VNU Journal of Science: Comp. Science & Com. Eng, Vol. 38, No. 1 (2022) 47-53

VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

# TTS - VLSP 2021: The NAVI's Text-To-Speech System for Vietnamese

Le Minh Nguyen*, Do Quoc An, Vu Quoc Viet, Vo Thuc Khanh Huyen

*Hanoi University of Science and Technology, 1 Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam*

Received 27 December 2021
Revised 5 April 2022; Accepted 5 May 2022

**Abstract:** The Association for Vietnamese Language and Speech Processing (VLSP) has organized a series of workshops intending to bring together researchers and professionals working in NLP and attempt a synthesis of research in the Vietnamese language. One of the shared tasks held at the eighth workshop is TTS [14] using a dataset that only consists of spontaneous audio. This poses a challenge for current TTS models since they only perform well constructing reading-style speech (e.g, audiobook). Not only that, the quality of the audio provided by the dataset has a huge impact on the performance of the model. Specifically, samples with noisy backgrounds or with multiple voices speaking at the same time will deteriorate the performance of our model. In this paper, we describe our approach to tackle this problem: we first preprocess the training data then use it to train a FastSpeech2 [10] acoustic model with some replacements in the external aligner model, finally we use HiFiGAN [4] vocoder to construct the waveform. According to the official evaluation of VLSP 2021 competition in the TTS task, our approach achieves 3.729 in-domain MOS, 3.557 out-of-domain MOS, and 79.70% SUS score. Audio samples are available at https://navi-tts.github.io/.

*Keywords:* VLSP-2021, Spontaneous, Text-to-speech.

## 1. Introduction[1]

Text to speech (TTS) is a system aiming to synthesize intelligible and natural audios which are indistinguishable from human recordings. With the rapid progress of deep learning approaches in recent years, a lot of acoustic end-to-end models are proposed like Tacotron2 [11], FastSpeech2 [10], TransformerTTS [6] ... and achieve great results. With regards to Vietnamese speech synthesis, for VLSP share-task in both 2019 and 2020, Tacotron2 was used as acoustic model [1] [17], which proves this model's effectiveness. Moreover, progress has also been made to lower-level features, such as prosodic boundary [15] and text normalization [13].

However, the effectiveness of these models on the spontaneous datasets is not guaranteed. Experiments conducted in the paper proposing

* Corresponding author.
*E-mail address:* nguyen.lm162998@sis.hust.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.347

these models usually use clean, studio-quality datasets. For example, Tacotron2 [11] was trained on speech spoken by a professional female speaker, FastSpeech [19] and FastSpeech2 [10] were trained on LJSpeech [2] dataset, containing speech of audiobooks provided by LibriVox project. Spontaneous speech datasets pose a huge challenge for current text-to-speech models such as the existence of pauses, such as um and uh, and the diversity of rhythm.

Despite its difficulties, enabling the model to train on spontaneous data is crucial because, with it, synthesized speech sounds natural and is better suited for conversation applications. However, creating such a dataset in studio-level quality also comes with obstacles. It requires too much time and effort from speakers due to the absence of a script. For that reason, it is necessary to utilize existing spontaneous speech.

The TTS shared task [14] organized at the eighth workshop of the Association for Vietnamese Language and Speech Processing (VLSP) requires participants to create a Vietnamese TTS system able to synthesize natural sounding audios while having trained on a spontaneous and noisy dataset. To be precise, this year's dataset for TTS uses speech crawled from videos of a female Hanoi YouTuber named "Giang oi". Even after preprocessing and cleaning, there still exist four problems that need to be addressed: First is the existence of pauses or hesitations, such as um and uh, alongside the existence of background noise coming from a less-than-ideal recording environment. The second one is that the audio samples provided sometimes contain voices different from the main speaker, such as her friends or family members. The third problem is the large variety of intensity, stress, and prosody across the dataset. The final challenge is the incorrectness of the transcript despite having been validated by humans.

