BỘ GIÁO DỤC
VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

PHẠM NGỌC PHƯƠNG

DANH MỤC CÔNG TRÌNH CÔNG BỐ

NGHIÊN CỨU PHÁT TRIỂN HỆ THỐNG THÍCH NGHI GIỌNG NÓI TRONG TỔNG HỢP TIẾNG VIỆT VÀ ỨNG DỤNG

LUẬN ÁN TIẾN SĨ NGÀNH HỆ THỐNG THÔNG TIN

Mã số: 9 48 01 04

Hà Nội, 2023

DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ CỦA TÁC GIẢ

1. Pham Ngoc Phuong, Tran Quang Chung, Luong Chi Mai: “Adapt-TTS: High-quality zero-shot multi-speaker text-to-speech adaptive-based for Vietnamese”. *Journal of Computer Science and Cybernetics,* V.39, N.2 (2023), pp. 159-173. 1-DOI: 10.15625/1813-9663/18136, VietNam.
2. Pham Ngoc Phuong, Tran Quang Chung, Luong Chi Mai: “Improving few-shot multi-speaker text-to-speech adaptive-based with Extracting Mel-vector (EMV) for Vietnamese”. *International Journal of Asian Language Processing*, 2023, Vol. 32, No. 02n03, 2350004, pp. 1-15, Singapore.
3. Pham Ngoc Phuong, Tran Quang Chung, Do Quoc Truong, Luong Chi Mai: “A study on neural-network-based Text-to-Speech adaptation techniques for Vietnamese”, *International Conference on Speech Database and Assessments (Oriental COCOSDA) 2021*, pp. 199-205. IEEE, Singapore.
4. Pham Ngoc Phuong, Tran Quang Chung, Nguyen Quang Minh, Do Quoc Truong, Luong Chi Mai: “Improving prosodic phrasing of Vietnamese text-to-speech systems”, *Association for Computational Linguistics, 7th International Workshop on Vietnamese Language and Speech Processing*, 12/2020, pp. 19-23, VietNam.
5. Nguyen Thai Binh, Nguyen Vu Bao Hung, Nguyen Thi Thu Hien, Pham Ngoc Phuong, Nguyen The Loc, Do Quoc Truong, Luong Chi Mai: “Fast and Accurate Capitalization and Punctuation for Automatic Speech Recognition Using Transformer and Chunk Merging”, *International Conference on Speech Database and Assessments (Oriental COCOSDA)* 2019, IEEE, pp. 1-5, Philippines.
6. Pham Ngoc Phuong, Do Quoc Truong, Luong Chi Mai: "A high quality and phonetic balanced speech corpus for Vietnamese" *International Conference on Speech Database and Assessments (Oriental COCOSDA)* 2018, pp. 1-5 Japan.
7. Tác giả Bảo hộ quyền sở hữu trí tuệ “Phần mềm chuyển đổi văn bản thành giọng nói Adapt-TTS “số 7590/QTG ngày 26/9/2022 tại *Cục Bản quyền tác giả.*

Journal of Computer Science and Cybernetics, V.39, N.2 (2023), 159–173
DOI no 10.15625/1813-9663/18136

ADAPT-TTS: HIGH-QUALITY ZERO-SHOT MULTI-SPEAKER TEXT-TO-SPEECH ADAPTIVE-BASED FOR VIETNAMESE

PHUONG PHAM NGOC[1,2], CHUNG TRAN QUANG[2,3], MAI LUONG CHI[4,*]

[1] *Thai Nguyen University, Vietnam*
[2] *AIMed Artificial Intelligence Solution, Vietnam*
[3] *Japan Advanced Institute of Science and Technology (JAIST), Japan*
[4] *Institute of Information Technology, Vietnam Academy of Science and Technology, Vietnam*

Abstract. Current adaptive-based speech synthesis techniques are based on two main streams: 1) Fine-tuning the model using small amounts of adaptive data; 2) Conditionally training the entire model through a speaker embedding of the target speaker. However, both of these methods require adaptive data to appear during training, which makes the training cost to generate new voices quite expensive. In addition, the traditional text to speech (TTS) model uses a simple loss function to reproduce the acoustic features. However, this optimization is based on incorrect distribution assumptions leading to noisy composite audio results. In this paper, we propose the Adapt-TTS model that allows high-quality audio synthesis from a small adaptive sample without training to solve these problems. The main contributions of the paper are: 1) The extracting mel-vector (EMV) architecture allows for a better representation of speaker characteristics and speech style; 2) An improved zero-shot model with a denoising diffusion model (mel-spectrogram denoiser) component allows for new voice synthesis without training with better quality (less noise). The evaluation results have proven the model's effectiveness when only needing a single utterance (1-3 seconds) of the reference speaker, the synthesis system gave high-quality synthesis results and achieved high similarity.

Keywords. Zero-shot TTS; Multi-speaker; Text-to-speech; Diffusion models; Mel-spectrogram denoiser; Extracting mel-vector; EMV, Adapt-TTS.

1. INTRODUCTION

Currently, speech synthesis techniques (TTS text-to-speech) based on neural networks have achieved the same naturalness as humans and are widely applied in real life. However, today's most popular and advanced synthesis models, such as Tacotron2 [1], Fastspeech2 [2], and VITS [3],... still require large amounts of data from a single speaker or multi-speaker. It also requires a long time to retrain the entire model every time a new speaker is added. The above TTS models can synthesize high quality with the voices in the training data or seen in training progress. However, without retraining, synthesis quality remains a significant challenge [4,5]. There is a great need for new speech learning applications with only a small

*Corresponding author.
E-mail addresses: phuongpn@tnu.edu.vn (P.N. Phuong); chungtran@ai4med.vn (T.Q. Chung); lcmai@ioit.ac.vn (L.C Mai)

amount of reference speaker data (target speaker), but still ensures that the synthesized voice achieves similarity with the sample voice, so adaptive techniques were proposed to solve these problems. Currently, two main adaptation techniques are popularly used: 1) Fine-tune all or part of the layer with adaptive data based on a pre-trained model (which has been trained with large amounts of data) [6]; 2) Use a vector to capture the representation of the speaker's characteristics with a small amount of adaptive sample [7,8]. These two methods give a good synthesis quality and a high similarity of the synthesized voices to the target voice. However, they require expensive computational resources, and besides, there are still two problems: First, speakers with too small sample data (target voice only one sentence or few seconds) are not adaptable, not good or not trainable; Second, to learn a new voice, it is still necessary to fine-tune a sample of the target voice to update the model parameters and the seen speaker training process for a long time (hours or even days). This leads to consuming computational resources and time-consuming to generate new voices, limiting many possibilities for practical application. A new approach called zero-shot is adopted to adopt a new voice with just one utterance or seconds of the sample without additional training. This technique allows the adaptation of the new voice without retraining; moreover, the data required for training is tiny (just one sentence or a few seconds of target voice data) [5]. Zero-shots in speech synthesis are techniques aimed at training a model that allows the generation of new voices under the condition that these voices have never appeared during training or are unknown during supervised learning (unseen speaker) [10]. These studies open up several useful applications, such as smart speaker systems (with small computational resources) that can tell stories or communicate with their voice, learn new voices on-site without retraining, and flexible speaker voice-over systems are provided on-site. Zero-shot multi-speaker TTS models typically use speaker embedding that can be easily adapted to the new speaker, allowing them to generate a new speaker's voice with much smaller data than other methods adapted by fine-tuning. These models have shown promising results regarding synthesizer quality and generalizability for new speakers. All in all, Zero-shot multi-speaker TTS is an exciting and rapidly growing area and has the potential to significantly impact how TTS systems are built and used in the future. The process of modeling speaker features in TTS consists of 3 steps: 1) Extracting the features of the target speaker; 2) Using these features as conditions for a synthetic TTS model, and 3) Generating the mel-spectrogram based on that representation. In the first step, the zero-shot TTS model typically uses a speaker embedding to represent the target speaker features best. Most of the research focuses on speaker encoder enhancement. However, it is difficult to accurately extract speaker characteristics in zero-shot conditions such as speaker characteristics, speaking style, and emotion. In steps 2 and 3, synthetic models such as non-autoregressive cannot produce diverse synthetic speech. It is because the model is often optimized using a simple regression loss function (e.g. L1, L2), and there is not any probabilistic model to reconstruct the acoustic features [11, 12]. The paper is structured into five main parts: The introduction presents an overview of TTS in the conditions of very little sample data, no training, and low computational cost, thereby posing the need for zero-shot adaptation; Related work presents related research on Zero-shot in TTS, diffusion model, and style vector; The main part presents the Adapt-TTS model with two improvements applying for Multi-speaker TTS zero-shot: 1) Propose Extracting Mel-vector (EMV) architecture allows voice feature representation for better generalization. This architecture has effectively learned speaker characteristics from meager target voice

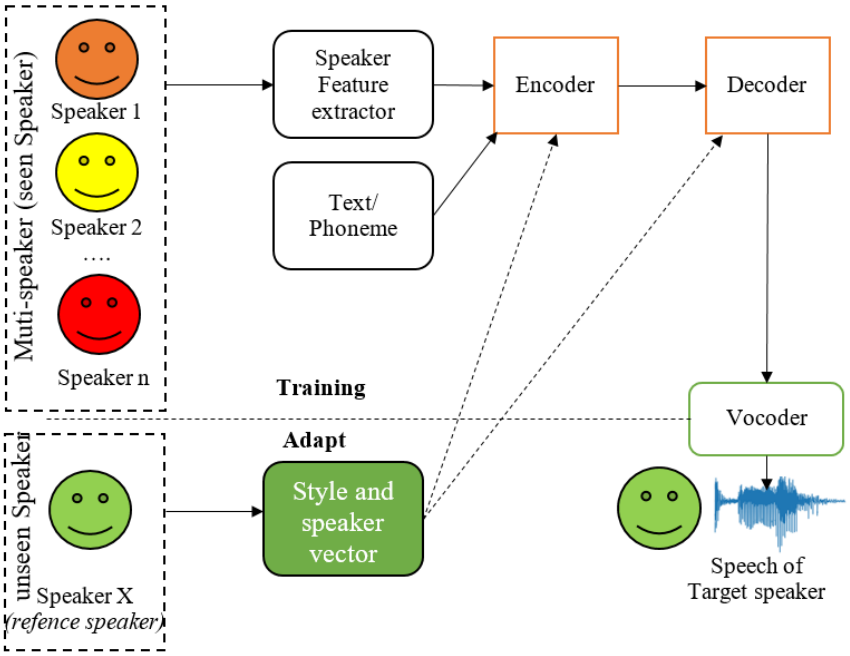


Figure 1: Basic speaking TTS multi-person zero-shot model

samples. 2) Propose a Mel-spectrogram denoiser with kernel architecture using the denoising diffusion model for zero-shot Multi-speaker TTS model to improve synthesis quality and denoising ability; Finally, the experiments are evaluated and concluded.

2. RELATED WORKS

2.1. Zero-shot multispeaker TTS

Zero-shot multi-speaker TTS was first proposed by Arik et al., [8]. The idea of using a speaker encoder as a conditioning signal was further explored [4, 13], trying to close the quality gap between the speakers seen in the training set and those not in the training set (unseen) in the zero-shot Multi Speaker TTS model using embedding as extra information (Fig. 1). This study proposed a speaker embedding that uses neural network-based LDEs speaker embeddings to enhance the similarity and naturalness of voices and uses x-vectors to increase the scalability of the speaker verification task. With the use of embedding parts of the speaker, attention is given to encoding a more general speaking style instead of the speaker's audio [14]; [15] as well as methods that decode differently in the acoustic space such as generative flow [16], further efforts have been made to close the quality gap between seen speakers and unseen speakers. In addition, adapting Multi-speaker TTS models for voice transcription with few target voices requires diversity (including high-quality voice plurals and multiple speech attributes) of the speakers in the training data, and It is very important to achieve high generalization on the unseen-speaker dataset[8]. Therefore, these are still major challenges needed to be solved.

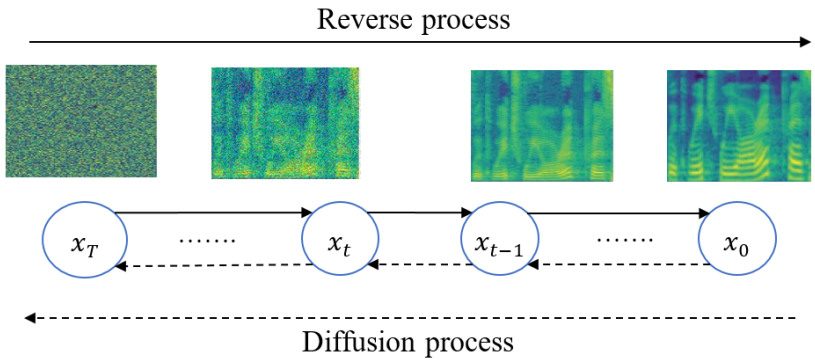


Figure 2: Visual depiction of the Diffusion model's reverse and diffusion process

2.2. Diffusion probabilistic models

The denoising diffusion probabilistic model (referred to as the denoising diffusion model) has shown high efficiency in image and sound generation [17, 18]. The denoising diffusion model is a Markov sequence that has been parameterized and trained using variable inference to generate matching patterns that resemble the original data after a finite time [17]. The transformations of this sequence are learned to reverse diffusion; the Markov series gradually add noise to the data in the direction opposite to the sampling direction until the signal is destroyed. When the diffusion includes a small amount of Gaussian noise, it is sufficient to set the sampling sequence transformations to Gaussian conditional, allowing for a particularly simple neural network parameterization. The diffusion model consists of two opposite processes, as depicted in Fig. 2: 1) The diffusion process is a Markov series with fixed parameters to convert complex data into an isotropic Gaussian distribution by gradually adding Gaussian noises; 2) The reverse process is a Markov sequence implemented by a neural network to learn how to recover the original data from repeated Gaussian white noise. The goal consists of two things, again the distance between the forward-diffusing noise x_t and the reverse diffusing decoder x_T, and argmax how to log-likelihood the maximum reverse diffusion probability between x_0 based on the noise decoder. The diffusion model is highly flexible and allows architecture with the same input and output sizes. That is essential in applying the diffusion model in speech synthesis to achieve the highest quality and likely-hook synthesized voice possible.

2.3. Style vector

An encoder is a component that encodes variable-length strings into fixed-dimensional representation vectors. In the basic multi-speaker TTS model [2, 7, 19], in the speaker encoder, an essential component is the speaker embedding to represent each speaker's voice signal as a feature vector. These vectors do not carry the speaker's features but carry the speaker's identity information. Adaptation-based TTS multi-speaker systems must use speaker features to train and refine the adaptive model. In order to do that, speech processing systems must first convert each variable-length audio clip into a fixed-length vector representing the speaker's identity, called speaker embedding, and real now cluster based on these vectors. Speaker embedding is also widely used in speech-processing tasks, such

as speaker recognition, speaker classification, speech tuning, and language synthesis [19–22]. Traditional methods often use the embedding module to extract a representative vector of the speaker's features. We can model the traditional method as the following formula

$$emb = Emb(Speaker\_ID). \tag{1}$$

However, it can be seen that this simple technique cannot represent the characteristics of each speaker (identity, gender, age, health) because it only uses speaker identifiers as input for the module. Some studies suggest another representative vector that carries information about the speaker's speaking style: style vector. Such as a study [14] that introduced GST (global style token) trained with unknown labels to learn how to model audio expressions and thereby control the synthesis in various styles such as speed, utterance, and textual independence. Sometimes the model shows a successful style transition. However, interleaved training only guarantees that some possible combinations of style classes are seen during training, resulting in a loss of representation of the speaker's style. The study of [11] used SALN (style-adaptive layer normalization) to align the gain and bias of the text input with style extracted from a reference short audio. Thus, it is possible to describe in general the style vector s representing the style of speaker X from the $Speech\_X$ reference audio input encoded by the style encoder as follows

$$s = Style\_encoder(Speech\_X). \tag{2}$$

3. ADAPT-TTS

3.1. Overall architecture

The adapt-TTS architecture consists of the main components: The architecture of Adapt-TTS consists of the following main components: EMV module to extract speaker features and styles of speech into a feature vector. Phoneme encoder module to convert phoneme sequences into phoneme hidden sequences. Then, the variance adapter will add duration, pitch, and energy information to the hidden sequences. Based on the diffusion model kernel, the mel-spectrogram denoiser will receive the hidden information from the previous steps to decode the output into high-quality mel-spectrograms. Finally, the vocoder module converts these mel-spectrograms into speech signals. The overall architecture is depicted in Fig. 3. The detailed architecture and functionality of the proposed enhancement modules are shown below.

3.2. Extracting mel-vector (EMV)

We propose a new module called "mel extraction vector" (EMV module), which can extract a fixed vector from the speaker's mel-spectrogram to accurately represent the speaker's features as the speaker and speaking style. EMV is to take the reference voice X as input. This block aims to extract an embedding stv vector containing the style and features of speaker X

$$stv = EMV(Mel). \tag{3}$$

In this module block, we use three main components, namely encoder feature, decoder feature, and embedding feature.

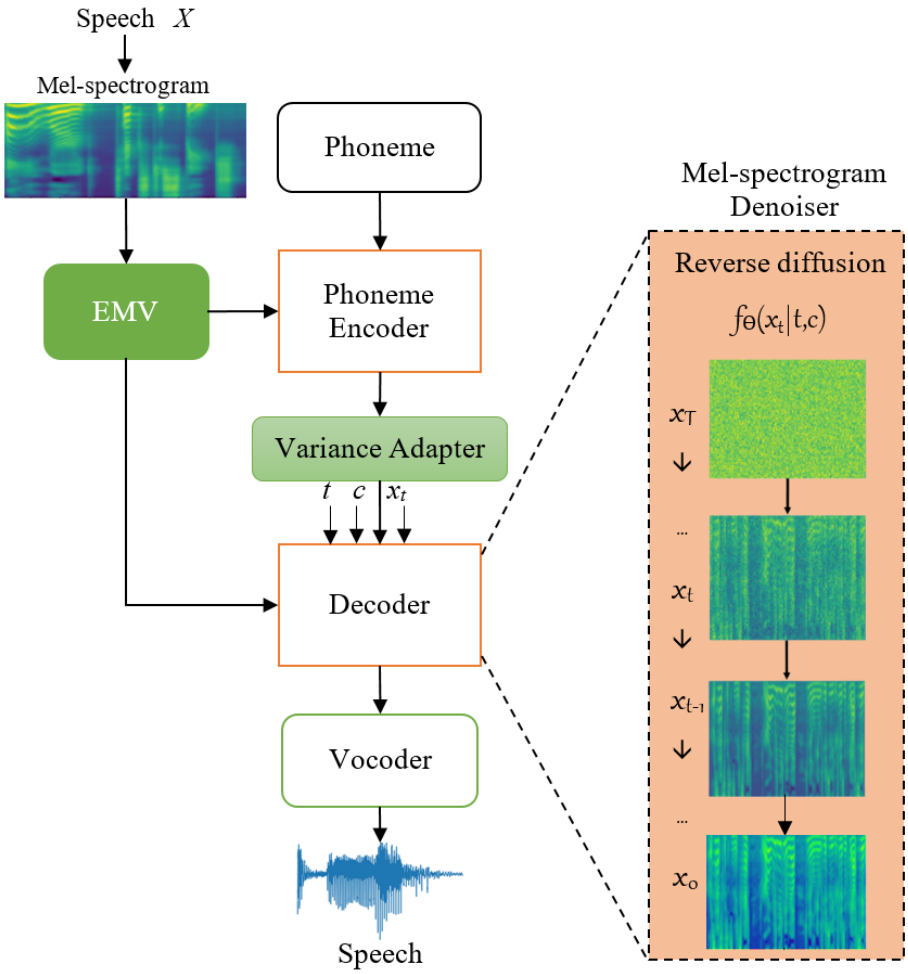


Figure 3: Overall architecture of adapt-TTS

First, at the encoder feature module, the mel-spectrogram input is first fed to the fully connected (FC) layer, and the Mish activation functions convert each frame of the mel-spectrogram into the hidden sequence, which then passes through the two FC layers. The purpose of the encoder feature block is to convert the input feature into an encoder feature. Next, this vector will be passed through the decoder feature module. By using Conv1D +ReLu with the residual result to capture the information sequence from the given speech, this module aims to convert the decoder feature to the decoder feature. In addition, we also integrate skip connection, which will use the valuable features of the previous blocks. Finally, the decoder feature will be moved to the embedding feature module, which has a self-attention module with redundant connectivity plus the affine layer to encode the genetic information. We apply it at the frame level so that EMV can extract better style information even with a short speech sample. Then we temporarily average self-attention output to get a one-way style vector *emb*. Thus this module will generate a vector representing the Mel-spectrogram, and this vector will add to the text-to-speech model. The representation vector will drive the output of the TTS model and produce a synthetic voice similar to the input

