The End-to-End Speech Synthesis System for the VLSP Campaign 2019

Quang Pham Huu
R&D Lab, Sun* Inc
pham.huu.quang@sun-asterisk.com

***Abstract*—The traditional speech synthesis systems are typically built by multiple components, such as including a text analysis front-end, an acoustic model and an audio synthesis module. Building these components often requires a lot of people possessing extensive domain experts and may contain brittle design choices. In this paper, we describe how we build a Vietnamese speech synthesis system (TTS) based on Deep Learning techniques. We completed the build of two speech synthesis systems, with BigCorpus (Mean Opinion Score of 3.47) and SmallCorpus (Mean Opinion Score of 4.13) in text-to-speech shared-tasks of VLSP 2019. In addition, transfer learning and fine-tuning techniques are also applied to solve noise data problems of training data in BigCorpus and shortage of data in SmallCorpus.**

***Index Terms*—Text-to-Speech, Vietnamese speech synthesis, Deep learning, Tacotron, Tacotron-2.**

I. Introduction

The speech synthesis is not a new problem, but it is still one of the challenges for organizations and businesses. One of the four shared-tasks of The sixth International Workshop on Vietnamese Language and Speech Processing (VLSP 2019), TTS shared-task requires participators to build a speech synthesis system with voices of two speakers that VLSP organizers provided. BigCorpus includes 22 hours of recording and SmallCorpus includes about 1 hour of recording. The biggest challenge for participants is to solve this shared-task with a large amount of noise data in BigCorpus and lack of training data with SmallCorpus. In this paper, we describe the speech synthesis system which we participated in the TTS shared-task of the 2019 VLSP evaluation campaign.

The rest of the paper is organized as follows: Section 2 presents a brief review of related work. In section 3, the pre-processing data handler and the model architecture used will be discussed. Experimental set-ups and results are presented in Section 4. Finally, we conclude the paper in Section 5.

II. Related work

Text-to-speech is not a new field for researchers [2]. Currently, there are several different methods to synthesize speech, and each method belongs to one of the following categories: articulator synthesis [1], formant synthesis [3], and concatenation synthesis [4], etc.

In [9], Yuxuan Wang *et. al.* proposed a deep learning architecture to the problem of speech synthesis, which is called Tacotron model. Thereafter, Tacotron-2 is proposed by Jonathan Shen *et. al.* in [6]. Tacotron-2 is currently state-of-the-art in this research field.

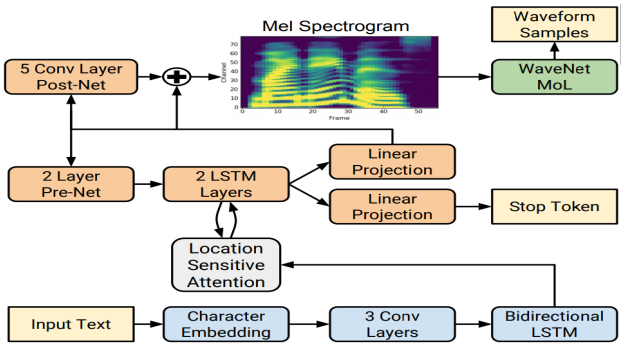

Figure 1. Block diagram of the Tacotron-2 system architecture with two component [6]. (1) a recurrent sequence-to-sequence feature prediction network with attention which predicts a sequence of mel spectrogram frames from an input character sequence, and (2) a modified version of WaveGlow which generates time-domain waveform samples conditioned on the predicted mel spectrogram frames.

III. Our Approach and Implementation

A. Model Architecture

To complete this challenge, we used the Tacotron-2 architecture with some changes to suit the problem-solving for Vietnamese. We used and inherited a open source repository that publishing in Github[1].

The first part of the Tacotron-2 model is responsible for converting texts into mel-spectrograms and these spectrograms are fed in a WaveGlow [5] model to produce audio waveforms. More specially, these two parts of the Tacotron architecture (Seq2Seq [7] and WaveGlow vocoder) can be trained independently. In this challenge, we only train Seq2Seq model, and WaveGlow will uses trained weights from the pre-trained model beforehand [2]. In the Seq2Seq model, the first part is an Encoder converting the character sequences into word embedding vectors, then these embedding vectors are fed into the Decoder for predicting spectrograms.

Tacotron-2 architecture is demonstrated in Figure 1. The details of the model are mentioned at [6].

B. Text Pre-processing

It is apparently true that text normalization is an important task in Text-to-speech system, especially in Vietnamese, an extremely complex language. In Vietnamese, a word can be

[1]https://github.com/NVIDIA/tacotron2
[2]https://github.com/NVIDIA/waveglow

pronounced in many different ways depending on the its context as we can not pronoun these words by applying letter-to-sound rules. For example, "thứ 4" ("wednesday") is spoken "thứ tư" ("wednesday") but "số 4" ("four") is spoken "số bốn" (not spoken "số tư") ("four"). The pronunciation depends on the context of word. Another problem is the form of abbreviations, such as HN ("Hà Nội") ("Hanoi"), clb ("câu lạc bộ") ("the club"). Furthermore, we do not have any rules to pronoun a word which is in abbreviation form. To make the speech synthesis become more useful for general tasks, we defined a list of regression rules for some kind of special words such date, time, score, currency and measurement,etc.., and created a dictionary of word abbreviation. To enhance the performance, we would also define a list of foreign words and normalize them by constructing Vietnamese phonetic transcription, and the module is implemented and based on regular expression.