To cope with the problems stated above, in this VLSP 2021 challenge, we designed an end-to-end TTS system having 3 main components as follows:

- A sophisticated pre-processing pipeline with multiple steps like separating noise from audio, removing audio samples of different speaker or of multiple speakers, filtering audios which have good, comprehensible speaking style and normalizing volume.
- The Fastspeech [10] acoustic component with the replacement of MFA external aligner module by TransfomerTTS [6] - an auto-regressive model. We convert the words into phonemes by a grapheme-to-phoneme module and only use MFA for predicting the silences between phonemes as well trimming the audios, from that we got the data to train the aligner model
- The HiFiGAN [4] vocoder that generates a 22kHz waveform corresponding with the predicted mel-spectrogram of the acoustic model. After that, the waveform is fed to a model called HiFiGAN denoiser to make the audio sound a little bit cleaner and more natural.

According the official evaluation of VLSP 2021 in the TTS task, our proposed system achieves 3.7295 in-domain MOS, 3.557 out-of-domain MOS (the baseline is 5) and 79.70% SUS score. Moreover, with that results, we ranked first among other participants, which demonstrates the effectiveness of our system in tackling the noisy dataset of spontaneous speech. We attach audio samples generated by our system at https://navi-tts.github.io/.

## 2. Data Pre-processing

This section will go in-depth into our Data Pre-processing pipeline. We will also discuss the reasons why we use the techniques mentioned.

### *2.1. Grapheme to Phoneme*

The Grapheme to Phoneme (G2P) conversion is the process that generates the phoneme sequence (pronunciation) according to the grapheme sequence (word). It is considered essential in several tasks such as in text-to-speech (TTS) and automatic speech recognition (ASR) systems. In this work, we convert the grapheme sequence into the phoneme sequence

using a public tool for Vietnamese [16] with the implementation Viphoneme.

### *2.2. Noise Reduction*

The first step of our pre-processing pipeline is noise reduction. Since the audio samples of the dataset contain noises from a poor recording environment, such as noise of cooking equipment or noise coming from a nearby neighborhood, reducing this type of sound will also make generated speech sounds clearer. We do this by using the music source separation tool [5] developed by ByteDance AI Lab. We use model = MobileNet_Subbandtime with input_channels = 2, output_channels=2 and sample_rate=44100. The task addressed by this model is to separate audio recordings into multiple sources. For example, given an audio recording with a singer's voice, drum sound, and "accompaniment" sound, the model's objective is to separate these sounds into 3 distinct audio files. In our case, we only care about the voice audio file.