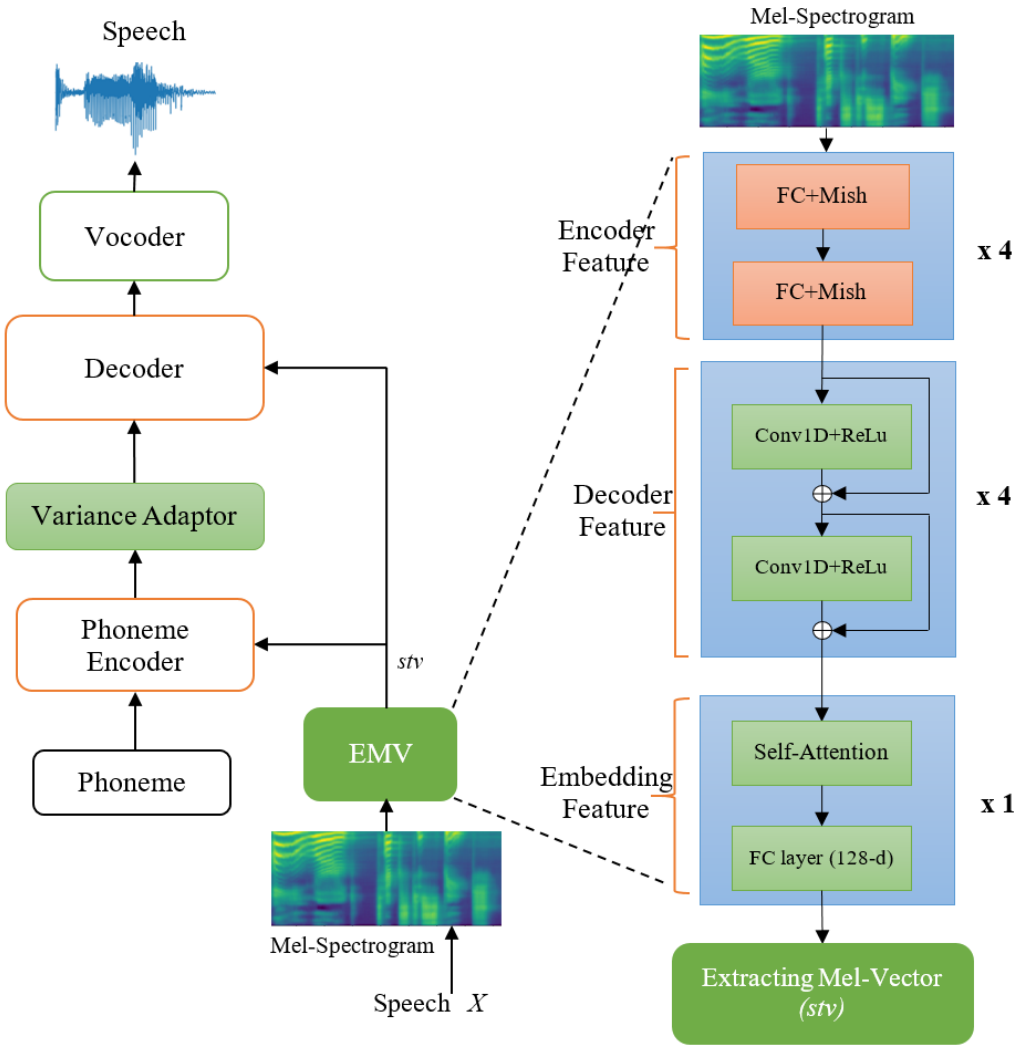


Figure 4: The detailed structure of the EMV module

vector. Architectural details of EMV are shown in Table 1 and Fig. 4 respectively.

3.3. Mel-spectrogram denoiser

The decoder block takes input from the hidden phoneme sequence through the variance adapter to add variance information (e.g., duration, pitch, and energy) and then combines it with the EMV vector (representing human features). Then, Mel-spectrogram-denoiser module will take as input sequence x_t, text c, and time step t to perform high-quality audio denoising and synthesis based on the diffusion model. The inference process of the diffusion model for multi-speaker TTS will optimize the objective function $f_\theta(x_t|t,c)$ to convert the noise distributions into a mel-spectrogram distribution corresponding to the given text and the model. It includes two main processes:

Diffusion process. First, the mel-spectrogram is gradually corrupted with Gaussian noise and transformed into latent variables. This process is called the diffusion process. Assuming

a sequence of variables x_1,..., x_T. with equal dimensions, where $t = 0, 1, ..., T$ is the index for diffusion time steps, the diffusion process transforms the mel-spectrogram x_0 into Gaussian noise x_T through a chain of Markov transitions. Each transition step is defined by a predetermined variance schedule β_1,β_2,..., β_T. Specifically, each transformation is performed using the Markov transition probability $q(x_t|x_{t-1},\ c)$, which is assumed to be independent of the text c and is defined as follows

$$q(x_t|x_{t-1}, c) = N(x_t, \sqrt{1-\beta_t}x_{t-1}, \beta_t I). \tag{4}$$

The entire diffusion process $q(x_{1:T}|x_0, c)$ is a Markov process and can be analyzed as follows

$$q(x_1..., x_T|x_0, c) = \prod_{t=1}^{T} q(x_t|x_{t-1}). \tag{5}$$

Reverse process. The reverse process for generating a mel-spectrogram is the opposite of the diffusion process. Rather than introducing noise, the goal of the reverse process is to recover a mel-spectrogram from Gaussian noise. This process is defined by the conditional distribution $p_\theta(x_{0:T-1}|x_T, c)$ and can be decomposed into multiple transitions based on the Markov chain property

$$p_\theta(x_0..., x_{T-1}|x_T,\ c) = \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t,\ c). \tag{6}$$

Using the reverse transitions $p_\theta(x_{t-1}|x_t,\ c)$, the latent variables gradually reconstruct a mel-spectrogram corresponding to the diffusion time-step with the text condition. Mel-spectrogram denoiser thus learns a model distribution $p_\theta(x_0|c)$ via the reverse process. Let $q(x_0|c)$ be the mel-spectrogram distribution. To achieve a good approximation of $q(x_0|c)$, the reverse process aims to maximize the log-likelihood of the mel-spectrogram, $E_{\log q(x_0|c}[\log p_\theta(x_0|c)]$. As $p_\theta(x_0|c)$ is intractable, we use the parameterization trick demonstrated in [17] to calculate the variational lower bound of the log-likelihood in a closed form. Set $\alpha = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{s=1}^{T} \alpha_s$. The training objective of the mel-spectrogram denoiser is as follows

$$minL_\theta = E_{x_0,\epsilon,t}\Big[||\epsilon - \epsilon_\theta(\sqrt{\overline{\alpha}_s x_0} + \sqrt{1-\overline{\alpha}_s}\epsilon, t, c||_1)\Big], \tag{7}$$

where t is uniformly taken from the entire diffusion time step. Mel-spectrogram denoiser only requires the L1 loss function between the model output $\epsilon_\theta(.)$ and Gaussian noise $\epsilon \sim N(0, I)$, without any auxiliary losses. In the inference phase, mel-spectrogram denoiser recovers a mel-spectrogram from a latent variable by iteratively predicting the diffusing noise added at each forward transition using $\epsilon_\theta(x_t, t, c)$ and then removing the corrupted portion in the following manner

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_s}}\epsilon_\theta(x_t, t, c)) + \delta_t z_t, \tag{8}$$

where $z_t \sim N(0, I)$ and $\delta_t = \eta\sqrt{\frac{1-\overline{\alpha}_{s-1}}{1-\overline{\alpha}_s}\beta_t}$. The scaling factor of the variance is represented by the temperature term η. In mel-spectrogram denoiser, the diffusion time-step t is used as input, allowing for shared parameters across all time-steps. This enables the iterative sampling over all preset time steps, ultimately resulting in the distribution $p(x_0|c)$ for the final mel-spectrogram.

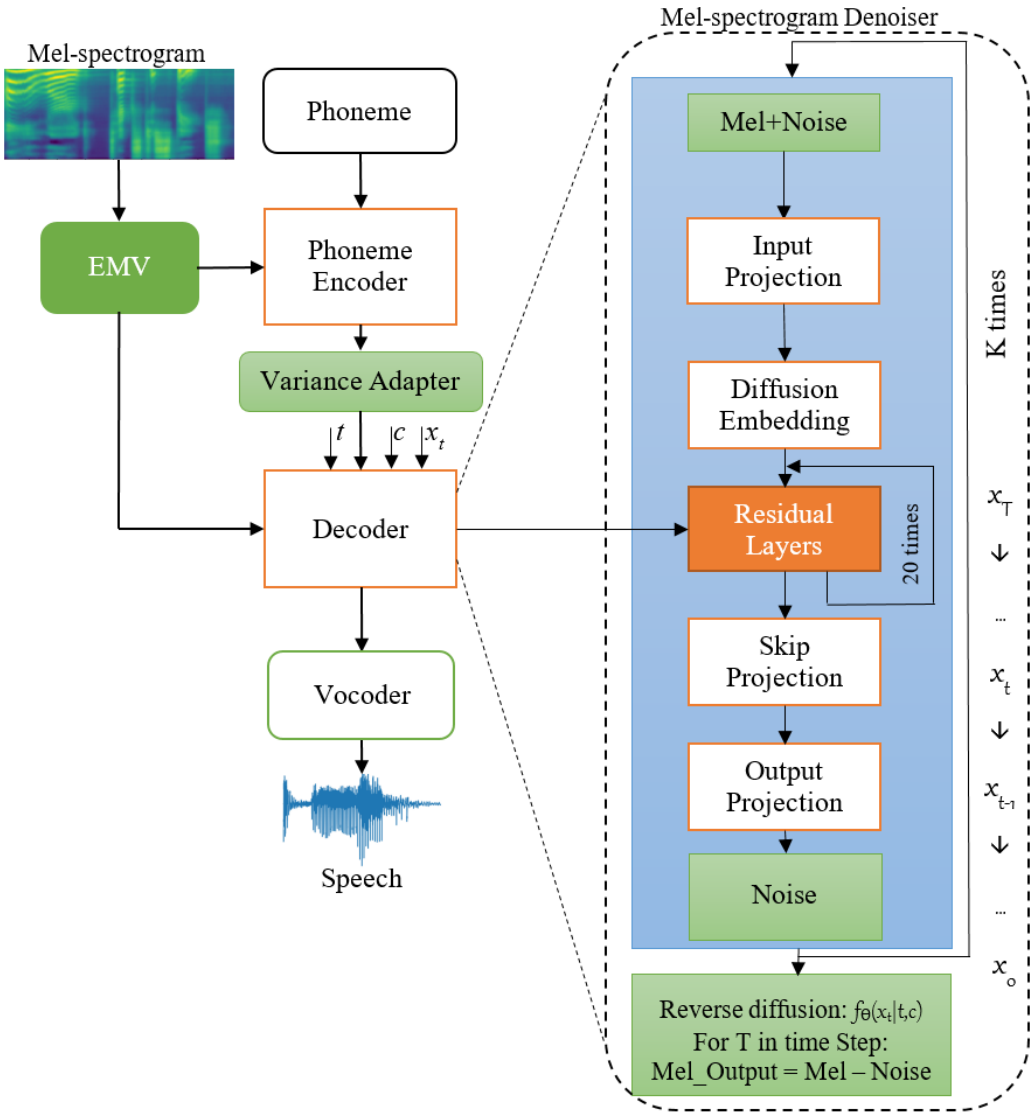


Figure 5: Detailed architecture of the Mel-spectrogram denoiser block

Brief of training and inference

Training. Besides the sample reconstruction loss described above, to assess the quality of the predicted output in terms of pitch, energy, and duration, the loss values of the variation information are computed using the mean squared error (MSE) metric with respect to the ground truth. Additionally, to evaluate the similarity of the predicted mel-spectrogram to the actual audio, the loss is calculated using the mean absolute error (MAE) and structural similarity index measure (SSIM), which provide a measure of audio fidelity. The final loss value during Mel-spectrogram denoiser training includes the following parts

$$L_{final} = L_\theta + L_{SSIM} + L_{duration} + L_{pitch} + L_{energy}. \tag{9}$$

1. L_θ (sample reconstruction loss): MSE mean square error between predicted and target

mel-spectrogram sample;
2. L_{SSIM} (structural similarity index measure loss - SSIM): One minus the SSIM index between the predicted and target mel-spectrogram sample;
3. $L_{duration}, L_{pitch}, L_{energy}$ (variance reconstructs loss): Mean squared error between duration of syllables, pitch, and energy of prediction sample and target.

Inference: During inference, the mel-spectrogram denoiser predicts the input x_0 without noise and then re-adds the noise using the posterior distribution, thereby generating mel-spectrogram planes with increasing details. Specifically, the denoising model $f_\theta(x_t, t, c)$ first predicts x_t, then x_{t-1} is sampled using the posterior distribution $q(x_{t-1}|x_t, x_0)$ given by x_t and predicts x_{t-1}. Finally, a pre-trained vocoder converts the spectrogram plane generated from x_0 to a waveform.

4. EXPERIMENTS AND RESULTS

4.1. Experiments

Dataset. To evaluate the model, a labeled multi-speaker dataset of Vietnamese language was utilized. The dataset comprised 54 speakers, with 26 male and 28 female voices. The dataset also included both Northern and Southern dialects, with each speaker recording approximately 500 utterances. To evaluate the quality of the synthesized sound generated from the proposed models, we prepared 5 sets of data of Vietnamese: of which 4 sets were synthesized from audio references with durations of 1 second, 3 seconds, and 5 seconds, respectively, and 1 set includes the ground-truth audios for matching.

Evaluate results. We will use two models to synthesize: 1) Baseline model proposed by studies [2], and 2) Adapt-TTS model. We use 30 listeners who are Vietnamese officials, teachers, and students studying and working at universities in Vietnam to listen to and grade the sounds provided through a web-based assessment application. We evaluate the integrated system by combining the evaluation both by objective assessment method (Subjective using quantitative indicators such as WER) and subjective assessment (Objective using qualitative indicators) such as MOS/SIM).

Experiment 1: Assess the quality of speech synthesis MOS (mean opinion score) index evaluates audio or video quality based on human judgment. We conducted the MOS assessment by asking a group of listeners and rating their satisfaction with the sound quality synthesized by the models on a scale of 1 to 5. This scale includes 5 ratings as: 5: Excellent; 4: Good; 3: Medium; 2: Poor; 1: Bad. The results of the MOS are calculated by averaging the scores of all the reviews. To ensure objectivity, we also mix ground-truth sounds to determine the maximum scale for the speaker's voice. We also use the WER(word-error-rate) index to measure the percentage of misrecognized words in the synthetic audio word recognition text compared to the ground-truth audio recognition text. This WER index provides additional quality assessment information through the speech-to-text recognition capabilities of existing ASR systems [23].

Experiment 2. Evaluate the similarity between the synthesized voice and the human voice: We use the SIM (similarity) index to measure the similarity between the synthesized and ground-truth audio of the target speaker. We ask listeners to listen and score the similarity synthesized by the models and the ground truth. These assessments are through listening

Table 1: MOS/WER composite quality assessment results of baseline and proposed models with 95% confidence intervals.

Times/Models	**Baseline**		**Adapt-TTS**	
	MOS(↑)	WER(↓)	MOS(↑)	WER(↓)
Groundtruth	4.53	1.35	4.53	1.35
1 second	2.05	8.78	2.89	**3.38**
3 seconds	2.16	7.77	**3.29**	3.14
5 seconds	2.18	6.76	3.31	3.04

to the corresponding pairs of sounds using the 4-scale similarity score based on the query and category suggested in [24]. This scale includes 4(four) ratings: 4. Definitely the same; 3. Maybe the same; 2. Maybe different; 1. Definitely different.

4.2. Results

4.2.1. Quality

Table 2 shows that with only 3 seconds of adaptation audio from the reference speaker, the Adapt-TTS model synthesized audio with a MOS score of 3.29 compared to 4.53 of the human voice without requiring retraining. This score is higher than the Baseline model's score of 2.16. The WER score also shows that with only 1 second of reference speaker audio, the system was able to synthesize audio with a WER of 3.38.

4.2.2. Similarity

Table 3 demonstrates that adapt-TTS achieved a SIM score of 2.22 compared to 3.9 of the speaker's voice with only 3 seconds of adaptation audio from the reference speaker. On the other hand, the baseline model only obtained a SIM score of 1.24. Moreover, by comparing the spectrogram in Fig. 6 with 3 seconds of adaptation samples, it can be seen that the mel-spectrogram image (highlighted in the rectangular box) between the audio generated by adapt-TTS and ground-truth has a significantly higher similarity than the audio produced by the baseline model. Additionally, the audio generated by the baseline model is blurry and contains a lot of noise.

Table 2: SIM similarity assessment results of baseline and proposed models with 95% confidence intervals.

Duration/Model	**Baseline**	**Adapt-TTS**
Groundtruth	3.90	3.90
1 second	1.16	1.71
3 seconds	1.24	**2.22**
5 seconds	1.31	2.6

In order to gain a deeper understanding of the effectiveness of the adapt-TTS model, we illustrate the EMV vectors through the visualization method by computing the distance matrix between the data points of the synthesized audio and the human voice. Figure 7 presents the t-SNE [25] projection of EMV vectors obtained from unseen speakers in the

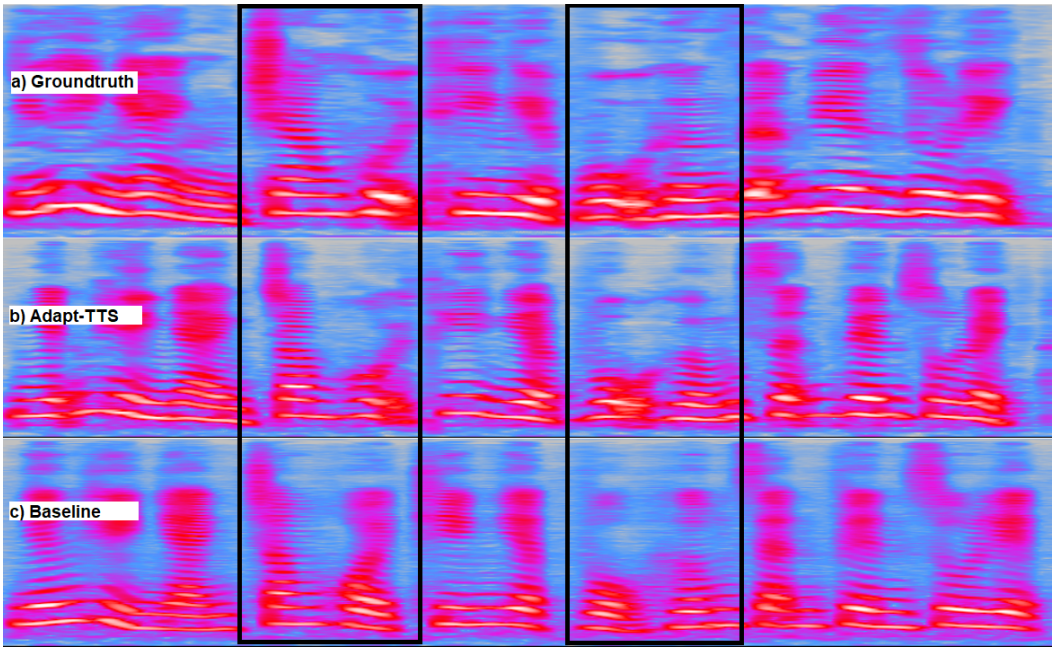


Figure 6: Mel spectrogram of 3 audio: a) Ground truth b) Audio generated by adapt-TTS and c) Audio generated by baseline model.

Vietnamese multi-speaker dataset; Specifically, we chose 10 speakers (5 male and 5 female). Adapt-TTS shows an improved separation of the style vectors compared to the baseline model. The t-SNE chart by the adapt-TTS model (Fig. 7 a) shows that the synthesized and original sounds of the same speaker tend to cluster closely together. Gender characteristics are also clearly clustered in 2 different regions.

5. CONCLUSION

The article proposes an architecture that allows synthesizing a new voice using zero-shot speaker adaptation with only one utterance of the reference speaker without requiring retraining. The proposed approach utilizes EMV for better feature and speaking style representation and mel-spectrogram denoiser for synthesizing higher quality and less noisy speech. The experiments demonstrate that a single 1-3 second sample of the reference speaker's voice is sufficient to synthesize a voice with a MOS of 3.3/4.5 and a similarity score of 2.2/3.9. Although the sound quality produced by the proposed zero-shot multi-speaker TTS model cannot match or replace traditionally trained models. However, it allows for quick learning of new voices without retraining while maintaining acceptable sound quality and achieving high similarity with the target voice. The adapt-TTS model works well at cloning speakers with only a short sentence (several seconds), but if the sample data is increased significantly, the quality and similarity of the voice do not change much. The adapt-TTS model proposed in the article enables adaptive speech synthesis with the potential for diverse applications in daily life.