We using this module to normalization for all text in training data (BigCorpus and SmallCorpus).

C. Audio Cleaning and Normalization

We explored that, in the BigCorpus dataset, there are quite a lot of audio whose quality are not qualified to feed into TTS system. Thus, we have to investigate, revise and manually select 1200 audio that was quality enough to be included in the fine-tuning process. With each over 12 seconds audio, it would be truncated and the text label is also adjusted accordingly. Finally, audio is converted to the sample rate of 22050Hz.

IV. Experimental Setup and Training

A. Experimental Setup

After the data has been cleaned and ready for training, 90%, 10% of the dataset are split and used in the training and validating phase, respectively. The input text data of systems comprise both alphanumeric characters and punctuation in Vietnamese, and adding punctuation helps the model learn how to beat out between clauses in sentences correctly.

Specifically, the key techniques in our training strategy are transfer learning [8] and fine-tuning techniques. Firstly, all data from the BigCorpus was put into training phase. After 35.000 steps of training, we proceed to save the model's weight; also load weight from trained model then freeze Encoder model in Tacotron architecture and only train on the 1200 audio data set prepared in the audio cleaning step and training until convergence.

As mentioned above, the Encoder of Seq2Seq model in Tacotron architecture is an embedding model that converts the character sequences into word embedding vectors. We had to put all the training data to the first training phase so that we could have a quality embedding model, which would allow the system to generate the sound of any word in Vietnamese. After 35.000 steps of training, we realized that our model was able to generate the speech of Vietnamese words correctly even though they were extremely noisy. This proves that our Encoder model is of good enough quality and it needs to be frozen so that the Tacotron architecture can learn on a smaller and cleaner data set.

Similarly, for the SmallCorpus dataset, we also load weights from BigCorpus trained checkpoint, freeze Encoder model in Tacotron architecture and training until convergence.

After synthesizing speech from the model, we use Noisereduce[3] library to reduce noise in the audio. This has helped us reduce noise in the audio, but it also has a negative impact on our synthesized speech.

Our experiment is conducted on a computer with Intel Core i5-7500 CPU @3.40GHz 4 cores, 32GB of RAM, GPU GeForce GTX 1080 Ti, 1TB SSD Harddisk, and the operating system is Ubuntu 16.04 64 bit. We use Pytorch framework version 1.1.0.

Finally, in training, our model, batch size of 32 with learning rate of 0.001 is trained in total of 36 hours with BigCorpus and 12 hours with SmallCorpus.

Table I
THE SUBJECTIVE RESULT OF SPEECH SYNTHESIS SYSTEM

Dataset	**Sample tests**	MOS Average score	NATURAL
BigCorpus	44	3.47	4.32
SmallCorpus1	80	4.13	4.57

B. Evaluation Metrics and Results

To evaluate the quality of a speech synthesis system, we used mean opinion score (MOS) tests, where the subjects were asked to rate the naturalness of the speech audio in a 5-point scale score. The evaluation of both systems is executed by 24 native Vietnamese listeners, who evaluated the naturalness and intelligibility of each system. The final score is calculated by averaging the evaluation score of all listeners.

The subjective results presented in Table 1. This table shows the result of evaluation of our speech synthesis system with BigCorpus and SmallCorpus. As can be shown that our speech synthesis system has MOS of 3.47 in BigCorpus and MOS of 4.13 in SmallCorpus.

V. Conclusion

In this paper, an end-to-end technique in speech synthesis is implemented. Two Vietnamese speech synthesis model using Tacotron-2 are trained with BigCorpus (22 hours) and SmallCorpus (1 hour) dataset. The output of both models is high-quality speech audio. We hope this system can provide the best speech synthesis system for Vietnamese to produce high quality of voice from text. In future work, we want to improve the performance of our system by improve quality of training data and build custom model architecture.

Acknowledgement

This work is partially supported by ***Sun* Inc***. We would also thank our colleagues at ***Sun* Inc*** for their advice and expertise. Without their support, this experiment would not have been accomplished.

[3]https://pypi.org/project/noisereduce/

We would like to thank the VLSP organizers for their hard work in maintaining the conference and providing datasets for this evaluation campaign.

REFERENCES

[1] A. D. Girson and R. D. Williams. Articulator-based synthesis for conversational speech. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 873–876 vol.2, April 1990.

[2] Ramita Karpe. A survey :on text to speech synthesis. *International Journal for Research in Applied Science and Engineering Technology*, 6:351–355, 03 2018.

[3] S. Lukose and S. S. Upadhya. Text to speech synthesizer-formant synthesis. In *2017 International Conference on Nascent Technologies in Engineering (ICNTE)*, pages 1–4, Jan 2017.

[4] I. Y. Ozum and M. M. Bulut. Knowledge based speech synthesis by concatenation of phoneme samples. In *Proceedings of MELECON '94. Mediterranean Electrotechnical Conference*, pages 51–54 vol.1, April 1994.

[5] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. *CoRR*, abs/1811.00002, 2018.

[6] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. *CoRR*, abs/1712.05884, 2017.

[7] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

[8] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. *CoRR*, abs/1808.01974, 2018.

[9] Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: A fully end-to-end text-to-speech synthesis model. *CoRR*, abs/1703.10135, 2017.