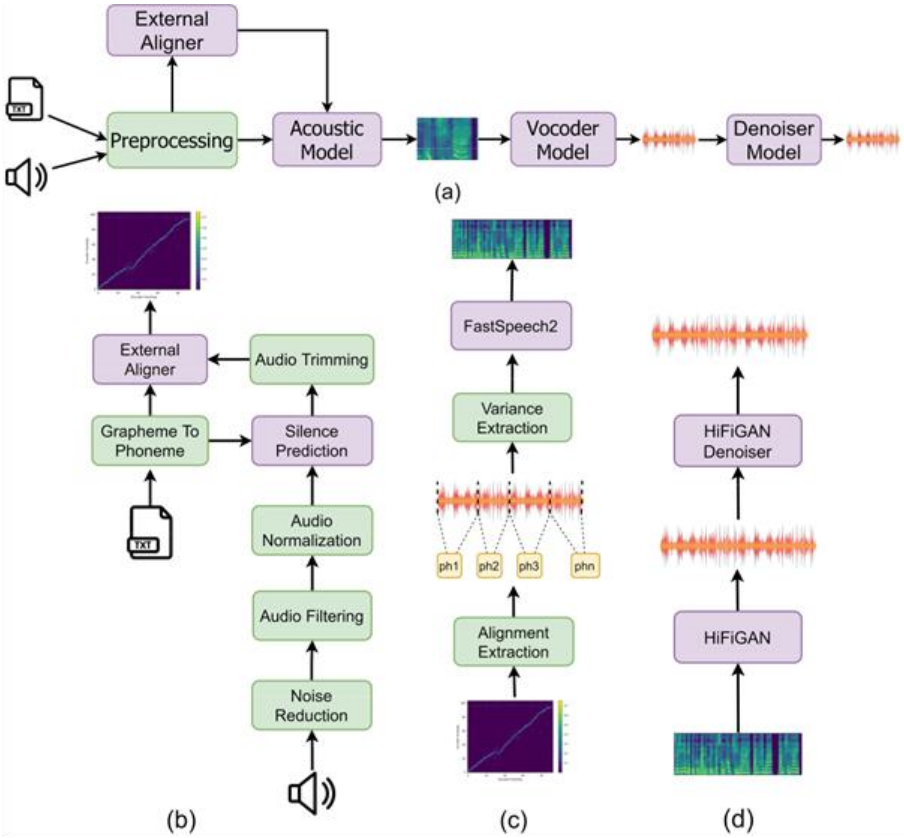


Figure 1. Overview of our proposed system. Figure (a) shows overall pipeline of our system. Figure (b) shows the detailed pipeline of preprocessing component. Figure (c) shows overall modules of acoustic model. Figure (d) show overall modules of vocoder model.

### *2.3. Audio Filtering*

This part of the pre-processing pipeline aims to solve the problems regarding voices from other speakers and also to remove recordings with abnormal speaking styles. We choose 5 audio samples that have the main speaker's voice. After that, we use a pre-trained [9], which is trained on Voxceleb 1 [7] and Voxceleb 2 [8] datasets, to embed all audio files then compare them to the embedding of reference recording. To compare the

embedding vectors, we use cosine similarity. Samples that are vastly different from these references, i.e average cosine similarity smaller than 0, will be assumed to be spoken by another speaker or multiple speakers simultaneously. We also check recordings with average cosine similarity larger than 0 but smaller than 0.25 manually to make sure that we do not miss any faulty data.

### *2.4. Audio Normalization and Punctuation Prediction*

After the filtering step, we got acceptable audios but they had different volume levels, so we conducted normalizing the volume of audios to -3dB level by using peak normalization with ffmpeg [12] tool. Moreover, in the given data set there were no sentences which have punctuation, this is a very bad problem because punctuation is critical for the acoustic model to learn to predict the silences in the output audio. To address this problem, we used a tool called Montreal Forced Aligner (MFA) – an open-source system for learning the alignment between text-audio pairs without any manual alignment annotations. We train MFA on the cleaned data set and process the result to add the silences into the data set. Based on the alignment results, the predicted silences which have a duration greater than 0.15s we will regard as punctuation, and from that, we also trim the audios which have silence at the start or the end.

## 3. Model Architecture

There are 2 main components to our system: The acoustic model and the Vocoder. We use FastSpeech2 [10] for our Acoustic model and HiFiGAN [4] Vocoder combined with a HiFiGAN Denoiser.

### *3.1. Acoustic Model*

In this competition we use Fastspeech2 [10] as the acoustic model. Fastspeech2 is a non-auto-regressive model which processes the information in parallel by using a feed-forward structure including multiple Feed-Forward Transformer [18] stacks. Since we did not make any modifications to the architecture of the network, the number of parameters does not change.

In the original paper, the authors used the MFA tool (as mentioned in 2.4) as an external aligner to predict the duration of each phoneme from training data then conducted extracting the acoustic information from that (pitch, energy, etc). The inputs of this model are text-audio pairs with the text has been removed punctuation and a lexicon file which includes words and the corresponding phonemes line by line. We tried using the same aligner model as the original paper but the synthesized audios were noisy and robotic. We suspect that the MFA tool is not efficient in learning the alignment for spontaneous data set. With that assumption in mind, we tried replacing the MFA tool with an auto-regressive model – Transfomer TTS [6] with the implementation as-ideas TransformerTTS - as an external aligner model with the same training configuration, the results from MFA are still utilized but only for adding the silences into the training phoneme sequences (as mentioned in 2.4) before training the main aligner (as shown in Figure 1c). After training the TransformerTTS [6] aligner, we processed the alignment matrix to get the predicted duration for each phoneme and conducted extracting the acoustic information based on that duration. The extracted phoneme duration is an integer number which is the number of mel-spectrograms attended to that phoneme in the mel-scale. For example phoneme sequence [x, I, n, −, c, h, a, o] has the corresponding duration sequence is [5, 3, 2, 7, 4, 5, 5, 2]. Next, we used pyworld tool with dio algorithm and stonemask refinement which has been implemented in place to extract pitch and energy based on the predicted phoneme duration. We also converted the raw waveform into 80 channels mel-spectrograms following [3] to get enough inputs for training Fastspeech2 [10]. The

training inputs for Fastspeech2 [10] includes phoneme duration, extracted energy, extracted pitch and mel-spectrogram. We trained it up to 600K steps with batch_size = 16, initial learning_rate = 0.001 with anneal_rate = 0.3 and anneal_steps = [300K, 400K, 500K]. The idea of replacing MFA with TransformerTTS [6] in our architecture was motivated by Fastspeech1 [19] and we saw it give us good results for the spontaneous dataset in this challenge.