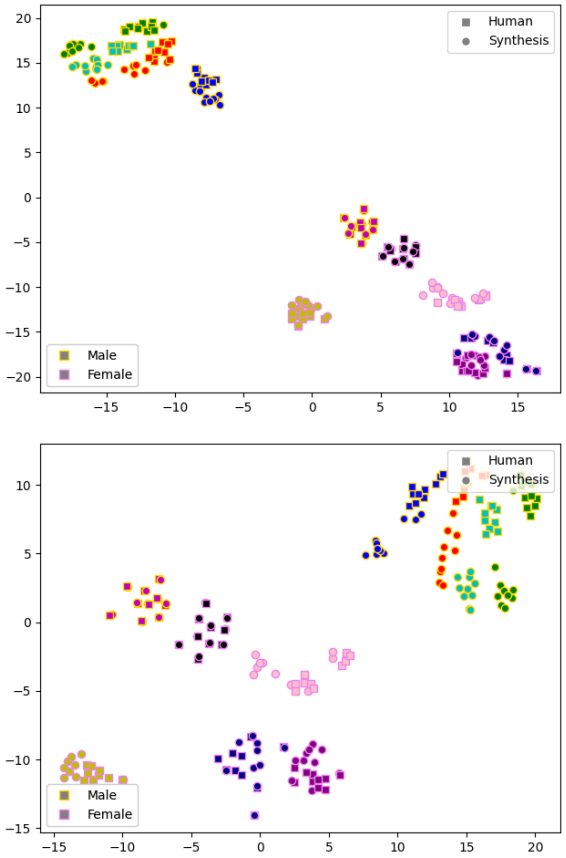


Figure 7: Modeling the spatial distribution of t-SNE between the synthesized voice of the proposed model on the human voice by 10 speakers by a) Adapt-TTS model and b) Baseline model.

ACKNOWLEDGMENT

The authors would like to thank AIMED Co, ltd for funding this research. This work also is supported by the National Science Project under number KC4.0/19-25.

REFERENCES

[1] J. Shen et al., "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, 2018, pp. 4779-4783. Doi: 10.1109/ICASSP.2018.8461368.

[2] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, T-Y. Liu, "Fastspeech 2: Fast and high-quality end-to-end text to speech," *arXiv preprint arXiv:2006.04558.*, 2020.

[3] Shen J, Pang R, Weiss RJ, Schuster M, Jaitly N, Yang Z, Chen Z, Zhang Y, Wang Y, Skerrv-Ryan R, Saurous RA, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," *International Conference on Machine Learning*, pp.5530–5540, 2021. PMLR.

[4] E. Cooper et al., "Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 6184-6188. Doi: 10.1109/ICASSP40776.2020.9054535.

[5] Y. Wu, X. Tan, B. Li, L. He, S. Zhao, R. Song, T. Qin, T-Y. Liu, "Adaspeech 4: Adaptive text to speech in zero-shot scenarios," *arXiv preprint arXiv:2204.00436*, 2022.

[6] E. Cooper, C.I. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen, and J. Yamagishi, "Exploring transfer learning for low resource emotional TTS," *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys), vol 1*, pp.52–60, 2020. Springer International Publishing.

[7] Q. Xie et al., "The multi-speaker multi-style voice cloning challenge 2021," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada, 2021, pp. 8613-8617. Doi: 10.1109/ICASSP39728.2021.9414001.

[8] S. Arik, J. Chen, K. Peng, W. Ping, Y. Zhou, "Neural voice cloning with a few samples," *Advances in Neural Information Processing Systems (NeurIPS 2018),* vol 31, 2018.

[9] F. Pourpanah et al., "A review of generalized zero-shot learning methods," *in IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 45, no. 4, pp. 4051-4070, 1 April 2023. Doi: 10.1109/TPAMI.2022.3191696.

[10] W. Ping, K. Peng, A. Gibiansky, S.O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: 2000-speaker neural text-to-speech," *Proceedings 6th International Conference on Learning Representations (ICLR)*, pp.214–217, 2018.

[11] D. Min, D.B. Lee, E. Yang, S.J. Hwang, "Meta-stylespeech: Multi-speaker adaptive text-to-speech generation," *Proceedings of Machine Learning Research*, pp.7748–7759, 2021.

[12] J. Liu, C. Li, Y. Ren, F. Chen, Z. Zhao, "Diffsinger: Singing voice synthesis via shallow diffusion mechanism," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol 36, no.10, pp.11020–11028, 2022. https://doi.org/10.1609/aaai.v36i10.21350

[13] Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, P. Nguyen, R. Pang, Lopez I. Moreno, Y. Wu, "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *Advances in Neural Information Processing Systems*, vol 31, pp.6184–6188, 2018.

[14] Y. Wang, D. Stanton, Y. Zhang, R.S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, R.A. Saurous, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," *Proceedings of Machine Learning Research,* vol. 80, pp.5180–5189, 2018.

[15] S. Choi, S. Han, D. Kim, S. Ha, "Attentron: Few-shot text-to-speech utilizing attention-based variable-length embedding," *arXiv preprint arXiv:2005.08484*, 2020.

[16] E. Casanova, C. Shulby, E. Gölge, N.M. Müller, De F.S. Oliveira, A.C. Junior, A.D. Soares, S.M. Aluisio, M.A. Ponti, "SC-glowtts: An efficient zero-shot multi-speaker text-to-speech model," *arXiv preprint arXiv:2104.05557*, 2021.

[17] J. Ho, A. Jain, P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol 33 pp.6840–6851, 2020.

[18] A.Q. Nichol, P. Dhariwal, "Improved denoising diffusion probabilistic models," *Proceedings of Machine Learning Research*, vol. 139, pp.8162–8171, 2021.

[19] S.-F. Huang, C.-J. Lin, D.-R. Liu, Y.-C. Chen, and H.-Y. Lee, "Meta-TTS: Meta-learning for few-shot speaker adaptive text-to-speech," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 30, pp. 1558-1571, 2022. Doi: 10.1109/TASLP.2022.3167258.

[20] Y. Liu, L. He, J. Liu, M.T. Johnson, "Introducing phonetic information to speaker embedding for speaker verification," *EURASIP Journal on Audio, Speech, and Music Processing*, vol 2019, pp.1–17, 2019. https://doi.org/10.1186/s13636-019-0166-8

[21] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," *2016 IEEE Spoken Language Technology Workshop (SLT)*, San Diego, CA, USA, 2016, pp. 165-170. Doi: 10.1109/SLT.2016.7846260.

[22] Y. Kwon, J.W. Jung, H.S. Heo, Y.J. Kim, B.J. Lee, J.S. Chung, "Adapting speaker embeddings for speaker diarisation," *arXiv preprint arXiv:2104.02879*, 2021.

[23] S. Schneider, A. Baevski, R. Collobert, M. Auli, "Wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[24] M. Wester, Z. Wu, J. Yamagishi, "Analysis of the Voice Conversion Challenge 2016 Evaluation Results," *Interspeech*, pp.1637–1641, 2016.

[25] G. Hinton, van der L. Maaten, "Visualizing data using t-SNE," *Journal of Machine Learning Research,* vol. 9, pp. 2579–2605, 2008.

Received on March 01, 2023
Accepted on May 07, 2023

International Journal of Asian Language Processing
Vol. 32, Nos. 2 & 3 (2022) 2350004 (15 pages)

DOI: 10.1142/S2717554523500042

World Scientific
www.worldscientific.com

Improving Few-Shot Multi-Speaker Text-to-Speech Adaptive-Based with Extracting Mel-Vector (EMV) for Vietnamese

Phuong Pham Ngoc*
Thai Nguyen University, Thai Nguyen, Vietnam
phuongpn@tnu.edu.vn

Chung Tran Quang
AIMed Vietnam Artificial Intelligence Solutions, Ha Noi, Vietnam
Japan Advanced Institute of Science and Technology
(JAIST) Nomi, Ishikawa 923-1292, Japan
chungtran@ai4med.vn

Mai Luong Chi
Institute of Information Technology
Vietnam Academy of Science and Technology
Ha Noi, Vietnam
lcmai@ioit.ac.vn

Received 3 January 2023
Revised 6 May 2023
Accepted 8 May 2023
Published 29 June 2023

Training a multi-speaker Text-to-Speech (TTS) model requires multiple speakers' voices to generate an average speech model. However, the average speech synthesis model will be distorted or averaged, resulting in low quality if the new speaker's voice has too little data to train. The existing methods require fine-tuning the model; otherwise, the model will achieve low adaptive quality. However, for synthesis voice to achieve high adaptive quality, at least thousands of fine-tuning steps are required. To solve these issues, in this paper, we propose a Vietnamese multi-speaker TTS adaptive-based technique that synthesizes high-quality speech and effectively adapts to new speakers, with two main improvements: (1) propose an Extracting Mel-Vector (EMV) architecture with three components, the Encoder–Decoder–Embedding Features, which enables complete learning of speaker features with Mel-spectrograms as input for few-shot training and (2) a continuous-learning technique called "data-distributing" preserves the new speaker's characteristics after many training epochs. Our proposed model outperformed

*Corresponding author.

the baseline multi-speaker synthesis model and achieved a MOS score of 3.8/4.6 and SIM of 2.6/4 with only 1 min of the target speaker's voice.

Keywords: Speaker adaptation; multi-speaker text-to-speech; speaker embedding; EMV.

1. Introduction

End-to-end Text-to-Speech (TTS) models are becoming more and more popular and dominant in terms of quality and simplified engineering implementation. The end-to-end model is the recommended model for simplifying text analysis modules and directly taking character strings or phonemes as input, simplifying audio features using Mel-spectrogram, and allowing the direct generation of audio from the input text. Some popular end-to-end state-of-the-art models such as Tacotron2,[1] FastSpeech2,[2] and VITS[3] can synthesize voices close to human voices.

Training a multi-speaker TTS model from scratch is computationally expensive, and adding new speakers to the dataset requires the model to be retrained. Some neural network-based TTS models include speaker encoder modules to extract hidden representations of speaker's characteristics such as speaker characteristics (physiological features such as identity, gender, age, and health) and speaking style (psychological features such as individual and collective).[4–6] A speaker encoder that can extract the speaker embedding vector space from one or several desired speakers is an essential component in a multi-speaker TTS system. This embedding vector is used to customize the TTS output and generate new speeches from the target speaker. The speaker encoder can be trained with the rest of the multi-speaker TTS modules. The speaker encoder can also be pre-trained to generate embeddings to train multi-speaker TTS on a multi-speaker dataset.[7] Multi-speaker TTS systems require considerable data to model the characteristics of the speakers during training. However, many personalized applications have limited data from the target speaker, such as restoring the voices of great people and the voices of the deceased.

In addition, with rich-resource languages such as English, Japanese, and Chinese, through the long development process, there are large and diverse databases. In Vietnam, research on speech processing has existed since the early 20th century. Up to now, a number of high-quality speech datasets have been widely published, such as VOV (radio source), MICA VNSpeechCorpus, VAIS-1000, VLSP. However, Vietnamese databases for speech synthesis research are still limited and lack high-quality recorded datasets that exceed 10 h for TTS.[8] Moreover, Vietnamese is a complex language compared to other languages due to its monosyllabic nature and tonal system, in which each syllable is associated with a specific tone.[9] Gathering such a significant amount of data is a costly, time-consuming process and is not feasible for creating new speakers. So, the study of an adaptive method to solve the problem of lack of data for Vietnamese speech synthesis is an indispensable requirement. The adaptive technology allows synthesizing a new voice with only a few reference samples, known as "few-shot TTS" adaptations. There are two main

approach methods to deploy an adaptive few-shot TTS system: fine-tuning-based and embedding-based adaptation.

With the fine-tuning-based adaptive approach, several studies have proposed fine-tuning either all the parameters of the model or a part of them based on a small target voice. Some proposals require several minutes of adaptive data and are less attractive than the actual requirements. A traditional approach is fine-tuning a part or whole model with a pre-trained model using a small dataset of target speakers.[10–12]

With the embedding-based adaptive approach, several studies have proposed specialized embedding methods to represent various speech features (e.g., speaking styles and speaker identities) or switch to a variable-length embedding method to preserve transient information.[13–16] However, TTS adaptation techniques (in terms of similarity and quality) show that speakers seen during training consistently give better quality than speakers not seen during training.[17]

Furthermore, when training a multi-speaker TTS system, there will be a situation where when fine-tuning in serial to learn new speakers with small data, the model will forget the speakers that have already been learned. Using speaker embedding could not improve the quality compared to fine-tuning adaptation. So, we need a data distribution technique to ensure that adaptive voices with little data and the sample data are still fixed during training so that the model does not forget speaker characteristics after many rounds of continuous training.[18,19] Therefore, this paper proposes two solutions to improve the quality of few-shot multi-speaker TTS adaptive based for Vietnamese: First, we propose an Extracting Mel-Vector (EMV) architecture that allows learning speaker features better than others. Speaker embedding baseline architecture; Second, we propose a new technique called "data-distributing" for continual-learning in TTS models and propose to create various advantages such as expanding multi-speaker TTS systems and reducing training costs, and improving the quality of an existing speaker's voice when only a small sample of the target speaker's voice is needed.

2. Related Works

2.1. *Text-to-speech*

Currently, the statistical parameter-based speech synthesis model has completely replaced the unit selection-based speech synthesis by its ability to adapt and control the speaker's characteristics and speech style. The two most popular and advanced TTS speech synthesis Deep Neural Network (DNN)-based architectures: (1) Tacotron2[1] represents autoregressive architecture and (2) FastSpeech2/2s[2] represents a nonautoregressive architecture. Autoregressive neural network-based TTS models such as Tacotron2 generate Mel-spectrograms from text and then synthesize speech from generated Mel-spectrograms using a trained vocoder set private. They often suffer from slow inference speed and persistence problems (skipping and repeating words). In recent years, nonautoregressive TTS models have been

designed to create Mel-spectrograms at high speed and avoid issues while achieving high quality close to previous autoregressive models. Among those nonautoregressive TTS methods, FastSpeech2 is one of the most successful. FastSpeech2 has two ways to reduce the one-to-many mapping problem: first, to remove the distillation pipeline between teacher–student and to directly use the Mel-spectrogram of ground truths as the training target, and second, to use the set of Mel-spectrograms to train the model. Variance adaptor includes not only duration predictors but also pitch and energy predictors. FastSpeech2 further simplifies the training process and leads to a complete end-to-end system that directly generates waveforms from the text without generating the Mel-spectrogram according to the acoustic, end-to-end model, in the end, generates the audio waveform at the vocoder. However, for high-quality synthetic sound, FastSpeech2 needs dozens of hours of labeled audio to train a single speaker model, or hundreds of hours of labeled audio to train multi-speakers model (approximately 30 min/speaker).

2.2. *Speaker adaptation*

When the TTS system can synthesize high-quality speech, the next most important task is efficient speech synthesis to reduce the cost of speech synthesis, including collecting and labeling training data. Where TTS adaptation using fewer data to help low-resource languages is an exciting direction becoming the leading research target. So, adaptive speech synthesis synthesizes arbitrary speech from any person with a small amount of accurate sample data. The synthesized voice will have the characteristic of the target voice with its voice characteristics and prosodic features. Two main approach methods to deploy an adaptive TTS system are fine-tuning-based and embedding-based. When fine-tuning, the model only changes the decoder module's parameters. The model can learn the whole speaker's features from a small amount of data after fine-tuning. The primary acoustic model of end-to-end TTS is a formula for transforming text information into acoustic features. It demonstrated that a common approach is fine-tuning the pre-trained multi-speaker acoustic model with the target speaker's corpus. However, with a fine-tuned traditional approach, creating a new voice in a new language different from a pre-trained model still requires a large amount of data (more than 5 h, which is difficult with low-resource languages). Using too small amounts of data makes it easy to cause overfitting by adapting directly to the end-to-end acoustic model. Previous work has proposed a "multi-pass fine-tune" model to borrow an English pre-trained model and first fine-tune with an intermediate Vietnamese pre-trained model, then second fine-tune with an adaptive voice.[20] The main acoustic features learned/transferred from the English (large dataset) and Vietnamese (medium dataset) to generate a new voice that only needs a small amount of adaptive data to model learning voice features of the target speaker according to word pronunciation, the phoneme mapping model between source and target linguistic symbols. The speaker embedding vector of the

old speaker has been frozen and only lets the new speaker adapt its embedding to the TTS model.

With the embedding-based adaptive approach, an embedded vector or an embedded network encodes speaker features and speaker styles. During the multi-speaker average model training, the embedding vector was used to distinguish acoustic features to identify speakers and speaking styles. When inferring, the vector representation of the target speaker is used as an input to generate the adaptive voice. Recent research on speaker embedding has put forth Gaussian Mixture Model (GMM) and DNN-based models that aim to extract fixed-dimensional vectors. Snyder *et al.*[21] introduced the x-vector, a feature obtained by directly training a DNN model for speaker discrimination. In speaker verification systems, x-vectors are generally computed at the sentence level. Xie *et al.*[22] proposed a Thin-ResNet model that utilizes CNN and NetVLAD (or GhostVLAD) to generate fixed-length vectors that surpass the performance of both i-vector and x-vector. The approach involves learning to create frame-level speaker embedding and aggregate vectors over time.

Several studies propose an alternative approach involving a style vector representing the speaker's speaking style. For instance, in a study,[5] a global style token (GST) was introduced, which was trained without labels to learn how to model audio expressions and control synthesis in various styles, including speed, utterance, and textual independence. This method occasionally demonstrates successful style transitions. However, since interleaved training only ensures exposure to some possible combinations of style classes during training, it may result in a loss of representation of the speaker's style. Arik *et al.*[23] proposed speaker encoding is based on training a separate model to directly infer a new speaker embedding. Another study[24] utilized Style-Adaptive Layer Normalization (SALN) to adjust the gain and bias of the text input with style obtained from a reference short audio, enabling a general description of the style vector to represent the speaking style of the speaker from the reference audio input encoded by the style encoder.

2.3. *Catastrophic forgetting*

In addition, catastrophic forgetting (CF) has been a well-known problem in neural networks for many years.[25] In particular, the method of speech synthesis by direct fine-tuning can cause a CF because the training data of the new speaker are too small. Various continuous-learning methods (also known as continuous learning) address this problem from different perspectives, such as (1) experience replay or adjusting the weights of the network and class-incremental learning[26]; (2) introduce regularization-based new loss functions to distill previous knowledge or penalize important parameter updates[27]; (3) effectively reuse different speakers' knowledge while keeping privacy.[28,29]

3. Propose Multi-Speaker Text-to-Speech Adaptive-Base

This section will present a proposed model of multi-speaker TTS adaptive-base through two main improvements: EMV vectors that allow learning to represent speaker features and "data-distributing" training technique to ensure the adaptive data always appear during training.

3.1. *Extracting mel-vector*

The encoder is a component that allows the encoding of variable-length strings into fixed-dimensional representation vectors. In a baseline multi-speaker TTS model,[2,30,31] in the speaker encoder, a critical component is the speaker embedding, representing the speaker's voice signal as a feature vector. Adaptation-based multi-speaker synthesis systems must use speaker features to train and tune the adaptive model. To do that, speech processing systems must first transform each variable-length audio clip into a fixed-length vector representing the speaker's identity, called speaker embedding, and perform clustering based on these vectors. Speaker embedding is also widely used in speech processing tasks such as speaker identity, speaker diarization, speech adaptation, and language synthesis.[30,32–34] Traditional methods often use an embedding module to extract a representative vector. We can model the traditional method as the following formula. Multi-speaker speech synthesis system using baseline speaker embedding, as depicted in Fig. 1.

$$\text{emb} = \text{Emb}(\text{Speaker\_ID}).$$

Nonetheless, this basic method cannot capture the unique characteristics of each speaker, such as their identity, gender, age, and health, as it solely relies on speaker identifiers as input. Thus, we propose a new module Mel-Vector Extraction

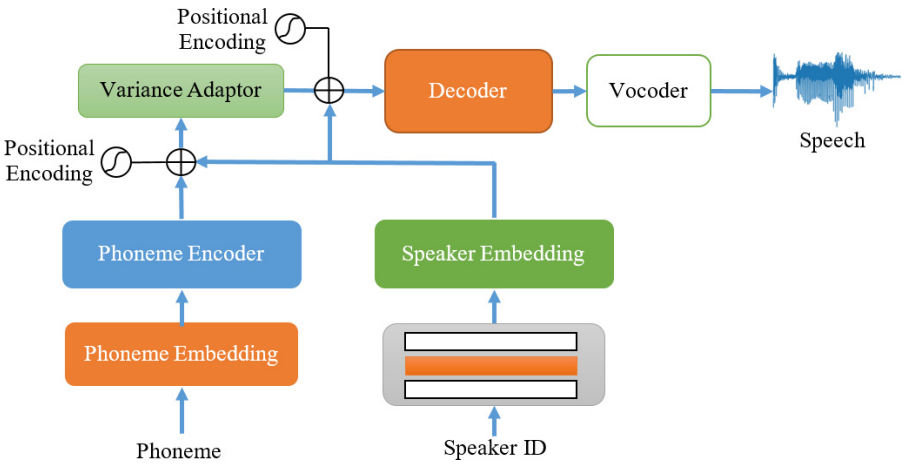


Fig. 1. Architectural diagram of a baseline multi-speaker speech synthesis system using baseline speaker embedding.

