

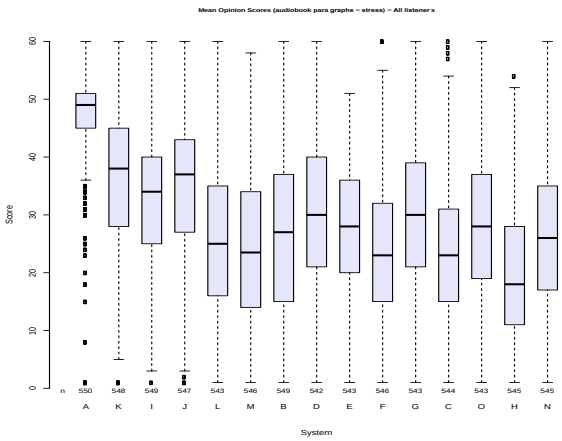


Figure 7. Stress of audiobook paragraphs evaluation results (in MOS) reported by all listeners on 2018-EH1 task.

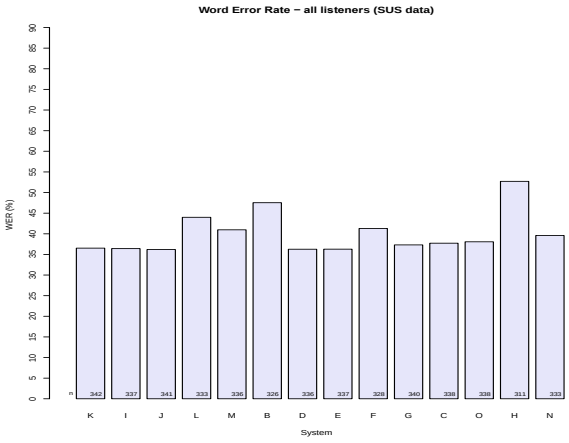


Figure 8: *Speech intelligence evaluation results (in word error rate, WER (%)) reported by all listeners on 2018-EH1 task.*

## 6. Conclusions

Since, this system is our first DNN-based version, its synthesized speech quality is not yet satisfied. However, its performance on speech pauses and stress scores of audiobook paragraphs is almost above the average. Therefore, this system will be a good starting point and will be continuing polished in the future. Especially, speech prosody will be considered to enhance its strength on emotion modeling.

## 7. Acknowledgements

This study is conducted under the "III system-of-systems driven emerging service business development Project" of the Institute for Information Industry which is subsidized by the Ministry of Economic Affairs of the Republic of China and is partially supported by the Ministry of Science and Technology of the Republic of China under the contract number 106-3011-F-027-006, 106-2221-E-027-128 and 107-2911-I-027-501.

## 8. References

[1] "Blizzard Challenge 2018 - SynSIG." [Online]. Available: https://www.synsig.org/index.php/Blizzard_Challenge_2018. [Accessed: 15-Jul-2018].

[2] "Home - HMM/DNN-based speech synthesis system (HTS)." [Online]. Available: http://hts.sp.nitech.ac.jp/. [Accessed: 15-Jul-2018].

[3] Z. Wu, O. Watts, and S. King, "Merlin : An Open Source Neural Network Speech Synthesis System," *9th ISCA Speech Synth. Work.*, pp. 218–223, 2016.

[4] University of Edinburgh, "Festival Speech Synthesis System." [Online]. Available: http://festvox.org/festival/. [Accessed: 17-Jul-2018].

[5] K. Sawada, C. Asai, K. Hashimoto, K. Oura, and K. Tokuda, "The NITech text-to-speech system for the Blizzard Challenge 2017," *Blizzard Chall. Work.*, pp. 1–6, 2017.

[6] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, "Tandem-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 3933–3936.

[7] H. Kawahara, "Exploration of the other aspect of Vocoder revisited," *ISCA Work. Speech Synth.*, pp. 32–37, 2010.

[8] S. Imai, T. Fukada, K. Tokuda, T. Kobayashi, S. Imai, and C. C. Llaguno, "Cepstral analysis synthesis on the mel frequency scale, and an adaptative algorithm for it.," *Synthesis (Stuttg).*, 2008.

[9] D. Talkin, W. B. Kleijn, and K. K. Paliwal, "A Robust Algorithm for Pitch Tracking (RAPT)," *Speech Coding and Synthesis, The Eds. Amsterdam, Netherlands:Elsevier*. pp. 495–518, 1995.

[10] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications *," no. 7, pp. 1877–1884, 2016.

[11] F. Espic, C. Valentini-Botinhao, and S. King, "Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2017–Augus, pp. 1383–1387, 2017.

[12] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, pp. 3844–3848, 2014.

[13] S. An, Z. Ling, and L. Dai, "Emotional statistical parametric speech synthesis using LSTM-RNNs," *Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC), 2017*, no. December, pp. 1613–1616, 2017.

[14] Z. Wu and S. King, "Investigating gated recurrent networks for speech synthesis," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2016–May, pp. 5140–5144, 2016.

[15] A. van den Oord *et al.*, "WaveNet: A Generative Model for Raw Audio," pp. 1–15, 2016.

[16] Y. Wang *et al.*, "Tacotron: Towards end-To-end speech synthesis," *Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH*, vol. 2017–Augus, no. May, pp. 4006–4010, 2017.

[17] Alexis Moinet, "parallel makefile for data in slt demo (2.3)." [Online]. Available: http://hts.sp.nitech.ac.jp/hts-users/spool/2016/msg00037.html. [Accessed: 16-Jul-2018].

[18] B. Lee, "Parallel Processing of the HTK Commands." [Online]. Available: http://www.ifp.illinois.edu/~bowonlee/research/cluster/HTK_parallel.htm. [Accessed: 15-Jul-2018].

[19] X. Yao and F. Ding, "Multi-process supporting patch for HTS-demo_CMU-ARCTIC-SLT." [Online]. Available: http://hts.sp.nitech.ac.jp/hts-users/spool/2008/msg00415.html. [Accessed: 16-Jul-2018].

[20] "TensorFlow." [Online]. Available: https://www.tensorflow.org/. [Accessed: 17-Jul-2018].

[21] Y. Bulatov, "Tensorflow on shared GPUs: how to automatically select the one that is unused - Stack Overflow," *Stack Overflow*, 2017. [Online]. Available: https://stackoverflow.com/questions/41634674/tensorflow-on-shared-gpus-how-to-automatically-select-the-one-that-is-unused. [Accessed: 16-Jul-2018].

[22] "SoX - Sound eXchange | HomePage." [Online]. Available: http://sox.sourceforge.net/. [Accessed: 16-Jul-2018].