### *3.2. Vocoder*

To achieve better vocoding quality and higher efficiency, we utilize a HiFiGAN[4] vocoder. Our network architecture is similar to config V1. A mel-spectrogram is used as input of the generator and upsampled through transposed convolutions until the length of the output sequence matches the temporal resolution of a raw waveform. After that, to reduce noise from the output waveform we pass the audio through HiFiGAN denoiser model (as shown in Figure 1d).

Table 1. Statistics of original dataset:

| No. | Type | Value | Unit |
|---|---|---|---|
| 1 | Total Clips | 5,341 | clips |
| 2 | Total Words | 105,209 | words |
| 3 | Total Characters | 330,849 | characters |
| 4 | Total Duration | 7.25 | hours |
| 5 | Mean Duration | 4.88 | seconds |
| 6 | Min Clip Duration | 0.19 | seconds |
| 7 | Max Clip Duration | 125.77 | seconds |
| 8 | Mean Words per Clip | 19.70 | words/clip |
| 9 | Distinct Words | 3,935 | words |

Table 2. Statistics of our final dataset:

| No. | Type | Value | Unit |
|---|---|---|---|
| 1 | Total Clips | 4,614 | clips |
| 2 | Total Words | 90,407 | words |
| 3 | Total Characters | 278,902 | characters |
| 4 | Total Duration | 5.85 | hours |
| 5 | Mean Duration | 4.57 | seconds |
| 6 | Min Clip Duration | 0.58 | seconds |
| 7 | Max Clip Duration | 29.52 | seconds |
| 8 | Mean Words per Clip | 19.69 | words/clip |
| 9 | Distinct Words | 2,836 | words |

## 4. Experiment

Within three weeks of this contest, we spent half cleaning data and the remaining for training the model. We found that the preprocessing phase in this challenge plays a critical role in our system, enabling the model to converge. After many experiments and modifications, we found the suitable architecture for the spontaneous dataset of VLSP 2021 TTS challenge. We trained acoustic model Fastspeech2 until 600K steps and vocoder model HiFiGAN until 1M steps within one week on one GPU Tesla V100 32GB. The statistics of the dataset before and after our preprocessing steps are shown in table 1 and table 2 respectively.

Our final training corpus can be found at: navi-tts-vlsp2021-final-training-corpus

Table 3. TTS scores of all participants:

| Name | MOS-OD | MOS-ID | SUS |
|---|---|---|---|
| Team 1 | 3.56 | 3.73 | 0.20 |
| Team 2 | 2.81 | 3.27 | 0.25 |
| Team 3 | 3.52 | 3.81 | 0.22 |
| Team 4 | 2.66 | 3.79 | 0.32 |
| Team 5 | 3.37 | 3.98 | 0.16 |
| Team 6 | 3.30 | 3.94 | 0.15 |

## 5. Results

The evaluation of the TTS shared task in VLSP 2021 challenge includes three aspects: naturalness test with MOS (Mean Opinion Score) for in-domain and out-domain scope, SUS (Semantically Unpredictable Sentences) test with WER (Word Error Rate). In the MOS part, listeners hear audio samples and for each sample, they have to choose a score on a scale from 1 to 5 to answer the question "How do you rate the naturalness of the sound you have just heard?". In the SUS test, the listeners hear an utterance and type in what they heard, the

result will be evaluated based on Word Error Rate (WER). Table 3 presents the scores of our system (Team 1) and other participants.

The results suggest that although our model does not significantly outperform those of other teams on all metrics, it achieves the best result for the out-domain MOS score. The average out-domain MOS of all models is 3.02, which is 0.54 points lower than our result. With regards to the other 2 metrics, our results are very close to those of the best performing models. Concretely, in terms of the in-domain MOS and SUS tests, our scores are only 0.25 points and 0.05 points lower than the highest scores, respectively.