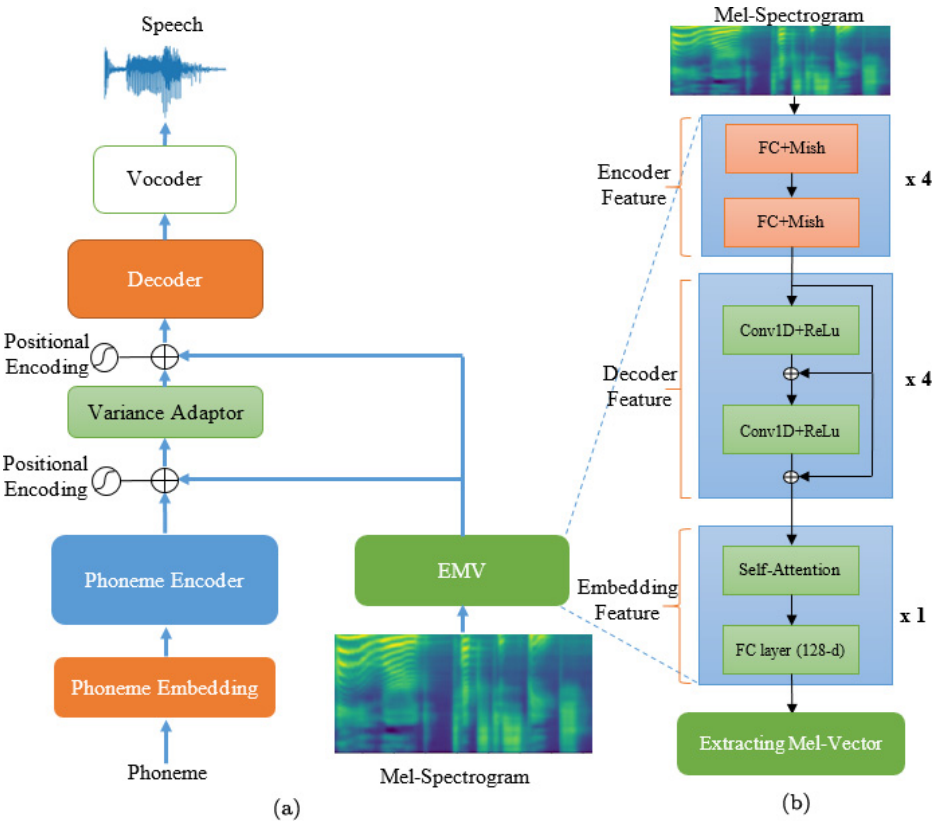


Fig. 2. (a) Architectural diagram of a Vietnamese multi-speaker TTS adaptive-based model with the EMV module and (b) detailed structure of the EMV module.

(EMV module), based on the Mel-style encoder original architecture that modified to efficiently extract a fixed vector from a Mel-spectrogram,[24] as depicted in Fig. 2.

$$\text{emb} = \text{EMV}(\text{Mel}).$$

More specifically, in this module, we use three components (Encoder Feature, Decoder Feature, and Embedding Feature). The first module, Encoder Feature, takes the Mel-spectrogram (*Mel*) input and passes it through a fully connected (FC) layer with Mish activation functions to convert each frame into a hidden sequence. This sequence then goes through two more FC layers to create the Encoder Feature, which is the module's goal. The resulting vector is then passed to the Decoder Feature module, which utilizes Conv1D + ReLu and residual connections to capture information from the speech and convert it to the Decoder Feature. This module also incorporates skip connections to retain valuable features from previous blocks. Finally, the Decoder Feature output goes through the Embedding Feature module, which contains a self-attention module with redundant connectivity and an affine

Table 1. EMV architecture.

Layer	Input × Output
Mel	$T \times 80$
FC + Mish	$T \times 128$
FC + Mish	$T \times 128$
ConV1D + ReLu	$128 \times T$ (transpose)
ConV1D + ReLu	$128 \times T$
Self-attention	$T \times 128$ (transpose)
FC layer	1×128

layer to encode genetic information. This module operates at the frame level, allowing for the extraction of better style information from short speech samples. The self-attention output is temporarily averaged to generate a style vector, which is added to the TTS model. This vector drives the TTS model's output and produces a synthetic voice similar to the input vector. The EMV's architectural details are illustrated in Table 1 and Fig. 2.

3.2. *Data distributing*

In the speech synthesis training phases, if we choose batch size = 32, that is, for one iteration, we will randomly give 32 audio clips of multiple speakers running forward in the neural network. Then, feed another 32 random audio samples, without including the previous audio samples, into the network and continue until there are no more audio samples in the training dataset. Finally, finish a training epoch. However, choosing a random audio sample to train multiple speakers will create a loss of control when the speakers' data are of different sizes. There will be a speaker that dominates in terms of speaking duration and a large number of unique syllables, but also a speaker with a low speaking duration and a small number of unique syllables. This will have the advantage that voices with a few samples will learn pronunciations from speakers with extensive data. However, the problem of CF in the neural network will make the synthesis model poor for speakers with small data (no longer like the sample voice and poor quality). With a new speaker with only a few minutes of sample data, the aggregate quality is poor (in terms of similarity and quality) and the model will be biased learning data with many samples. To overcome that problem, we use a training technique called "data-distributing" to ensure that the batch size always keeps a fixed amount of audio of the adaptive voice for each training session. That will ensure the adaptive voice will not forget the characteristic parameters after each training epoch. With a large number of speakers in a multi-speaker dataset or large batch size, the data-distributing technique will not change, processing will take the audios of each speaker in turn into batches and attach it with a fixed batch containing adaptive audio to train. The process repeats until all the audios of all speakers exist in the dataset for training. An example is shown in Fig. 3, when training with batch size = 32 of 4 speakers, which includes 1 (one) voice to adapt. We will always keep 8 (eight) audios from

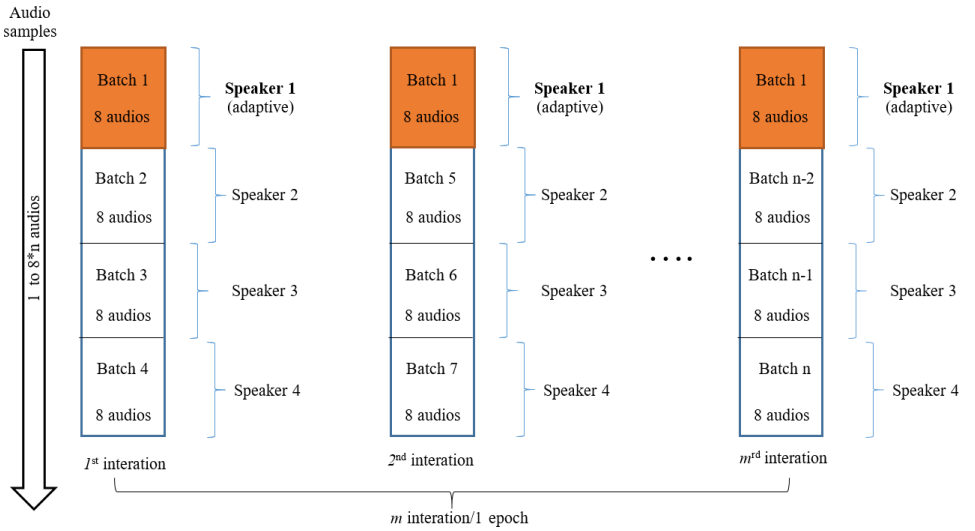


Fig. 3. Batch size-distributing training with batch size = 32 of 4 speakers, which includes 1 adaptive voice (audio samples kept in the first batch) and 3 speakers.

all the audio of the adaptive voice in the first batch and choose 24 random audios from the other speakers in the remaining batches.

4. Experiments and Results

4.1. *Experiments*

To evaluate the multi-speaker TTS systems, we use a state-of-the-art multi-speaker speech synthesis network as FastSpeech2[2] and HifiGAN vocoder as baseline models, as depicted in Fig. 1. The FastSpeech2 architecture consists of main parts: (1) The Phoneme Encoder converts the phoneme embedding sequence into the hidden phoneme sequence; (2) The variance adaptor aims to add variance information such as duration, pitch, and energy to the hidden phoneme sequence; (3) The mel-spectrogram decoder converts the adapted hidden sequence into mel-spectrogram sequence in parallel; (4) Vocoder transforms from mel-spectrogram to waveform. A Montreal Forced Alignment (MFA) tool is used to extract the Vietnamese phoneme duration.[35] A baseline speaker embedding takes speaker IDs as input to encode the corresponding speaker into speaker feature vectors. The duration, pitch, and energy predictor are optimized with a mean square error (MSE) loss function to minimize the model's output with ground truth. In the multi-speaker TTS adaptive-based model as depicted in Fig. 4, we will replace baseline speaker embedding with an EMV module to encode speaker features directly from the mel-spectrogram; the data-distributing training technique is also used to keep the adaptive speech characteristic parameters. We conduct two experiments to evaluate the performance of the models for Vietnamese:

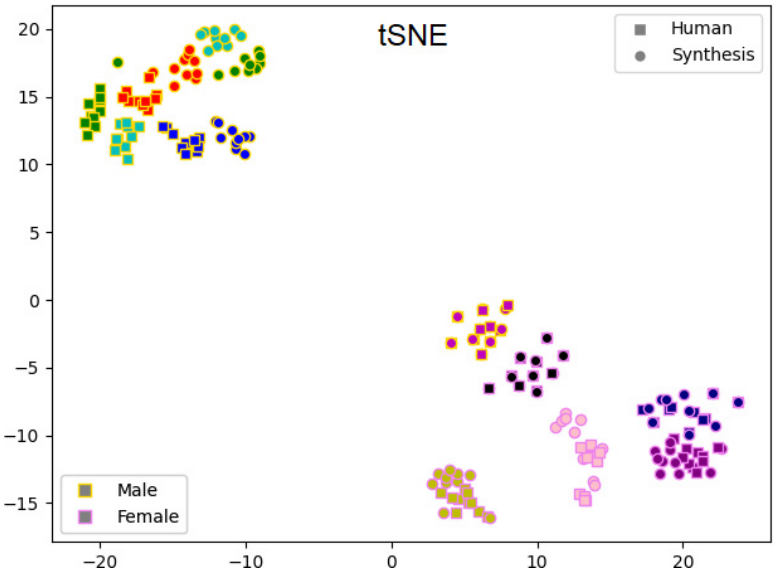


Fig. 4. t-SNE speaker style vector visualization of human voices and synthesis voices (using EMV).

Datasets: A labeled multi-speaker Vietnamese dataset was used to evaluate the model: 54 speakers in total, including 26 male voices, 28 female voices, and Northern–Southern dialect, with each speaker reading about 500 utterances. The adaptation data were divided into four groups (1, 2, 4, and 16 min, respectively) to train the few-shot models.

Experiment 1: Evaluate the quality of the synthesized sound generated by the baseline multi-speaker TTS model and the multi-speaker TTS adaptive-based model. To estimate the minimum amount of audio for training the adaptive-based model, 16 min is not required to evaluate. Groundtruth audio was also added to evaluate to ensure objectivity. The evaluation results are shown in Table 2.

Table 2. Quality assessment table between baseline multi-speaker TTS model (using baseline speaker embedding) and multi-speaker TTS adaptive-based model (using EMV module and data-distributing technique).

Times/models	Baseline multi-speaker TTS model		Multi-speaker TTS adaptive-based model	
	MOS(↑)	WER(↓)	MOS(↑)	WER(↓)
Groundtruth	4.60	—	4.60	—
1 min	3.39	8.40	**3.81**	**5.00**
2 min	3.52	7.28	3.87	2.75
4 min	3.59	6.16	4.00	2.00
16 min	**3.61**	5.60	—	1.25

Mean opinion score (MOS) scale is used to evaluate the quality of voice generated by the system. The sound synthesized from the baseline multi-speaker TTS model and multi-speaker TTS adaptive-based model will be evaluated by 26 listeners with 95% confidence intervals. Each listener will listen to a set of 120 audio mixed between the ground truth and the audio generated by the baseline and adaptive model. The listener will have five rating options for each audio: (1) Bad; (2) Poor; (3) Fair; (4) Good; (5) Excellent. In addition, we also use Word Error Rate (WER), which validates the intelligibility of the generated speech. We use a pretrained Wave2vec Automatic Speech Recognition (ASR) model to calculate the WER.[36] Neither Mel-Cepstral Distortion (MCD) nor WER is the absolute metric for assessing voice quality, so we only use them for relative comparisons.

Experiment 2: Assess the similarity of the synthesized audio generated by the baseline multi-speaker TTS model and the multi-speaker TTS adaptive-based model against the ground truth with only 1 min of adaptive data. Mel-spectrogram analysis of the synthesized audio is compared to groundtruth audio to see the difference between the samples. The evaluation results are shown in Table 3.

MCD is used to measure how different two sequences of Mel-cepstral are for evaluating speaker adaptation performance.[37] The smaller the MCD, the closer the synthetic voice is to natural speech reproduction. The SIM (Similarity) index is also used to compare the similarity of synthesized speech and ground truth. 26 listeners have four options to evaluate each audio pair with 95% confidence intervals: (1) definitely different; (2) maybe different; (3) maybe the same; (4) definitely the same. 90 audio pairs are used for evaluation (randomly mixed between audios generated by adaptive models, baseline models, and groundtruth audios).

4.2. *Results*

We assess the performance of EMV on TTS synthesis tasks with limited data in this section. Access to the audio samples is provided at http://demo.aimed.edu.vn.

4.2.1. *Quality*

Table 2 shows that, with only 1 min data of target speaker, the multi-speaker TTS adaptive-based model synthesized sounds with a MOS score of 3.81 compared

Table 3. Speaker similarity of baseline multi-speaker TTS model and multi-speaker TTS adaptive-based model compared to ground truth with only 1 min of adaptive data.

Experiment — Model (1 min adaptive data)	MCD	SIM
Groundtruth	—	4.0
Baseline multi-speaker TTS model	7.36	1.96
Multi-speaker TTS adaptive-based model	**6.54**	**2.60**

to a score of 4.6 for the human voice. That score is much higher than the 3.61 MOS generated from the baseline multi-speaker TTS model (using 16 min of target voice). The WER scores also show that the multi-speaker TTS adaptive-based model synthesizes voices better than the baseline multi-speaker TTS model.

4.2.2. *Similarity*

Table 3 shows that, with just 1 min of target voice data, the adaptive-based multi-speaker TTS model has a SIM similarity score of 2.60 compared to 4.0 for the human voice. This score is much higher than the 1.96 SIM points of the baseline multi-speaker TTS model (using 1 min of target voice). The MCD score of the adaptive-based multi-speaker TTS model also decreased by more than 10% compared to the baseline model.

Visualize style vectors to understand better the EMV vector's effectiveness in encoding individual speakers' styles. In Fig. 5, we illustrate the t-SNE[38] projection of the style vectors from the speakers of the human voice and the synthetic voice, respectively. Using 10-speaker voices (five male and five female), we can see that the EMV system models speaker features very well when representing the similarity between human and synthetic voices through clearly and closely clustered performance points in each area separately. The speakers are almost clearly gender-segregated in the t-SNE visualization, with all female speakers appearing on the right and all male speakers appearing on the left. The synthesizers and ground truth of each speaker are relatively close to each other, indicating that the speaker encoder has learned to represent the speaker's space properly.

In addition, Fig. 5, depicts the mel-spectrogram of the voice synthesized from the baseline multi-speaker TTS model, the multi-speaker TTS adaptive-based model and voice groundtruth models. It can be seen that, with only 1 min of adaptive

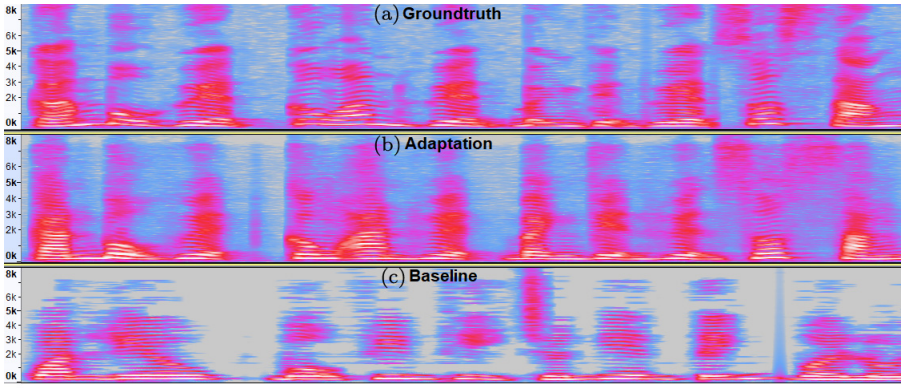


Fig. 5. Compare the Mel-spectrogram of the (a) groundtruth audio, (b) the audio generated from the adaptive model, and (c) the audio generated from the baseline model with 1 min-sample of target speech.

data, the wave spectrum is quite similar to the groundtruth spectrum and utterly different from the sound wave spectrum generated from the multi-speaker TTS baseline model. Compare the Mel-spectrogram of the (a) groundtruth audio, (b) the audio generated from the adaptive model, and (c) the audio generated from the baseline model with 1 min-sample of target speech.

5. Conclusions

This paper proposes an adaptive model to improve the quality of the Vietnamese multi-speaker TTS system with two improvements: the EMV module and the "data-distributing" training technique. The proposed model has shown superior performance over the baseline multi-speaker TTS model, which uses traditional speaker embedding. Experimentally, with only 1 min, the proposed model achieved high similarity and good speech quality compared to the groundtruth voice. With only 1 min of adaptive data, the adaptive-based multi-speaker TTS model achieved 3.8 MOS points, and this score is equivalent to the MOS score using 16 min of adaptive data based on the multi-pass fine-tune technique that we presented in report.[20] It demonstrates that the EMV module has brought full speaker features, is suitable for few-shot training models, and has the potential to show the hidden features of unseen-speaker in few-shot TTS systems. In the future, we will evaluate the EMV module to enhance the zero-shot TTS adaptive-based system with data not included in training progress (unseen-speaker).

Acknowledgment

The authors would like to thank AIMED Co., Ltd. for funding this research. This work also is supported by the National Science Project under number KC4.0/19-25.

References

1. J. Shen *et al.*, Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions, in *2018 IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*) (IEEE, 2018), pp. 4779–4783.
2. Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao and T.-Y. Liu, FastSpeech 2: Fast and high-quality end-to-end text to speech, arXiv:2006.04558.
3. J. Kim, J. Kong and J. Son, Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech, in *Int. Conf. Machine Learning* (PMLR, 2021), pp. 5530–5540.
4. W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman and J. Miller, Deep voice 3: 2000-speaker neural text-to-speech, *Proc. ICLR* (2018), pp. 214–217.
5. Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren and R. A. Saurous, Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis, in *Proc. 35th Int. Conf. Machine Learning*, eds. J. Dy and A. Krause, Proceedings of Machine Learning Research, Vol. 80 (PMLR, 2018), pp. 5180–5189.

6. T. Schultz, Speaker characteristics, in *Speaker Classification I* (Springer, 2007), pp. 53–54.
7. Y. Jia, Y. Zhang, R. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. Lopez Moreno and Y. Wu, Transfer learning from speaker verification to multi-speaker text-to-speech synthesis, in *Advances in Neural Information Processing Systems*, eds. S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett, Vol. 31 (Curran Associates, Inc., 2018).
8. T. T. T. Nguyen, H. K. Nguyen, Q. M. Pham and D. M. Vu, Vietnamese text-to-speech shared task VLSP 2020: Remaining problems with state-of-the-art techniques, in *Proc. 7th Int. Workshop on Vietnamese Language and Speech Processing* (Association for Computational Linguistics, 2020), pp. 35–39.
9. T. T. Vu, M. C. Luong and S. Nakamura, An HMM-based Vietnamese speech synthesis system, in 2009 *Oriental COCOSDA Int. Conf. Speech Database and Assessments* (IEEE, 2009), pp. 116–121.
10. N. Tits, K. El Haddad and T. Dutoit, Exploring transfer learning for low resource emotional TTS, in *Proc. SAI Intelligent Systems Conf.* (Springer, 2019), pp. 52–60.
11. H. Hemati and D. Borth, Using IPA-based Tacotron for data efficient cross-lingual speaker adaptation and pronunciation enhancement, arXiv:2011.06392.
12. H. B. Moss, V. Aggarwal, N. Prateek, J. González and R. Barra-Chicote, BOFFIN TTS: Few-shot speaker adaptation by Bayesian optimization, in *ICASSP* 2020 — 2020 *IEEE Int. Conf. Acoustics*, *Speech and Signal Processing* (*ICASSP*) (IEEE, 2020), pp. 7639–7643.
13. C.-M. Chien, J.-H. Lin, C.-Y. Huang, P.-C. Hsu and H.-Y. Lee, Investigating on incorporating pretrained and learnable speaker representations for multi-speaker multi-style text-to-speech, in *ICASSP* 2021 — 2021 *IEEE Int. Conf. Acoustics*, *Speech and Signal Processing* (*ICASSP*) (IEEE, 2021), pp. 8588–8592.
14. Y. Lee and T. Kim, Robust and fine-grained prosody control of end-to-end speech synthesis, in *ICASSP* 2019 — 2019 *IEEE Int. Conf. Acoustics*, *Speech and Signal Processing* (*ICASSP*) (IEEE, 2019), pp. 5911–5915.
15. G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen and Y. Wu, Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis, in *ICASSP* 2020 — 2020 *IEEE Int. Conf. Acoustics*, *Speech and Signal Processing* (*ICASSP*) (IEEE, 2020), pp. 6264–6268.
16. G. Sun, Y. Zhang, R. J. Weiss, Y. Cao, H. Zen, A. Rosenberg, B. Ramabhadran and Y. Wu, Generating diverse and natural text-to-speech samples using a quantized fine-grained VAE and autoregressive prosody prior, in *ICASSP* 2020 — 2020 *IEEE Int. Conf. Acoustics*, *Speech and Signal Processing* (*ICASSP*) (IEEE, 2020), pp. 6699–6703.
17. E. Cooper, C.-I. Lai, Y. Yasuda, F. Fang, X. Wang, N. Chen and J. Yamagishi, Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings, in *ICASSP* 2020 — 2020 *IEEE Int. Conf. Acoustics*, *Speech and Signal Processing* (*ICASSP*) (IEEE, 2020), pp. 6184–6188.
18. S.-A. Rebuffi, A. Kolesnikov, G. Sperl and C. H. Lampert, iCaRL: Incremental classifier and representation learning, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (IEEE Computer Society, 2017), pp. 2001–2010.
19. J. Kirkpatrick *et al.*, Overcoming catastrophic forgetting in neural networks, *Proc. Natl. Acad. Sci. USA* **114**(13) (2017) 3521–3526.
20. P. N. Phuong, C. T. Quang, Q. T. Do and M. C. Luong, A study on neural-network-based text-to-speech adaptation techniques for Vietnamese, in 2021 24*th Conf. Oriental COCOSDA Int. Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques* (*O-COCOSDA*) (IEEE, 2021), pp. 199–205.

21. D. Snyder, D. Garcia-Romero, G. Sell, D. Povey and S. Khudanpur, X-vectors: Robust DNN embeddings for speaker recognition, in 2018 *IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*) (IEEE, 2018), pp. 5329–5333.
22. W. Xie, A. Nagrani, J. S. Chung and A. Zisserman, Utterance-level aggregation for speaker recognition in the wild, in *ICASSP* 2019 — 2019 *IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*) (IEEE, 2019), pp. 5791–5795.
23. S. Arik, J. Chen, K. Peng, W. Ping and Y. Zhou, Neural voice cloning with a few samples, in *Advances in Neural Information Processing Systems* **31** (2018) 10019–10029.
24. D. Min, D. B. Lee, E. Yang and S. J. Hwang, Meta-StyleSpeech: Multi-speaker adaptive text-to-speech generation, in *Int. Conf. Machine Learning* (PMLR, 2021), pp. 7748–7759.
25. B. Pfülb, A. Gepperth, S. Abdullah and A. Kilian, Catastrophic forgetting: Still a problem for DNNs, in *Int. Conf. Artificial Neural Networks* (Springer, 2018), pp. 487–497.
26. H. Hemati and D. Borth, Continual speaker adaptation for text-to-speech synthesis, arXiv:2103.14512.
27. Z. Li and D. Hoiem, Learning without forgetting, *IEEE Trans. Pattern Anal. Mach. Intell.* **40**(12) (2017) 2935–2947.
28. A. Mallya, D. Davis and S. Lazebnik, Piggyback: Adapting a single network to multiple tasks by learning to mask weights, in *Proc. European Conf. Computer Vision* (*ECCV*) (Springer International Publishing, 2018), pp. 67–82.
29. Z. Jiang, Y. Ren, M. Lei and Z. Zhao, FedSpeech: Federated text-to-speech with continual learning, arXiv:2110.07216.
30. S.-F. Huang, C.-J. Lin, D.-R. Liu, Y.-C. Chen and H.-Y. Lee, Meta-TTS: Meta-learning for few-shot speaker adaptive text-to-speech, *IEEE/ACM Trans. Audio, Speech, Lang. Process.* **30** (2022) 1558–1571.
31. Q. Xie *et al.*, The multi-speaker multi-style voice cloning challenge 2021, in *ICASSP* 2021 — 2021 *IEEE Int. Conf. Acoustics, Speech and Signal Processing* (*ICASSP*) (IEEE, 2021), pp. 8613–8617.
32. Y. Liu, L. He, J. Liu and M. T. Johnson, Introducing phonetic information to speaker embedding for speaker verification, *EURASIP J. Audio Speech Music Process.* **2019**(1) (2019) 1–17.
33. D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel and S. Khudanpur, Deep neural network-based speaker embeddings for end-to-end speaker verification, in 2016 *IEEE Spoken Language Technology Workshop* (*SLT*) (IEEE, 2016), pp. 165–170.
34. Y. Kwon, J.-W. Jung, H.-S. Heo, Y. J. Kim, B.-J. Lee and J. S. Chung, Adapting speaker embeddings for speaker diarisation, arXiv:2104.02879.
35. M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner and M. Sonderegger, Montreal Forced Aligner: Trainable text-speech alignment using Kaldi, in *Interspeech*, Vol. 2017 (International Speech Communication Association (ISCA), 2017), pp. 498–502.
36. S. Schneider, A. Baevski, R. Collobert and M. Auli, wav2vec: Unsupervised pre-training for speech recognition, arXiv:1904.05862.
37. J. Kominek, T. Schultz and A. W. Black, Synthesizer voice quality of new languages calibrated with mean mel cepstral distortion, in *Spoken Languages Technologies for Under-Resourced Languages* (2008), pp. 63–68.
38. L. Van der Maaten and G. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.* **9**(11) (2008) 2579–2605.

A STUDY ON NEURAL-NETWORK-BASED TEXT-TO-SPEECH ADAPTATION TECHNIQUES FOR VIETNAMESE

Pham Ngoc Phuong[1], *Chung Tran Quang*[2], *Quoc Truong Do*[2], *Mai Chi Luong*[3]

[1]Thai Nguyen University, Vietnam
[2]Vietnam Artificial Intelligence Solution, VAIS
[3]Institute of Information Technology, Vietnam Academy of Science and Technology, Vietnam
phuongpn@tnu.edu.vn, chungtran@vais.vn, truongdo@vais.vn, lcmai@ioit.ac.vn

ABSTRACT

One of the main goals of text-to-speech adaptation techniques is to produce a model that can generate good quality audio given a small amount of training data. In fact, TTS systems for rich-resource languages have good quality because of a large amount of data, but training models with small datasets (or low-resources) is not an easy task, which often produces low-quality sounds. One of the approaches to overcome the data limitation is fine-tuning. However, we still need a pre-trained model which learns from large amount of data in advance. The paper presents two contributions: (1) a study on the amounts of data needed for a traditional fine-tuning method for Vietnamese, where we change the data and run the training for a few more iterations; (2) we present a new fine-tuning pipeline which allows us to borrow a pre-trained model from English and adapt it to any Vietnamese voices with a very small amount of data while still maintaining a good speech synthetic sound. Our experiments show that with only 4 minutes of data, we can synthesize a new voice with a good similarity score, and with 16 minutes of data, the model can generate audio with a 3.8 MOS score.

Index Terms— Speaker adaptation, Multi-pass fine-tune, TTS adaptation, Vietnamese TTS corpus

1. INTRODUCTION

Text-to-speech (TTS) based on neural network has thrived and achieved the quality of synthesis as good as human voice. This is because neural network-based speech synthesis techniques have developed strongly and many research works focused on different aspects of neural network-based TTS [1, 2, 3, 4, 5, 6, 7]. Neural network based TTS can be grouped into two categories: autoregression and non-autoregression with different advantages and disadvantages [8]. Autoregressive TTS models (Tacotron[3], Tacotron2 [4], Transformer TTS [6]) generate Mel-spectrogram autoregressively. These models achieved high synthesis quality but slow inferring speed (especially long sentences) and robustness synthesis voice (mispronouncing, skipping, or repeating words). Non-autoregressive TTS models (FastSpeech[7], Flow-TTS[9], GlowTTS[10]) are designed to reduce failure cases of synthesis issues. They allow to generate Mel-spectrogram at high speed and avoid robustness issues by generating sequences in parallel without explicitly depending on the previous elements, which can significant speed up the inference process. Recently, ForwardTacotron has shown improvement in terms of acceleration of the inference phase, by taking advantage of Tacotron and Fastspeech. The ForwardTacotron modifies the Tacotron to generate speech in a single forward pass and use a duration predictor to align text and generate mel spectrograms. But there are still many challenges in synthesizing new voices with small amounts of data to suit with low-resource languages [11]. To address this, the widely accepted efficient low data adaptation techniques are proposed in the literature [12]. This is how to use the minimum amount of target voice speech data (seen speaker adaptive data) to learn acoustic and prosody features to synthesize into a new voice with maximal similarity but still ensure the quality [13, 14, 15, 16, 17, 18]. There are many studies on speech synthesis with rich-resource languages such as English and Chinese that allow synthesis of new voice with small amount of data based on adaptive techniques. However, no study has evaluated exact number of minimum data samples to synthesize a new voice for Vietnamese. In addition, Vietnamese is a low-resource language, with almost no high-quality recorded dataset over 10 hours [19]. Meanwhile, training the TTS model (e.g. Tacotron2, FastSpeech2) requires at least 15 hours of Vietnamese high quality data (recorded in the studio). Collecting such a large amount of data is expensive, time consuming and impossible to include new speakers. Therefore, it is necessary to have a strategy to self-build TTS datasets from free quality recording sources. However, these sources are often not fully transcribed or properly labeled, which is a major challenge in the development of TTS corpus. This raises another challenge in the adaptation strategy, which is data adaptation on untranscribed data [20, 21, 22]. Previous works [23] have investigated the possibility to adapt

Thanks to VAIS .,JSC for funding.

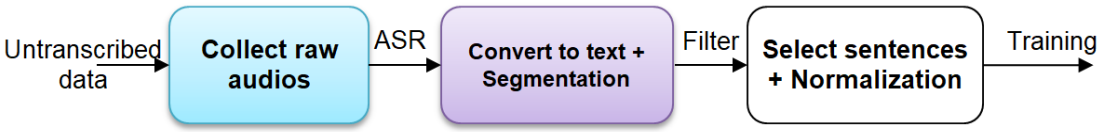


Figure 1: Building dataset strategy for TTS baseline model

a general TTS model to emotional TTS by fine-tuning a neutral TTS model with a small amount of emotional dataset. Although using the same English language, the adaptive dataset needs to be quite large (40 minutes). Or the study [24] demonstrated that using Tacotron2 for data efficient cross-lingual speaker adaptation and can transfer an existing TTS model to a new speaker with the same or different language using 20 minutes of data. However, by applying traditional fine-tuning, the results have low quality and low similarity. This paper presents two main contributions:
- A study on the amount of data needed for a traditional fine-tuning method for Vietnamese and a low-cost method to build a TTS baseline dataset from untranscribed data by using ASR models for transcribing speech data and using labeled data pairs for speech adaptation.
- A Vietnamese speaker adaptation method to synthesize new voice with small adaptation data by applying simple changes to Taocotron2 model to multi-pass fine-tuning.

2. DATA PROCESSING

We built two datasets: 1. Dataset to train the TTS baseline model, this is the large Vietnamese data of single speaker; 2. The adaptation dataset is divided into several sub-datasets of various sizes to evaluate the quality of the adaptive system.

2.1. Building Vietnamese dataset for TTS baseline

Collecting data to train the TTS baseline model using traditional methods is time-consuming and costly (selecting text, selecting speaker, recording...)[25]. However, this is an impossible task when we want to quickly build new voices on demand (because the speaker does not have enough time to record). The data collection strategy depicted in Figure 1 is as follows:
- Collect raw audio: From pre-recorded audio sources with standard quality and clear sound recorded in a studio with a minimum of 20 hours recording. The source of the sound we get is from a long story reading or the voice of an announcer or an MC on TV or radio.
- Convert audio to text and segmentation: We use a Vietnamese ASR system to convert raw audio to text by cutting the audio files into segments with a length of 3-10 seconds (depending on the pause of silence) and converting to text corresponding to each segment. Audio tracks with noise or unclear pronunciation are automatically removed. Only audio segments with clear text are retained (choose ASR confidence above 95%)[26].
- Select text sentences and normalize audio: We filter the recognized text sentences to keep the smallest set of cover sentences but contain the most number of syllables in Vietnamese. Only keep the audio set about 15 hours in size. This size covers quite a lot of syllables and it is efficient enough in order to create the synthesized voice for Vietnamese. Before training, it is necessary to normalize audio to .wav file format; sample rate is 20500 Hz; mono channels. To reduce noise, pre-training audio is trimmed with silence at the start and end of each file, then reduced by 50% in volume.

2.2. Building experimental Vietnamese dataset for TTS adaptation

To build small datasets to assess adaptive quality, we firstly divided the dataset into small sets and conducted a preliminary evaluation, then filtered and kept only representative datasets. We kept and divided the target speakers into sub-datasets with various sizes: 50, 200, 800 and 4500 sentences (corresponding to 4, 16, 60 minutes and 5 hours). From the target speaker's dataset, we adopted text selection based on greedy search to find the optimal sentences that have the best phonemic coverage [27].

2.3. Building phoneme level for Vietnamese

Phonologically, the Vietnamese language is monosyllabic. While there are around 19.000 pronounceable syllables in Vietnamese, only approximately 7000 syllables (with and without tone) are usually used in daily conversations and newspapers [28]. Each Vietnamese syllable can be represented by three components which are consonant, vowel, and tone. In addition, the tone is split into 6 units based on pitch, length, melody, intensity, and phonation. In our research, there are 30 consonants and 15 vowels, and 6 tones. When we combined vowels with tones in order to represent the intonations of vowels, and as a result, we had a total of 90 different intonations of vowels. This means that we had 120 phonemes (30 consonants and 90 intonations of vowels) to express all Vietnamese syllables. For example, "trường học" (school) could split into two syllables "trường" and "học" like a form of collocation in English, and each syllable could be represented by the following phonemes shown in Table 1.

Table 1. Mapping Vietnamese syllables to phonemes

Syllable	Phoneme
trường	t͡ɕ iə-1 ɳɣ
học	h ɔ-5 kpz

3. MULTI-PASS FINE-TUNE FOR TTS ADAPTATION

Transfer learning is a typical approach to quickly learning parameters from pre-trained models. Transfer learning is used when your dataset has too little data to train a full-scale model from scratch. In speech synthesis, fewer data can be accommodated by fine-tuning in the two smaller sub-spaces, and the synthesized speech is more natural and similar. For TTS, previous works have concentrated on transfer learning and meta-learning methods for adapting new speakers [29, 30, 31]. A traditional approach is fine-tuning of a part or whole model with a pre-trained model using a small dataset of target speakers [23, 24]. When fine-tuning, the model only changes the decoder module's parameters. The model can learn the whole speaker's features from a small amount of data after fine-tuning.

The basic acoustic model of end-to-end TTS is a formula for transforming text information into acoustic features, and it was demonstrated that a common approach is to fine-tune the pre-trained multi-speaker acoustic model with the target speaker's corpus using Tacotron2. The method of which is shown on the left of Figure 2 [32]. However, with a fine-tune traditional approach, to create a new voice in a new language different from a pre-trained model still requires a large amount of data ($\geq$ 5 hours, which is difficult with low-resource languages). If we use too small amounts of data, it is easy to cause overfitting by adapting directly on the end-to-end acoustic model. To solve these issues, we propose a model to borrow an English pre-trained model and 1st fine-tune with an intermediate Vietnamese pre-trained model, then 2nd fine-tune with an adaptive voice. We refer to it as the "multi-pass fine-tune" method, and the same is shown in the diagram on the right of Figure 2, which requires only a small sample to adapt a new voice. Because the main acoustic features have been learned/transferred from the English (large dataset) and Vietnamese (medium dataset), to generate a new voice, we only need a small amount of adaptive data to model learning voice features of the target speaker. Transfer learning is also effectively applied to cross-lingual learning for low-resource languages, the target language and the source language must be different. The phoneme mapping model between source and target linguistic symbols according to word pronunciation. The speaker embedding vector of the old speaker has been frozen and only let the new speaker adapt their embedding to the TTS model. Thus, this multi-pass fine-tune technique also allows transfer learning for a new language.

4. EXPERIMENTAL SETUP

4.1. Dataset

To evaluate the minimum size and number of fine-tuning layers to create a new voice effectively, we use three types of datasets:

- *English dataset* : Using LSpeech-1.1 corpora, this is a public domain speech dataset consisting of 13,100 short audio clips, read by a single female speaker. Each clip is provided with a transcription. Clips vary in length from 1 to 10 seconds and have a total length of approximately 24 hours.
- *Intermediate Vietnamese dataset*: Using intermediate Vietnamese dataset built from Section 2.1. This dataset consists 15,125 short audio clips of a single speaker reading TV news with a male voice dataset. Clips vary in length from 1 to 10 seconds and have a total length of approximately 15 hours.
- *Adaptation datasets*: First, we use the techniques described in Section 2.1 to build an adaptive dataset of 5 hours of the female voice (corresponding to 4,544 sentences) to train the model from scratch. Then we split the dataset into 3 small sets with the number of sentences is 50, 200, 800 respectively (duration corresponding to 4, 16 and 60 minutes).

4.2. Experimental Setup

To evaluate the TTS system, we used Tacotron2 network and Waveglow vocoder. The Tacotron2 architecture consists of two distinct parts: 1. Spectrogram Prediction Network is used to convert text strings to mel-spectrograms in the frequency domain; 2. Vocoder - Transforms from mel-spectrogram to waveform. The architecture of Spectrogram Prediction Network is quite simple, including Encoder and Decoder connected by Location Sensitive Attention. The input of Vietnamese speech synthesis model uses phoneme level instead of character level[33]. An attention network consumed the encoder output, which summarized the entire encoded sequence as fixed-length context. From the encoded input sequence, the decoder predicts a mel-spectrogram. The mean squared error (MSE) loss function was utilized to minimize the output of the model with ground trust. Train the model with a warm-start from pre-trained models corresponding to the following experiments:

Experiment 1. Determine the minimum amount of Vietnamese data to train the Tacotron2 model from scratch and traditional fine-tune. The evaluation results are shown in Table 2:

- *Column 2*: Training model directly from scratch with adaptive data (without pre-trained model).
- *Column 3*: Training model from English pre-trained model with adaptive data (Pre-trained model using LJspeech 1.1 dataset).

Experiment 2. Evaluate the multi-pass fine-tune speaker adaptation model by training model from intermediate Vietnamese pre-trained model (which has been trained by fine-

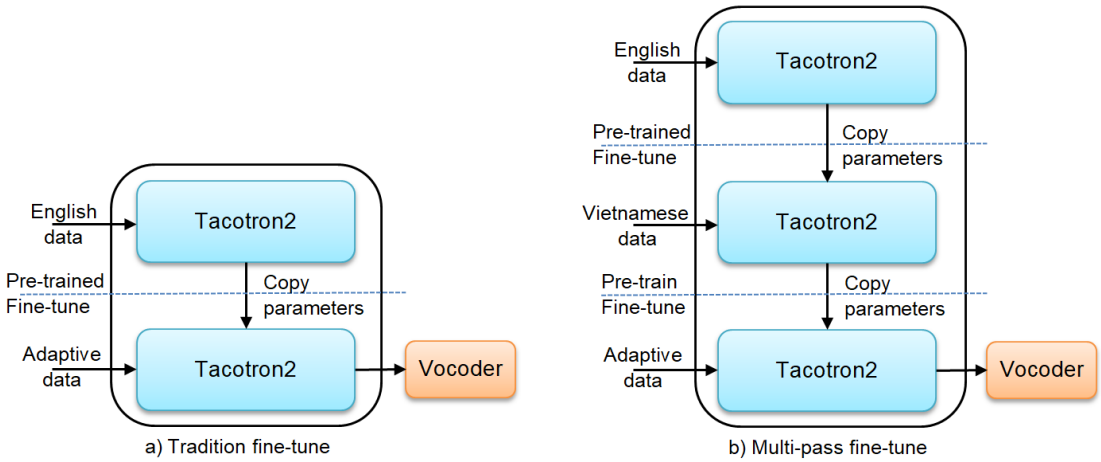


Figure 2: Speaker Adaptation a new voice with multi-pass fine-tune

Table 2. Multi-pass fine-tune and data-driven adaptive quality statistics table

Time	From scratch (VN corpora)	English pre-trained + Adaptive data	Intermediate VN pre-trained + Adaptive data
16 mins	1.29	1.33	**3.78**
60 mins	1.31	2.68	3.87
5 hours	2.66	N/A	N/A

tuning from the English pre-trained model). Advantages of adaptive training to create a new voice from adaptive data with a multi-pass fine-tune technique. The adaptation sample size varied from 16 minutes to 1 hour to assess the adaptation quality. The evaluation results are shown in columns 4 of Table 2.

Experiment 3. Evaluate the similarity between the Groundtruth voice and the adaptive voice of the traditional fine-tune model and the multi-pass fine-tune model with only 4 minutes of adaptive data. The evaluation results are shown in columns 2 and 3 of Table 3.