## 6. Summary

In summary, in this work, we showed our method in preprocessing the data as well as the models used. Our preprocess approach includes multiple steps of filtering audio data and the models used are Fastspeech acoustic model with some modifications and HiFiGAN vocoder. Our approach achieved 3.729 in-domain MOS, 3.557 out-of-domain MOS and 79.70% SUS score for the dataset given in this contest which has proven effective in our system in synthesizing the high quality audios on condition that the data set was spontaneous and not clean.

However, there are improvements to be made. For future work, we will explore ways to properly exploit audio samples that are too short or too long, as well as experiment with more acoustic model architectures, especially ones that are created to tackle the challenge of spontaneous dataset.

## References

[1] Q. P. Huu, D. Lab, The End-To-End Speech Synthesis System For The Vlsp Campaign, 2019, Pp. 3.

[2] K. Ito, L. Johnson, The Lj Speech Dataset. Https://Keithito. Com/Lj-Speech-Dataset/, 2017.

[3] R. J. W. M. Schuster, N. Jaitly, Z. Y. Zhifeng, C. Y. Zhang, Y. Wang Shen, R Pang, Natural Tts Synthesis By Conditioning Wavenet On Mel Spectrogram Predictions, 2018.

[4] J.L Kong, J. Kim, J. Bae. Hifi-Gan: Generative Adversarial Networks For Efficient And High Fidelity Speech Synthesis, 2020.

[5] Q. Kong, Y. Cao, H. Liu, K. Choi, Y. Wang, Decoupling Magnitude And Phase Estimation With Deep Resunet For Music Source Separation. In Ismir. Citeseer, 2021.

[6] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, M. Zhou. Neural Speech Synthesis With Transformer Network, 2019.

[7] A. Nagrani, J. S. Chung, A. Zisserman. Voxceleb: A Large-Scale Speaker Identification Dataset, 2017.

[8] A. Nagrani, J. S. Chung, A. Zisserman. Voxceleb2: Deep Speaker Recognition, 2018.

[9] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J .C. Chou, S. L. Yeh, S. W. Fu, , Speechbrain: A General-Purpose Speech Toolkit, 2021. Arxiv:2106.04624.

[10] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, And Tie-Yan Liu, Fastspeech 2: Fast And High-Quality End-To-End Text To Speech, 2021.

[11] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. J. S.-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, Y. Wu, Natural Tts Synthesis By Conditioning Wavenet On Mel Spectrogram Predictions. Corr, 2017.

[12] S, Tomar, Converting Video Formats With Ffmpeg. Linux Journal, Vol. 146, 2006, Pp.10.

[13] N. T. T. Trang, D. X. Bach, N. X. Tung, A Hybrid Method For Vietnamese Text Normalization. In Proceedings Of The 2019 3rd International Conference On Natural Language Processing And Information Retrieval, Nlpir 2019, Pages 104–109, New York, Ny, Usa, June 2019. Association For Computing Machinery.

[14] N. T. T. Trang, N. H. Ky, Vlsp 2021 - Tts Challenge: Vietnamese Spontaneous Speech Synthesis. Vnu Journal Of Science: Computer Science And Communication Engineering, Vol. 38, No. 1, 2022.

[15] N. T. T Trang, N. H. Ky, A. Rilliard, C. D'alessandro, Prosodic Boundary Prediction Model For Vietnamese Text-To-Speech. In Interspeech 2021, Pages 3885–3889. Isca,

August 2021.

[16] N. M. Tri, C. X. Nam, Vietnamese Speech Synthesis With End-To-End Model And Text Normalization. In 2020 7th Nafosted Conference On Information And Computer Science (Nics), Pp. 179–184.

[17] K. D. Trieu, B. Q. Dam, Q. B. Nguyen. Development Of Smartcall Vietnamese Text-To-Speech For Vlsp 2020. In Proceedings Of The 7th International Workshop On Vietnamese Language And Speech Processing, 2020, Pp. 24–29,

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin,, Attention Is All You Need. Arxiv:1706.03762 [Cs], December 2017. Arxiv: 1706.03762.

[19] X. Tan, T. Qin, S. Zhao, Z. Zhao, T. Y. Liu, Y. Ren, Y. Ruan, Fastspeech: Fast, Robust And Controllable Text To Speech, 2019.