5. RESULT

5.1. Quality of traditional fine-tune model

We use the MOS scale to evaluate the quality of voice generated by the system. The Vietnamese speech synthesis system was evaluated by two groups of 23 listeners. Each listener will hear a set of 120 audio sentences that have been mixed between Groundtruth audios and audios generated from Traditional fine-tune, multi-pass fine-tune models and train from scratch models. The listeners will have five (5) options: 1. Bad; 2. Poor; 3. Fair; 4. Good; 5. Excellent, where 1 is the lowest perceived quality and 5 is the highest perceived quality. In the columns 2 and 3 of Table 2, we compare the sound quality (MOS) between the training model from scratch and the traditional fine-tune model:

- If we train Vietnamese dataset (VN corpora) from scratch, with 5 hours of the dataset, quality speech is still very poor (MOS=2.66). If we train for less than 1 hour, we will not be able to hear anything.
- If fine-tuned from an English pre-trained model with 1 hour of Vietnamese adaptation data, the quality will be as good as training from scratch of 5 hours of Vietnamese dataset, but the voice quality is still poor (MOS=2.68).

5.2. Quality of multi-pass fine-tune model

In the columns 4 of Table 2, based on the English pre-trained model, if fine-tune from intermediate Vietnamese dataset to small adaptive dataset, then only needs 16 minutes (200 sentences), the voice quality is quite good with MOS score of 3.78/4.69 of the human voice.

5.3. Similarity

We use Mel-Cepstral Distortion (MCD) to measure how different two sequences of Mel-cepstral are for evaluating voice conversion performance [34]. The smaller the MCD between synthesis and natural Mel cepstral sequences, the closer the synthesized speech is to reproducing natural speech.

We also use SIM (similarity of the voices) to evaluate similarity objectively. 11 professional listeners evaluate the similarity of 90 pairs of audio sentences (mixed between Groundtruth audios and audios generated by traditional fine-

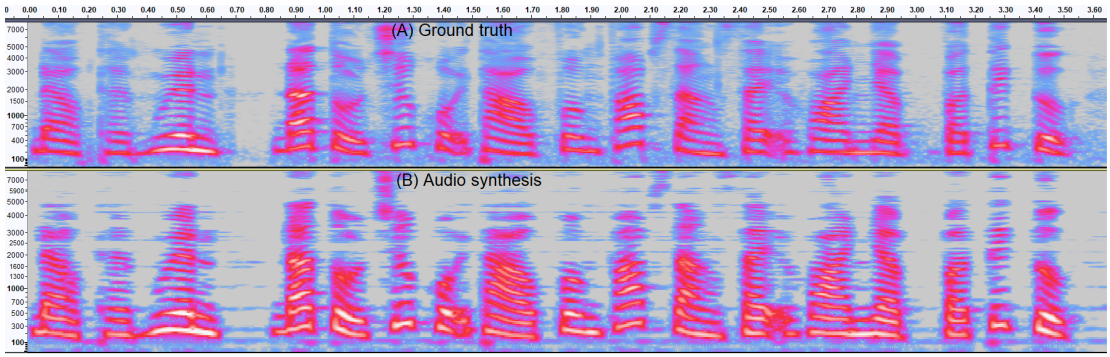


Figure 3: Speaker similarity of a adapted voice and Groundtruth with only 4 minutes of adaptive data

Table 3. Speaker similarity of tradition fine-tune and multi-pass fine-tune model compare to Groundtruth with only 4 minutes of adaptive data

Experiment - Model (4 mins adaptive data)	**MCD**	**SIM**
Groundtruth	-	3.99
Fine-tune trandition	10.65	1.13
Multi-pass fine-tune	**7.94**	**2.87**

tune and multi-pass fine-tune models). The listeners have four (4) options to give a score: 4. Definitely the same; 3. Maybe the same; 2. Maybe different; 1. Definitely different. The highest score is 3.99. The higher the score, the more significant, the more similarity between the synthetic voice and the Groundtruth.

Table 3 shows that multi-pass fine-tune creates new voices with a much lower MCD than traditional-fine-tune (reduced by 2.74). Also, in the same table, the results of Experiment 3 have shown that, with only 4 minutes of adaptive data, the multi-pass fine-tune model produced a synthesized voice with a much higher similarity than that of a synthetic voice from the traditional-fine-tune method (2.87/3.99 of Groundtruth).

Figure 3, Depicts the Mel-spectrogram of synthesized voice and the Groundtruth.

Reference utterance in Vietnamese: "Đại học Thái Nguyên đã tổ chức hội thảo phương pháp giảng dạy hòa nhập và tích cực".

Reference utterance in English: "Thai Nguyen University organized a workshop on inclusive and active teaching methods".

With only 4 minutes of adaptive data (corresponding to 50 sentences), the adapted voices showed similar of phonemic durations.

6. CONCLUSION AND FUTURE WORKS

We have demonstrated that if using traditional fine-tuning techniques, 1 hour of Vietnamese adaptive data is not enough to synthesize new voices; it requires a minimum of 3 hours. We also proposed simple changes to the basic Tacotron2 model with some multi-pass fine-tuning techniques to adapt new speakers to low-resource languages such as Vietnamese. We demonstrated that it only takes 4 minutes of adaptive data to generate a new voice with high similarity and only takes 16 minutes to generate a good quality voice. In addition, we also presented a low-cost method to build datasets for TTS systems from unlabeled data, which are available on the Internet by combining them with a good ASR system and a data collection and filtration strategy.

We also demonstrated that this adaptation resulted in better pronunciation of words in the target language for source speakers, and that it may be expanded to more languages in a variety of ways. We plan to study the efficient adaptation techniques in generating new speakers from multi-speaker and unseen speakers for future work.

7. REFERENCES

[1] Heiga Ze, Andrew Senior, and Mike Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 ieee international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 7962–7966.

[2] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[3] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: Towards end-to-end speech synthesis," *arXiv preprint arXiv:1703.10135*, 2017.

[4] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng

Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al., "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4779–4783.

[5] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan Ömer Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller, "Deep voice 3: 2000-speaker neural text-to-speech," 2017.

[6] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu, "Neural speech synthesis with transformer network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, pp. 6706–6713.

[7] Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu, "Fastspeech: Fast, robust and controllable text to speech," *arXiv preprint arXiv:1905.09263*, 2019.

[8] Kainan Peng, Wei Ping, Zhao Song, and Kexin Zhao, "Non-autoregressive neural text-to-speech," in *International conference on machine learning*. PMLR, 2020, pp. 7586–7598.

[9] Chenfeng Miao, Shuang Liang, Minchuan Chen, Jun Ma, Shaojun Wang, and Jing Xiao, "Flow-tts: A non-autoregressive network for text to speech based on flow," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7209–7213.

[10] Jaehyeon Kim, Sungwon Kim, Jungil Kong, and Sungroh Yoon, "Glow-tts: A generative flow for text-to-speech via monotonic alignment search," *arXiv preprint arXiv:2005.11129*, 2020.

[11] Gopala Krishna Anumanchipalli and Alan W Black, "Adaptation techniques for speech synthesis in under-resourced languages," in *Spoken Languages Technologies for Under-Resourced Languages*, 2010.

[12] Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu, "A survey on neural speech synthesis," *arXiv preprint arXiv:2106.15561*, 2021.

[13] Zewang Zhang, Qiao Tian, Heng Lu, Ling-Hui Chen, and Shan Liu, "Adadurian: Few-shot adaptation for neural text-to-speech with durian," *arXiv preprint arXiv:2005.05642*, 2020.

[14] Zvi Kons, Slava Shechtman, Alex Sorin, Carmel Rabinovitz, and Ron Hoory, "High quality, lightweight and adaptable tts using lpcnet," *arXiv preprint arXiv:1905.00590*, 2019.

[15] Henry B Moss, Vatsal Aggarwal, Nishant Prateek, Javier González, and Roberto Barra-Chicote, "Boffin tts: Few-shot speaker adaptation by bayesian optimization," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7639–7643.

[16] Seungwoo Choi, Seungju Han, Dongyoung Kim, and Sungjoo Ha, "Attentron: Few-shot text-to-speech utilizing attention-based variable-length embedding," *arXiv preprint arXiv:2005.08484*, 2020.

[17] Mingjian Chen, Xu Tan, Bohan Li, Yanqing Liu, Tao Qin, Sheng Zhao, and Tie-Yan Liu, "Adaspeech: Adaptive text to speech for custom voice," *arXiv preprint arXiv:2103.00993*, 2021.

[18] Dongchan Min, Dong Bok Lee, Eunho Yang, and Sung Ju Hwang, "Meta-stylespeech: Multi-speaker adaptive text-to-speech generation," *arXiv preprint arXiv:2106.03153*, 2021.

[19] Thi Thu Trang Nguyen, Hoang Ky Nguyen, Quang Minh Pham, and Duy Manh Vu, "Vietnamese text-to-speech shared task vlsp 2020: Remaining problems with state-of-the-art techniques," in *Proceedings of the 7th International Workshop on Vietnamese Language and Speech Processing*, 2020, pp. 35–39.

[20] Yuzi Yan, Xu Tan, Bohan Li, Tao Qin, Sheng Zhao, Yuan Shen, and Tie-Yan Liu, "Adaspeech 2: Adaptive text to speech with untranscribed data," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6613–6617.

[21] Katsuki Inoue, Sunao Hara, Masanobu Abe, Tomoki Hayashi, Ryuichi Yamamoto, and Shinji Watanabe, "Semi-supervised speaker adaptation for end-to-end speech synthesis with pretrained models," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7634–7638.

[22] Hieu-Thi Luong and Junichi Yamagishi, "Nautilus: a versatile voice cloning system," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2967–2981, 2020.

[23] Noé Tits, Kevin El Haddad, and Thierry Dutoit, "Exploring transfer learning for low resource emotional tts," in *Proceedings of SAI Intelligent Systems Conference*. Springer, 2019, pp. 52–60.

[24] Hamed Hemati and Damian Borth, "Using ipa-based tacotron for data efficient cross-lingual speaker adaptation and pronunciation enhancement," *arXiv preprint arXiv:2011.06392*, 2020.

[25] Pham Ngoc Phuong, Quoc Truong Do, and Luong Chi Mai, "A high quality and phonetic balanced speech corpus for vietnamese," *arXiv preprint arXiv:1904.05569*, 2019.

[26] Do Quoc Truong, Pham Ngoc Phuong, Hoang Tung Tran, and Luong Chi Mai, "Development of high-performance and large-scale vietnamese automatic speech recognition systems," *Journal of Computer Science and Cybernetics*, vol. 34, no. 4, pp. 335–348, 2018.

[27] J-s Zhang and Satoshi Nakamura, "An efficient algorithm to search for a minimum sentence set for collecting speech database," in *Proc. ICPhS*, 2003, pp. 3145–3148.

[28] Quoc Bao Nguyen, Tat Thang Vu, and Chi Mai Luong, "The effect of tone modeling in vietnamese lvcsr system," *Procedia Computer Science*, vol. 81, pp. 174–181, 2016, SLTU-2016 5th Workshop on Spoken Language Technologies for Under-resourced languages 09-12 May 2016 Yogyakarta, Indonesia.

[29] Ye Jia, Yu Zhang, Ron J Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," *arXiv preprint arXiv:1806.04558*, 2018.

[30] Yutian Chen, Yannis Assael, Brendan Shillingford, David Budden, Scott Reed, Heiga Zen, Quan Wang, Luis C Cobo, Andrew Trask, Ben Laurie, et al., "Sample efficient adaptive text-to-speech," *arXiv preprint arXiv:1809.10460*, 2018.

[31] Yu-An Chung, Yuxuan Wang, Wei-Ning Hsu, Yu Zhang, and RJ Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6940–6944.

[32] Tao Wang, Jianhua Tao, Ruibo Fu, Jiangyan Yi, Zhengqi Wen, and Rongxiu Zhong, "Spoken content and voice factorization for few-shot speaker adaptation.," in *INTERSPEECH*, 2020, pp. 796–800.

[33] Phuong Pham Ngoc, Chung Tran Quang, Quang Minh Nguyen, and Quoc Truong Do, "Improving prosodic phrasing of vietnamese text-to-speech systems," in *Proceedings of the 7th International Workshop on Vietnamese Language and Speech Processing*, 2020, pp. 19–23.

[34] John Kominek, Tanja Schultz, and Alan W Black, "Synthesizer voice quality of new languages calibrated with mean mel cepstral distortion," in *Spoken Languages Technologies for Under-Resourced Languages*, 2008.

Improving prosodic phrasing of Vietnamese text-to-speech systems

Pham Ngoc Phuong
Thai Nguyen University
phuongpn@tnu.edu.vn

Chung Tran Quang
Hanoi University of Science and Technology
chungtran@vais.vn

Quang Minh Nguyen
Vietnam Artificial Intelligence Solution
minhnq@vais.vn

Quoc Truong Do
Vietnam Artificial Intelligence Solution
truongdo@vais.vn

Abstract

End-to-end TTS architecture which is based on Tacotron2 is the state-of-art system. It breaks the traditional system framework to directly converts text input to speech output. Although it is shown that Tacotron2 is superior to traditional piping systems in terms of speech naturalness, it still has many defects in building Vietnamese TTS: 1) Not good at prosodic phrasing for long sentences, 2) Not good at expression for foreign words. In this paper, we used 2 methods to solve these defects: 1) Pause detection system for predicting and inserting punctuation into long sentences to improve speech naturalness. 2) Translation system for transcribing foreign words to Vietnamese words. In the VLSP 2020 evaluation campaign, our model achieved a mean opinion score (MOS) of 3.31/5 compared to 4.22/5 of humans.

***Index Terms*—** Text-to-speech, TTS, Vietnamese TTS, end-to-end speech synthesis

1 Introduction

Text-to-Speech (TTS) study is widely applied in real-life but it is still a challenge in the field of speech processing. Many techniques have been proposed such as concatenative synthesis (Hunt and Black, 1996), statistical parametric speech synthesis (SPSS). Although concatenative synthesis can reach highly natural synthesized speech, the approach is inherently limited by properties of the speech corpus used for the unit selection process. Meanwhile, SPSS allows product direct speech smoothly and intelligibly by a vocoder. A full SPSS system consists of text analysis, feature generation, and waveform generation modules a, some SPSS techniques are used for Vietnamese TTS: Hidden Markov model (HMM) (Tokuda et al., 2000), Deep neural networks(DNN) (Ze et al., 2013), generative adversarial networks (GAN)(Saito et al., 2017) and End-to-end architectures(Wang et al., 2017). Currently, DNN approaches have gradually replaced HMM models for the duration model and acoustics model. However, the generated voice is often muffled and becomes unnatural. Wavenet (Oord et al., 2016), Wave RNN (Kalchbrenner et al., 2018), GAN (Saito et al., 2017) produces audio with significantly improved naturalness but requirements deep experience and voices that are not as realistic as they are in reality. An end-to-end architecture (Tacontron 2 and WaveGlow vocoder) include five components: linguistic analysis, acoustic model, duration model, parameter generation, and post-filtering are replaced by encoder-attention-decoder networks (Wang et al., 2017; Shen et al., 2018), to be able to effectively optimize the mapping from input text to acoustic features. Finally, a neural vocoder such as Waveglow generated a waveform from the generated mel-spectrogram.

However, in a long sentences or long phrases, speech synthesis results will not be natural. This comes from the fact that human speakers usually break phrases by inserting word transitions instead of punctuation for the sake of expressivity, better comprehension or only taking a breath. The term phrasing is used to describe the phenomenon of grouping words into phrases and separating these phrases with pauses or punctuation inserts. In addition, there are many foreign words in the sentences that are not in the Vietnamese phonetic dictionary. If only replacing foreign words with International Phonetic Alphabet (IPA), the synthesized sentence will not be pronounced in Vietnamese standard. In this paper, 2 methods are applied to synthesize sen-

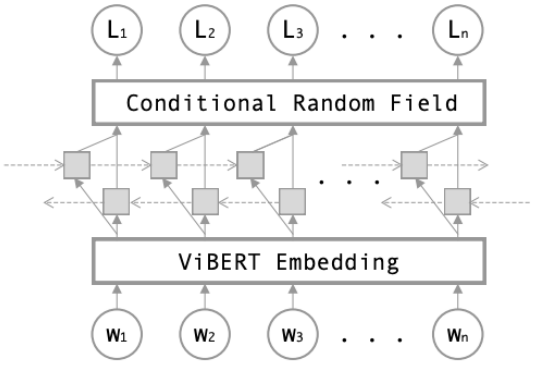


Figure 1: The CaPu model insert the punctuation into the sentences.

tences more naturally: 1) Pause detection module will insert punctuation into sentences to improve prosody of the TTS system, 2) Translation module will transforms foreign words into the Vietnamese standard pronunciation word.

2 Prosodic and pronunciation modeling

2.1 Prosodic modeling

When reading long sentences, the reader always stops at the punctuation or at the position of two or more words of equal syntactic importance (such as noun, verb, etc). So, pause prosodic detection is extremely important affecting the prosody of the TTS system. However, the provided data from the VLSP organization (Trang et al., 2020) was the result of the ASR system, so it had the text only. The synthetic sound quality of the deep neural network depends on the input data. Thus, adding the punctuation at a suitable position can enhance the prosody of our system. To solve this challenging problem, we integrate the Capitalization and Punctuation (CaPu) model (Nguyen et al., 2020) to recover the punctuation of the sentences. The CaPu model not only inserts the punctuation automatically to correct the text format but also places the punctuation at the location relating to breathing.

The CaPu model includes three components that is the embedding layer, the recurrent layer, and the classification layer. More specifically, the embedding layers is ViBERT model that embedded the input sentences to the fixed vectors. The fixed vectors passed through the bidirectional GRU layers. followed by the conditional random field layer to classify the punctuation-tag of each input word. ViBERT is a variation of RoBERTa$_{base}$ model with fewer layers than the original model, it contains 4 encoder layers, the number of heads is 4 and the hidden dimension size is 512. The model has 4 bidirectional GRU layers, the hidden size of GRU cell is 512. The figure 1 depicts CaPu architecture.

To train CaPu model, we collected a huge of text from many domains on the internet including wikipedia, law, politics, etc. This document has the punctuation in accordance with Vietnamese standard style. To mimic the pause of the reader, we use word time-stamp of the ASR system. If the silent time is more than 0.3 second, we put the commas at this silent position. Finally, we trained the CaPu model with the processed data. As a result, CaPu model can insert the punctuation at the proper location by 2 strategy, Vietnamese standard and reader style. Besides, we also added a dot at the end of transcript text to present the end of audio. The result of the CaPu model:

Raw transcript:

cảm giác đó đến một cách đột ngột nhưng mụ xua đuổi nó đi không cho nó chạm tới mụ cũng như không để cho nó chạm tới nền cộng hòa

After add commas to transcript:

cảm giác đó đến một cách đột ngột **,** nhưng mụ xua đuổi nó đi **,** không cho nó chạm tới mụ **,** cũng như không để cho nó chạm tới nền cộng hòa **.**

2.2 Pronunciation modeling

One of the biggest challenges for the VLSP Text-To-Speech (Trang et al., 2020) is that the transcript text has many foreign words. Because foreign words are out of the Vietnamese vocabulary and can not convert to the phoneme directly. This leads to trouble for the participants when joining and building the Vietnamese TTS system. To handle and tackle this problem, we used Vietnamese sound to pronounce these English words. For example, “kuttner” will be pronounced by “cắt nơ”, seeing more examples in Table 1. In order to transform from foreign words to Vietnamese words, we used the popular translation model-Transformer$_{base}$ (Vaswani et al., 2017) model.

The Transformer architecture has two modules, the encoder, and the decoder, and 2 component is connected through an attention mechanism. The Transformer model that we used for this challenge is composed of a stack of N=6 identical layers for both the encoder and decoder.

To train this translation model, we must create a large number of pair of English-Vietnamese words. The total dataset that we produced is more than 1

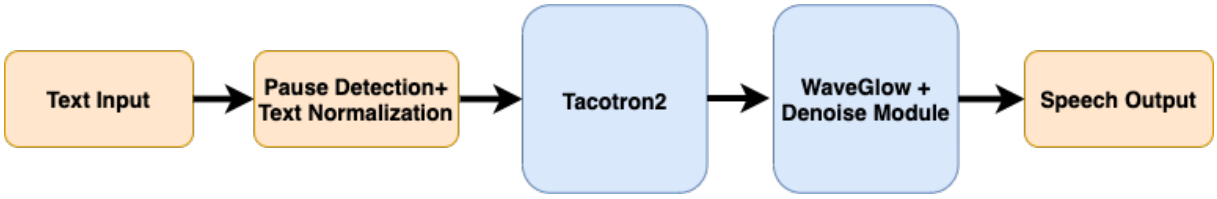


Figure 2: Our TTS pipeline, the input text passes to the pause detection and text normalization module. Subsequently, the processed data passes to Tacotron2 and WaveGlow to generate speech synthesis

English word	Vietnamese word
kuttner	cắt nơ
Anderson	an đơ sơn
vera	vê ra
reme	rê mi

Table 1: Convert English words to Vietnamese words

hundred million pairs. The result of the translation model was displayed in Table 1.

3 Text-to-Speech System

Nowadays, for the TTS task, the end-to-end speech synthesis pipeline consists of two phases, 1. converting text to Mel-spectrogram and 2. converting Mel-spectrogram to waveform synthesis. The model Tacotron2 combining with WaveGlow vocoder is still state-of-the-art for the TTS task. Tacotron2 is a deep neural network receiving a text to predict Mel-spectrogram signal. Then Mel-spectrogram will be converted to waveform thanks to WaveGlow. However, we realized that synthetic speech was noisy. Therefore, we used a denoiser model, attaching at the end of the WaveGlow model.

- *Tacotron2*: The network has two components an encoder and a decoder. We had a small change comparing with the original model. To adapt to the characteristic of the Vietnamese language, the input model was phoneme level instead of character level. Phoneme character passed to the embedding layer, which represented by 512-dimensional. Afterward, these vectors passed through a stack of 3 convolutional layers, followed by single bi-directional LSTM layers to generate the encoded features. The encoder output was consumed by an attention network which yielded a fixed-dimensional vector. Finally, the decoder had the mission of converting this vector to a Mel-spectrogram. To train the Tacotron2 model, we minimized the output of the model with ground trust using mean squared error(MSE).

- *WaveGlow*: The network that we used for the TTS challenge was similar to the original model. The model transformed the output of the Tacotron2 model to the waveform signals. WaveGlow is deployed using only a single network and single cost function, so it is fast, efficient and can produce high quality audio synthesis. The network has 12 coupling layers and 12 invertible 1 x 1 convolutions. In coupling module has 8 layers of dilated convolutions with 512 channels used as residual connections and 256 channels in the skip connection. For the challenge, we used the pre-trained model provided by the author to synthesize the audio.

- *Denoise Module*: This module will reduce the noise of synthetic audio generated from WaveGlow. Firstly, we produced bias audio by using WaveGlow infer a zero Mel-spectrogram with shape 1x80x88. Then both synthetic audio and bias audio will be transformed to Mel-spectrogram by the short-time Fourier transform method. Next, we used the synthetic Mel-spectrogram minus the bias Mel-spectrogram. As a result, we received the final Mel-spectrogram and we used the inverse Fourier transform function to convert it back to audio.

4 Experimental Setup

4.1 Dataset

The duration of the training dataset is about 5-6 hours of a single female speaker and has 7770 audio files. The duration of each file is from 2s to 11s. The sample rate is 44100Hz, 2 channels. To train the model, we resampled to a sample rate of 20500Hz and also convert it to mono channel (1 channel). Besides, we decreased the volume of each file audio by 50%. To reduce noise for the training data, audio in training dataset will be trimmed the silence at start and end position. All transcript text in the dataset is spelled out, for example, “30” is written as “ba mươi”.

Data Processing	Evaluation
No	Speech synthesis can not read the foreign words, the pause in the sentences is unnatural
Pause detection	Speech synthesis can pause at the punctuation correctly, prosody seem naturally
Pause detection + Text Normalization	Speech synthesis can pronounce foreign words.

Table 2: Data processing and evaluate the system

4.2 Experimental Setup

Both CaPu and translation model were implemented by Fairseq (Ott et al., 2019) framework. We used Adam optimizer with beta factor (0.9, 0.98), the learning rate of 0.0005. Conditional Random Field (CRF) loss was applied to train the model and the learning rate scheduler was the inverse square root. The warm-up initial learning rate is 1e-7, and the batch size is 64.

To train the Tacotron2 model, we use GeForce RTX 2080 Ti, 11GB, the learning rate is 1e-3, the weight decay is 1e-6 , the batch size is 64. Adam optimizer with β_1=0.9 and β_2=0.999, $epsilon$=1e-6.

5 Result

We used Tacotron2+Waveglow to evaluate the TTS system. We conducted many experiments relating to data processing, see Table 2 for more detail. Finally, when we combined 2 methods processing pause detection and text normalization, the TTS system yielded speech synthesis naturally. Not only prosody seem natural, but also our system can pronounce foreign words similar to Vietnamese people.

MOS was applied to evaluate the system. The speech synthesis was evaluated by three groups of listeners: speech experts, volunteers, and undergraduates. The listeners will have 5 options to give a score from 1-5: excellent(5), good(4), fair(3), poor(2), 1(bad).

In the VLSP 2020's challenge, as shown in Table 3, our architecture achieved a MOS of 3.31 for the naturalness. For intelligibility, the rate of hearing correct words is 83.10% and the rate of listening to correct syllabi's is 82.90%

	MOS
Our system	3.31
Human	4.22

Table 3: MOS Result for the VLSP Dataset

6 Conclusion and future works

In this paper, we describe our architecture for the Vietnamese Text-to-speech system. For the data from an organization, our approach yielded a MOS of 3.31. By conducting many experiments, we realized that data processing is very important in this challenge. By converting English words to Vietnamese words, also add commas to transcript text, these techniques assist model producing utterance synthesis very naturally.

In the future, we can experiment with more state-of-the-art architecture such as Hifi-Gan, Mel-Gan, Glow-TTS. Also, exploring many challenges of TTS such as how to training TTS with small data, TTS adaptation, etc.

References

Andrew J Hunt and Alan W Black. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376. IEEE.

Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. 2018. Efficient neural audio synthesis. *arXiv preprint arXiv:1802.08435*.

Thai Binh Nguyen, Quang Minh Nguyen, Thi Thu Hien Nguyen, Quoc Truong Do, and Chi Mai Luong. 2020. Improving vietnamese named entity recognition from speech using word capitalization and punctuation recovery models. *Proc. Interspeech 2020*, pages 4263–4267.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2017. Statistical parametric speech synthesis incorporating generative adversarial networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):84–96.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE.

Keiichi Tokuda, Takayoshi Yoshimura, Takashi Masuko, Takao Kobayashi, and Tadashi Kitamura. 2000. Speech parameter generation algorithms for hmm-based speech synthesis. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1315–1318. IEEE.

Nguyen Thi Thu Trang, Nguyen Hoang Ky, Pham Quang Minh, and Vu Duy Manh. 2020. Remaining problems with state-of-the-art techniques in proceedings of the seventh international workshop on vietnamese language and speech processing.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. 2017. Tacotron: Afully end-to-end text-to-speech synthesis model. *arXiv preprint arXiv:1703.10135*.

Heiga Ze, Andrew Senior, and Mike Schuster. 2013. Statistical parametric speech synthesis using deep neural networks. In *2013 ieee international conference on acoustics, speech and signal processing*, pages 7962–7966. IEEE.

Fast and Accurate Capitalization and Punctuation for Automatic Speech Recognition Using Transformer and Chunk Merging

Binh Nguyen[1,5], *Vu Bao Hung Nguyen*[1], *Hien Nguyen*[1,3], *Pham Ngoc Phuong*[1,3], *The-Loc Nguyen*[1,4], *Quoc Truong Do*[1], *Luong Chi Mai*[1,2]

[1]Vietnam Artificial Intelligence System, Vietnam
[2]University of Science and Technology of Hanoi, Vietnam
[3]Thai Nguyen University, Vietnam
[4]Hanoi University of Mining and Geology, Vietnam
[5]Hanoi University of Science and Technology, Vietnam

{binhnguyen|hungnguyen|locnguyen|truongdo}@vais.vn, nguyenthuhien@dhsptn.edu.vn, phuongpn@tnu.edu.vn, lcmai@ioit.ac.vn

Abstract

In recent years, studies on automatic speech recognition (ASR) have shown outstanding results that reach human parity on short speech segments. However, there are still difficulties in standardizing the output of ASR such as capitalization and punctuation restoration for long-speech transcription. The problems obstruct readers to understand the ASR output semantically and also cause difficulties for natural language processing models such as NER, POS and semantic parsing. In this paper, we propose a method to restore the punctuation and capitalization for long-speech ASR transcription. The method is based on Transformer models and chunk merging that allows us to (1), build a single model that performs punctuation and capitalization in one go, and (2), perform decoding in parallel while improving the prediction accuracy. Experiments on British National Corpus showed that the proposed approach outperforms existing methods in both accuracy and decoding speed.

Index Terms: speech recognition, capitalization and punctuation insertion

1. Introduction

In a typical setup of an ASR system, punctuation and capitalization of words are removed because they do not affect the pronunciation of words. As the result, the output of ASR contains purely a sequence of words or alphabet characters depending on the model type. While this output is sufficient for many applications, such as voice commands, virtual assistants, where speech segments are usually short and independent, it is difficult to be used in applications that transcribes long speech segments. It would be easier for human to read a document with proper punctuation and word capitalization. Moreover, when ASR results are fed into NLP models to perform machine translation (MT) or name entity recognition (NER), punctuation and word capitalization are crucial pieces of information that can help to boost the performance [1, 2, 3].

Regarding studies on segmentation and punctuation insertion for ASR, Cho et al. [1] proposed a method to use phrase-based translation models that consider the punctuation insertion as machine translation tasks. The model takes input is unpuctuted text and translates into a punctuated one. Zelasko et al. [4] and Tilk et al. [5] incoporate more information from speech signal to improve the performance. In [6, 7], dynamic conditional random fields (CRFs) [8] were used to predict punctuation. The works proposed by Cho et al. [9] and Tilk et al. [5] made use of end-to-end translation model with LSTM to predict punctuation and segmentation. They successfully demonstrated that the end-to-end models outperform conventional approaches. While existing works are capable of predicting punctuation, they share similar limitation. First, the models only handle one task which is punctuation insertion, however, output from ASR is also typically uncapitalized. While adding just punctuation might help speech translation to determine when to translate, other NLP tasks such as NER and PoS tagging do not get much help because one of the key feature of these models is word capitalization. Second, long input sentences are usually split into fix-length and non-overlapped chunks before feeding into the model. Although this method helps to speedup the inference by processing chunks independently and in parallel, it is prone to bad prediction of words around the chunk's boundary because there isn't enough both left and right context information in the area.

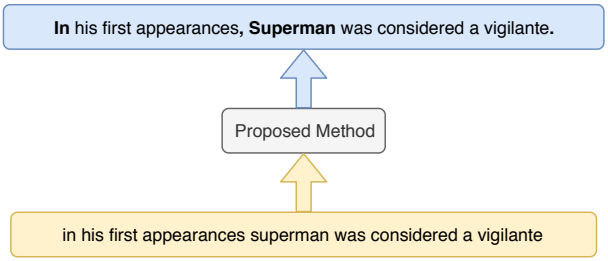
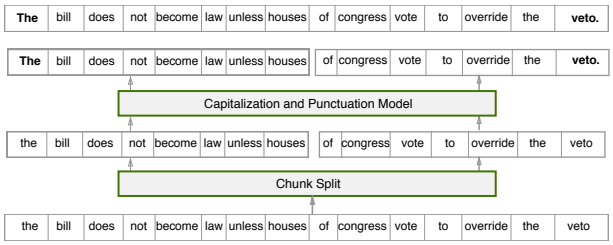
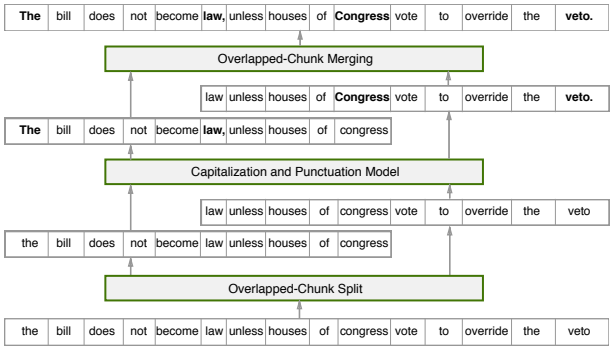


Figure 1: *The proposed method for performing both punctuation and word capitalization in one go*

In this paper, we proposed a method based on transformer models and overlapped chunk-merging to restore both word capitalization and punctuation in one go as illustrated in Figure 1. The system consists of 3 components (Figure 2 - b). The first component is an overlapped chunk spliting that takes a long input sequence and splits them into chunks with overlap. This process make sure that the second component, which is the capitalization and punctuation model, always have enough left and right context of words to make the prediction. The last component is the chunk-merging where the overlapped output are combined into a single sentence. This process decides which part of the overlap area to be removed and to be kept. The method allows us to (1), build a single model that performs



(a) *Capitalization and Punctuation System Without Overlapping Segments*



(b) *Proposed System Architecture for Capitalization and Punctuation. Because of more context, it can add comma after "law" and upper case "congress"*

Figure 2: *Capitalization and Punctuation System With and Without Overlap-ping Segments. Ground truth of this example is "The bill does not become law, unless Congress vote to override the veto."*

punctuation and capitalization without the need of pipeline results from one system to another, and (2), perform decoding in parallel while improving the prediction accuracy.

2. End-to-end Model for Punctuation and Segmentation

End2end models for punctuation works in a similar way with machine translation tasks [10, 11] where it takes input is a sequence of of lowercase, unpunctuated words and outputs a sequence with truecase and punctuation inserted. Figure 2a illustrates the use of end-to-end models for restoring capitalization and punctuation proposed in [12]. First, a long input text from ASR is split into small segments and then, they are fed into a translation model to produce an output sequence. While the approach can take advantages of LSTM models that it is able to learn longer context information, it usually failed to predict truecase or punctuation of words near the segment boundary.

Previous studies [13] has pointed out that Transformer performs better than LSTM models by exploiting its self-attention layer to capture context more efficiently and speedup the training process. Transformer is basically an encoder-decoder model. It contains multiple identical encoders and identical decoders stacked upon each other. Each encoder has a self-attention layer that extract surrounding words information when a word is being encoded. This layer is followed by a feed forward neural network; the networks in different encoders do not share weights. Each decoder also has a self-attention layer and a feed forward neural network, but to enhance the relevant parts of input, an attention layer (similar to attention in sequence-to-sequence model) is added between the 2 sub-components.

Transformer's architecture was hand-crafted manually, Evolved Transformer (ET) was created to enhance Transformer. The idea behind ET is using neural architecture search (NAS) [14] to look for the most promising setup among different alternatives of neural networks. To modify Transformer model configuration toward a better one, ET uses an evolution-based algorithm with an innovative approach to expedite the process.

3. Proposed Method

Figure 2b describes our system architecture. The system works as follows, first, output from and ASR module (lowercase without punctuation) is fed to the Overlapped-Chunk Split module to produce overlapped segments. Second, the Capitalization and Punctuation Model takes the split segments and processes them in parallel to output a list of outputs. Finally, the outputs are merged back to form a final sentence using the Overlapped-Chunk Merging module. Details of each modules are described in the following sections.

3.1. Capitalization and Punctuation Model

This section describes the architecture and hyperparameters of our models. To be certain that our method of overlapping segments are efficient regardless of models, we preformed the experiments on sequence-to-sequence LSTM model and Evolved Transformer framework one by one. Our models are implemented based on Tensor2Tensor[15] and OpenNMT[10] framework. Concatenating overlapped chunks is developed as a separated module and used only after the inferring process.

To replicate the same condition, both the models have 6 hidden layers, word embedding size of 256, batch size of 4096 and trained for 200 epochs; the number of head in transformer model is 8. Their jobs is to convert from a sequence of lowercase text without punctuation to another sequence of capitalized text with punctuation. With 500 MB of text data for training, each model took 20 hours to train on an NVIDIA 2080Ti GPU.

3.2. Algorithm for Overlapped-Chunk Split and Merging

From preliminary experiments, we observed that the model often makes mistakes when processing words near the chunk boundary. We hypothesize that there is not enough context information around the area, leading to the poor performance of the model. To mitigate the problem, we proposed a method to split long input sentences into chunks with a chunk size of k words and a sliding window of $k/2$ words so that 2 consecutive chunks are overlapped. Later, the output of the model are merged in the way that we only keep predictions of the model where there is enough context information (an example is illustrated in Figure 2b).

While splitting input sentences into overlapped chunks is straight-forward as we only need to decide the chunk and overlapped size, merging the overlapped results is more difficult. Since the output of the overlapped region between 2 consecutive chunks can be different, we need to decide which words to keep and which word to remove to form a complete sentence. According to the hypothesis above, we defined a parameter called `min_words_cut` that indicates the number of words at the end the first chunk to be removed and also the number of words to be kept at the end of overlapped words in the second chunk. It ranges from 0 to the overlap size. With the value of 0, the whole overlapped words in the first chunk are kept while the overlapped words in the second chunk are

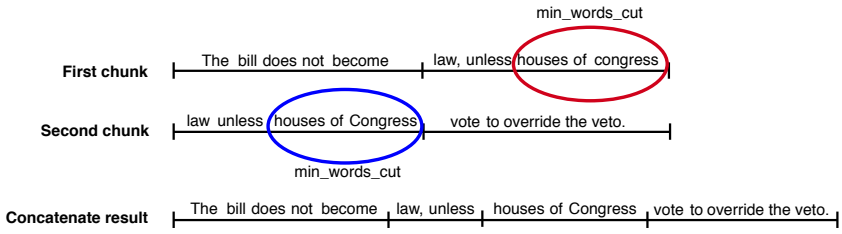


Figure 3: *Overlapped Chunk Concatenation*

Original data:
The bill does not become law, unless houses of Congress vote to override the veto.

Input data:

the	bill	does	not	become	**law**	**unless**	**houses**	**of**	**congress**
law	**unless**	**houses**	**of**	**congress**	vote	to	override	the	veto.

Plain text output:

The	bill	does	not	become	**law,**	**unless**	**houses**	**of**	**Congress**
law,	**unless**	**houses**	**of**	**Congress**	vote	to	override	the	veto.

Encoded output:

U$	L$	L$	L$	L$	L,	L$	L$	L$	U$
L,	L$	L$	L$	U$	L$	L$	L$	L$	L.

Figure 4: *Data samples with chunk size of 10*

removed (illustrated in Figure 3). The same principle is applied when `min_words_cut` equals to the overlapped size.

3.3. Data Preparation

To simulate the ASR output, we preprocess the dataset as followed. First, the characters are cleaned up: only the alphabet characters and three punctuation (comma, full stop and question mark) are kept. Then, we make sure that the punctuation belongs to the previous word, for instance, we use "laptop, mobile" not "laptop , mobile". Finally, we split data into chunks according to the split algorithm described in the above section. An example is shown in Figure 4.

We prepared 2 formats of training data: plain text and encoded text [9]. Both formats takes the lowercase text without punctuation as input. The plain text model, as the name suggest, provides output as plain text with punctuation and capitalization. The encoded text model, on the other hand, gives the result in an encoded format that contains only 6 classes as showed in Table 1. It is obvious that the encoded format will help the model to train and infer faster than the plain text since its vocabulary size is fixed and very limited. However, due to the limited vocabulary size, the decoder of the end-to-end model does not have much information of the words and the context information. We are interested to see how this method affect the quality in comparision with the plain text model.

4. Experiments and Results

4.1. Corpus Description

To train and evaluate the proposed method, we use the British National Corpus (BNC) [16] that contains 100 million words in both written and spoken language from a wide range of sources. It is designed to represent a large cross-section of British English from late 20th century. We use the XML edition which contains 4049 files with the size of 515 MB in total. The library NLTK [17] is used to extract 6M sentences from BNC dataset. For the test set, we use 67 thousand sentences. The number of label instances for each of the punctuation and capitalization classes available in our training and testing data set are displayed in Table 1.

Table 1: *BNC dataset detail. "U" and "L" respectively denote uppercase and lowercase word (either first or all character); ".", "," and "?" denotes full stop, comma, and question mark. The dollar sign ("$") indicates there are no punctuation coming after the word.*

Class	Training	Testing
U	13 M	146 K
L	81 M	1 M
.	4.6 M	54 K
,	4.9 M	57 K
?	380 K	5 K
$	87 M	1 M

4.2. Evaluation metric

The models (described in section 3.1) are evaluated using precision, recall, and F_1 scores. For ease of representation, we converted output words and punctuation to the 6-class encoded format as illustrated in Table 1. The evaluation results indicate how well the method can predict truecase of words and punctuation restoration. Since prediction of lowercase and blank space are good in every models, we ignore them in compare table.

4.3. Evaluation of chunk-merging

Table 2: *Comparison Seq2seq LSTM with and without using Chunk Merging for plain text format*

Model	Class	Precision	Recall	F1-score
Chunk Merging Seq2seq LSTM	U	0.74	0.53	**0.62**
	.	0.43	0.41	**0.42**
	,	0.10	0.87	**0.19**
	?	0.49	0.22	**0.30**
Non-Chunk Merging Seq2seq LSTM	U	0.70	0.53	0.61
	.	0.40	0.41	0.41
	,	0.10	0.85	0.18
	?	0.45	0.20	0.28

Table 2 shows the result of the seq2seq LSTM model with and without chunk-merging. As we can see, with the help of

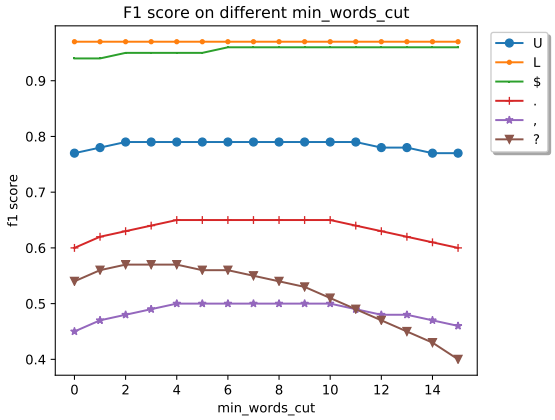


Figure 5: *F1-score on different min_word_cut. It peak in the middle range of overlap size (4-10). Predicting uppercase and lowercase are stable and independent from min_word_cut, question mark is quite sensitive with this hyper-parameter.*

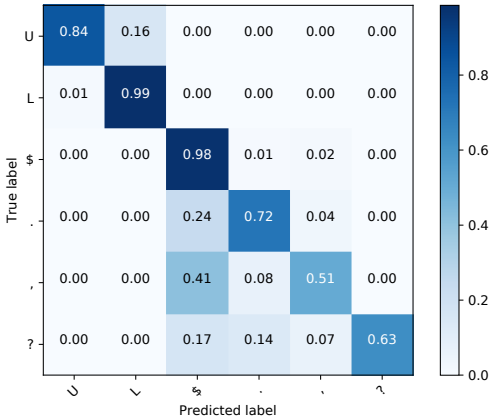


Figure 6: *Confusion matrix of Evolved Transformer model with plain text and overlapping format*

Table 3: *Comparison Evolved Transformer with and without using Chunk Merging for plain text format*

Model	Class	Precision	Recall	F1-score
Chunk Merging Evolved Transformer	U	0.90	0.84	**0.87**
	.	0.74	0.72	**0.73**
	,	0.61	0.51	**0.56**
	?	0.82	0.63	**0.71**
Non-Chunk Merging Evolved Transformer	U	0.84	0.79	0.81
	.	0.56	0.66	0.61
	,	0.40	0.42	0.41
	?	0.70	0.46	0.56

chunk merging, F_1 score on all classes are improved consistently by 1%. The result indicates that the overlapped words give the model more information to make better prediction, and that our chunk-merging method can select good portion of the overlap area.

The chunk-merging method even shows superior performance over non-chunk-merging when it is used with Evolved Transformer models. Results on Table 3 shows that the prediction accuracy of the question mark raises from 56% to 71%, this is a margin of 15% improvement and the minimum improvement of the system is 6% for the uppercase class. Figure 6 displays the confusion matrix of the model. The matrix shows that the comma is the most difficult class to predict and it is often mis-predicted as blank characters. In addition, the matrix also indicates that the model always predict a word (either lowercase or uppercase) when the input is word.

The results prove our hypothesis that there is not enough context for model to predict efficiently at the beginning and the end of each sample and that drawback can be overcome by adding more context with chunk overlapping and chunk-merging method.

4.4. Evaluation on plain-text model and encoded-text model

We further compare the result on models using plain text and encoded text. The ones with plain text outperform the ones with encoded text, however the model using encoded text has smaller model size and is faster for inference. The details are in Table 4

Table 4: *Comparison of results encoded text and plain text using Evolved Transformer*

Model	Class	Precision	Recall	F1-score
Encoded Text Chunk Merging Evolved Transformer	U	0.87	0.80	0.84
	.	0.68	0.66	0.67
	,	0.50	0.40	0.44
	?	0.76	0.55	0.63
Plain Text Chunk Merging Evolved Transformer	U	0.90	0.84	**0.87**
	.	0.74	0.72	**0.73**
	,	0.61	0.51	**0.56**
	?	0.82	0.63	**0.71**

To explore the impact of `min_words_cut` value to the quality of the result, we performed the experiment on sequence-to-sequence LSTM model with the overlapping of 15 words and `min_words_cut` ranges from 0 to 15. The outcome shown in Figure 5 indicates that f1-scores peak in the middle range of chunk size (4-10). It demonstrate that predictions of uppercase and lowercase are stable and independent from `min_words_cut`.

As processing chunks is paralleled and the concatenation algorithm has $\mathcal{O}(n)$, this approach is fast and proved to be superior to conventional methods.

5. Conclusion

In this research, we have proposed an end-to-end model that restores both punctuation and capitalization in one go. With chunk-split-merging, the method can splits and processes sentences in parallel and merges outputs to form the final sentence output. Experiments shows that the approach outperform existing methods that do not utilize chunk-merging by a significant margin, especially when combining with Evolved Transformer. In the future, we will integrate this solution with ASR model to form an end-to-end model that can transform speech to a well format text document.

6. References

[1] E. Cho, J. Niehues, and A. Waibel, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system," in *International Workshop on Spoken Language Translation (IWSLT) 2012*, 2012.

[2] M. Tkachenko and A. Simanovsky, "Named entity recognition: Exploring features." in *Proceeding of KONVENS*, 2012, pp. 118–127.

[3] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in *Proceedings of CICLing*, 2018, pp. 2145–2158.

[4] P. elasko, P. Szymaski, J. Mizgajski, A. Szymczak, Y. Carmiel, and N. Dehak, "Punctuation prediction model for conversational speech," *Interspeech 2018*, Sep 2018. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1096

[5] O. Tilk and T. Alumäe, "Lstm for punctuation restoration in speech transcripts," in *Sixteenth annual conference of the international speech communication association*, 2015.

[6] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the 2010 conference on empirical methods in natural language processing*, 2010, pp. 177–186.

[7] N. Ueffing, M. Bisani, and P. Vozila, "Improved models for automatic punctuation prediction for spoken and written text." in *Interspeech*, 2013, pp. 3097–3101.

[8] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[9] E. Cho, J. Niehues, and A. Waibel, "Nmt-based segmentation and punctuation insertion for real-time spoken language translation." in *INTERSPEECH*, 2017, pp. 2645–2649.

[10] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "OpenNMT: Open-source toolkit for neural machine translation," in *Proc. ACL*, 2017. [Online]. Available: https://doi.org/10.18653/v1/P17-4012

[11] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014.

[12] 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.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[14] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[15] A. Vaswani, S. Bengio, E. Brevdo, F. Chollet, A. N. Gomez, S. Gouws, L. Jones, L. Kaiser, N. Kalchbrenner, N. Parmar, R. Sepassi, N. Shazeer, and J. Uszkoreit, "Tensor2tensor for neural machine translation," *CoRR*, vol. abs/1803.07416, 2018. [Online]. Available: http://arxiv.org/abs/1803.07416

[16] B. Consortium, *The British National Corpus, version 3 (BNC XML Edition)*. Bodleian Libraries, University of Oxford, 2007.

[17] E. Loper and S. Bird, "Nltk: the natural language toolkit," *arXiv preprint cs/0205028*, 2002.

A HIGH QUALITY AND PHONETIC BALANCED SPEECH CORPUS FOR VIETNAMESE

Pham Ngoc Phuong[1], Quoc Truong Do[2], Luong Chi Mai[3]

[1]Information Technology Center, Thai Nguyen University; [2]Nara Institute of Science and Technology (NAIST);
[3]Institute of Information Technology, Vietnamese Academy of Science and Technology
[1]phuongpn@tnu.edu.vn; [2]do.truong.dj3@is.naist.jp; [3]lcmai@ioit.ac.vn

ABSTRACT

This paper presents a high quality Vietnamese speech corpus that can be used for analyzing Vietnamese speech characteristic as well as building speech synthesis models. The corpus consists of 5400 clean-speech utterances spoken by 12 speakers including 6 males and 6 females. The corpus is designed with phonetic balanced in mind so that it can be used for speech synthesis, especially, speech adaptation approaches. Specifically, all speakers utter a common dataset contains 250 phonetic balanced sentences. To increase the variety of speech context, each speaker also utters another 200 non-shared, phonetic-balanced sentences. The speakers are selected to cover a wide range of age and come from different regions of the North of Vietnam. The audios are recorded in a soundproof studio room, they are sampling at 48 kHz, 16 bits PCM, mono channel.

Index Terms: Speech database**,** Speech corpus, Vietnamese speech corpus

1. INTRODUCTION

In speech related research, especially, in speech analysis and speech synthesis studies, it is crucial to have high quality speech corpora. In some popular languages such as English and Japanese, there have been many intensive researchs on designing databases [1,2]. In Vietnam, the number of speech data research is alo increasing. There are many speech corpora such as VOV (Radio broadcast resources) [3], MICA VNSpeechCorpus [4], AIlab VIVOS [5], VAIS-1000 [6]. However, those corpora are either small or do not have a high quality sound. In particular, the VAIS-1000 corpus is designed from only a speaker with local accent from one particular region; The VIVOS corpus does not have high quality speech and it is designed specifictly for speech recognition tasks. The audio from VOV corpus are selected from media sources and are also only suitable for speech recognition tasks.

On the other hand, while research on speech synthesis adaptation which we can generate a model for a specific speaker with a very limited amount of speech samples is an active research field on popular languages [7], it is difficult to conduct such a research on Vietnamese due to a very high requirement on the data design. First, audio has to be recorded in a clean, soundproof recoding room to ensure the high quality speech. Second, the speakers have to be selected to cover wide range of ages as well as living areas. The corpus proposed in MICA VNSpeechCorpus [4] is well designed and contains good quality speech. However, although the total size of the corpus is big, the amount of short sentences that are suitable for speech synthesis adaptation is rather small.

In this paper, we present a high quality and large scale Vietnamese speech corpus. We design the corpus with a strategy that maximizes the coverage of monophone and biphone. Speakers are carefully selected with a wide range of age and living region. In the following section, we first describe Vietnamese phonetic structure (Section 2), it provides essential information to design and select the recording transcription described in Section 3. Finally, we provide details of the corpus along with analyses in Section 4.

2. BASIC PHONETIC STRUCTURE OF VIETNAMESE

Vietnamese language is a complex language compared with other languages because it is a monosyllable language with tones, every syllable always carries a certain tone [8,9]:

TONE			
Initial	FINAL		
	Onset	Nucleus	Coda

Table 1. Structure of Vietnamese syllables

There are 22 initials in Vietnamese, include: /b, m, f, v, t, t', d, n, z, ʐ, s, ʂ, c, ʈ, ɲ, l, k, χ, ŋ, ɣ, h, ʔ/. Onset /w/ has a function of lowering the tone of the syllables. The number of main finals consists of 16 phonemes, including 13 vowels and 3 dipthongs. Specifically, /i, e, ɛ, ɤ, ɤ̆, a, ɯ, ă, u, o, ɔ, ɔ̆, ɛ̆/ and 3 dipthongs / ie, ɯɤ, uo/. In addition to the final /zero/, there are 8 positive finals including 6 consonants /m, n, ŋ, p, t, k/ and 2 semi-vowel /-w, -j/.

There are 6 tones in Vietnamese. Five tones are represented by different diacritical marks such as l*ow*- falling tone, high-broken tone, low-rising tone, high-rising tone, low-broken tone. The tone called mid tone is not represented by a mark. Tones are differentiated in the following Table 2 [10,9]:

Contour / Pitch	Flat	Unflat	
		Broken	Unbroken
High	No mark	High-broken	High-rising
Low	Low-falling	Low-rising	Low-broken

Table 2. Structure of Vietnamese tones

The total number of unique syllables in Vietnamese is 19000 but there are only 6500 syllables used in practice [8,9].

3. DESIGN

In this section, we describe our recording transcription design strategy. To have a wide variety of context, we select the recording sentences from electronic newspapers. However, since the data from newspapers is typically noisy, we need to put it through a chain of data processing phases as illustrated in Figure 1. Then, we select the smallest subset of data that maintain the phonetic balance criterion.

3.1. Text data preprocessing

The text recording needs to be designed to meet the criteria that it is not too large but it must ensure the phonetic balance. We first collect a large amount of text from electronic newspapers. And then, process it in a chain of processing phases.

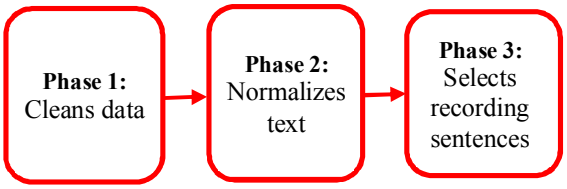


Figure 1. The process of refining and processing the recording texts

Phase 1: The data downloaded from newspapers has the main content stored in the <content> tags, other information outside the tag is metadata. To reduce noises, we only select texts from the <content> tag. Next we cut them into small sentences based on ending punctuations such as ".", "?", "!". To further reduce noises, we remove all lines contain post time, shortcuts, address, as well as arbitrary strings (asterisks, special characters, punctuation marks, author names, annotations, quoted source names etc).

Phase 2: In this phase, the main task is to normalize the text (includes many non standard words) according to the standards words in Vietnamese [11]. Non standard words include digit sequences; numbers; abbreviations; units of measurement; roman numerals; foreign proper names and place names... We analyzed text and used the technique to transforming (or expanding) a sequence of words into a common orthographic transcription. The process is done by 2 steps:

Step 1. To reduce pronunciation ambiguity, all numbers, date, time and measure units are spelled out with the following rules:

- Number format:

Numbers are transcribed in code by assigning them to arrays and transcribing them into corresponding strings (e.g. 1235 → một nghìn hai trăm ba mươi năm). Then exceptions are replaced with standard words (e.g. không mươi→ lẻ, mươi năm →mươi lăm, mươi một → mươi mốt).

- Time format:

Format dd/mm/yyyy is automatically transcribed into day...month...year .

Format (dd/mm, dd-mm-yyyy and dd-mm) with the word 'day' standing in front is transcribed as "day", "month".

Format hh:mm:ss is understood as hour, minute, second

Format hh:mm with "at" standing in front is transcribed into "hour", "minute".

- Units of measurement:

Separate alphanumeric characters with spaces (e.g. 10Kg → 10 k, 10m → 10 m, 11hz → 11 hz, 8/10 → 8 / 10, 90% → 90 %).

Then, replace words with transcribing digits for signs or measure units (e.g. 10 kg → ten kilograms, 10 meters → ten meters, 11 hz → eleventh hertz, 8 / 10 → eight per ten, 90% → ninety percent).

Step 2. Transcribe abbreviated acronyms or proper names with self-defined dictionaries (e.g. TP → city, HCM → Ho Chi Minh City, VND → Vietnam dong, Paris → Pa ri, Samsung → Sam Sung).

After normalizing the text, we split them into small sentences and only keep ones contain minimum 40 and maximum 90 syllables. This is an appropriate length for speech recording.

Phase 3: The final step is to select a good amount of sentences for audio recording. The recording sentences should maintain the phonetic balance property and be small to reduce recording cost. We adopt text selection based on greedy search to find the optimal sentences [12]. This step is repeated until a certain amount of sentences are selected.

3.2. Recording

To help speeding up the recording, as well as make it easier and less prone to human error. We designed a recording application as shown in Figure 2. The speaker can listen to their recorded audio and can also see the audio signal to ensure that there is 1 second of silence at the beginning and ending of utterances and there is no audio clipping occurred. During a recording session, if there is any sentence doesn't meet the requirement, the speaker will only need to record the sentence again. The quality assurance process is managed by an administrator using the same application with speakers.

Figure 2. Web-based recording application

The audio is recorded with a high-quality TakStar PC-K600 microphone using a Windows 10 computer running a Firefox browser. The audio sampling rate is 48Hhz, 16-bits PCM, and mono channel.

4. EXPERIMENT

The following Table 3 summarizes the data collected in the above text data preprocessing section:

No	Name	Size	Unit	Note
1	Original data	1.978/9,6	File/GB	Data in file format .txt
2	Phase 1	1.978/8,1	File/GB	Refine content, raw processing
3	Phase 2	4.036.312	Sentence	Text Normalization
4	Phase 3	250/2.400	share/non-shared sentences	Select sentences for recording: each person records 250 share common sentence and 200 none-share sentences.

Table 3. Steps of text data preprocessing

4.1 Corpus detail

To ensure the wide variety of speech, we selected 12 speakers from 5 provinces of North Vietnam including 6 males and 6 females aged 22 to 35. Our target number recording sentences is 250 that has to maintain the phonetic-balanced property. Our experiment on data selection algorithm described in the section above shown that only 250 utterances are needed to cover all monophone and 99% of bi-phone. This is a good sign that we do not have to select many sentences to meet the requirement. We use this 250 utterances for all speakers.

However, to increase the variety, we also want each speaker utter a separate set of text. Therefore, we run the data selection algorithm again to select other sets for each speaker. Each run, we removed the selected sentence to make sure there are no duplicated sentences in the corpus. As the result, we have 200 x 12 phonetic-balanced sentences. Note that male and females speakers share the same recording utterances. As the result, we have recorded 2,400 speech samples. The results as shown in Table 4.

Data set	Number of letters	Number of syllables	Number of syllables per sentence	Unique syllables
250 sentence set	14.954	3.268	59,8160	1.205
2400 sentence set	120.6320	26.308	50,2633	2.758

Table 4. Statistics of text sentences for recording

4.2 Data analysis

4.2.1 Phonemes statistic

To evaluate data corpus, we use several modules to count two text data sets based on occurrence frequency and deference of phonemes, syllables and words. The results as shown in Table 5.

No	*Set of 250 share common sentences*				*Set of 2,400 none-share sentences*			
	Bi phone	*Frequency of Occurence*	*Mono phone*	*Frequency of Occurence*	*Bi phone*	*Frequency of Occurence*	*Mono phone*	*Frequency of Occurence*
1	ea-ngz	89	ngz	526	a-iz	809	a	4482
2	a-iz	84	a	510	oo-ngz	644	ngz	4235
3	oo-ngz	78	iz	347	ea-ngz	636	iz	3133
4	l-a	76	nz	340	aa-nz	507	nz	2871
5	oa-ngz	59	k	296	ie-nz	504	k	2432
6	aa-nz	58	i	286	u-ngz	484	oo	2355
7	ngz-k	56	oo	265	l-a	482	i	2286
8	u-ngz	54	dd	236	a-nz	472	dd	1981
9	k-o	53	tr	232	aw-iz	469	tr	1863
10	aw-iz	52	aa	227	oo-iz	464	aa	1824
11	a-nz	51	wa	218	i-ngz	447	kc	1724
12	ie-uz	51	kc	216	w-a	436	aw	1658
13	wa-ngz	50	ie	211	oa-ngz	419	ie	1647
14	aa-tc	49	aw	202	k-o	411	wa	1598
15	ow-iz	49	ee	190	ie-uz	401	uz	1579
16	ie-nz	48	uz	188	ngz-k	400	o	1464
17	uw-ngz	48	o	183	k-uo	389	t	1443
18	w-a	45	uw	182	ow-iz	389	mz	1442
19	wa-kc	45	th	180	b-a	386	ee	1410
20	oo-iz	44	tc	175	uw-ngz	379	m	1386

Table 5. Statistics of 20 most popular phonemes in 2 data set (without sil)

4.2.2 Sound quality analysis

The sound quality was analyzed by Praat v6.0 software to evaluate the characteristics of sound waves, spectra, pitch and sound intensity [13]. The following example in Figure 3 will analyze the waveforms and spectrograms of a female voice extracted from the utterance "đây là một chuyên mục tôi cảm thấy tâm đắc và uy tín" (in English “this is the prestige category which I feel favorite”).

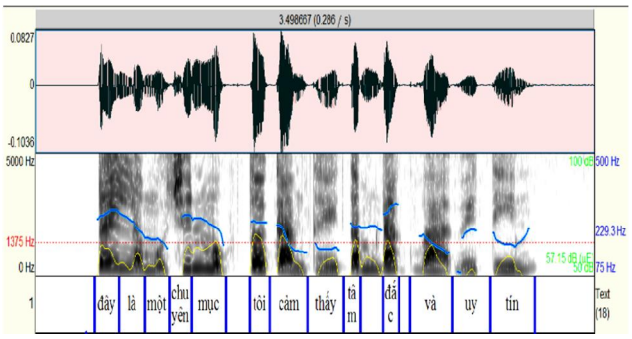


Figure 3. Waveform and spectrogram of the female voice

Data evaluation was done through the assessment of the recording environment, the noise ratio [14]. Through analysis and evaluation of all data, the recording was evaluated to be of good quality with clear sound and little noise.

4.2.3 Duration analysis

In this experiment, we are interested in the difference between genders and ages in term of duration of words. To obtain word duration, we build an automatic speech recognition (ASR) using Kaldi toolkit [15]. The training for

the ASR system is the same data used for the decoding process so that we can have accurate audio alignment results. State duration of each HMM are modeled by a multivariate Gaussian estimated from histograms of state durations which were obtained by the Viterbi segmentation of training data [16,17].

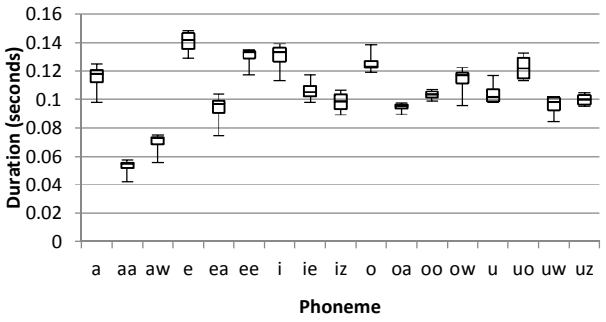


Figure 4. Duration distributions of vowels spoken by female voices at the same age

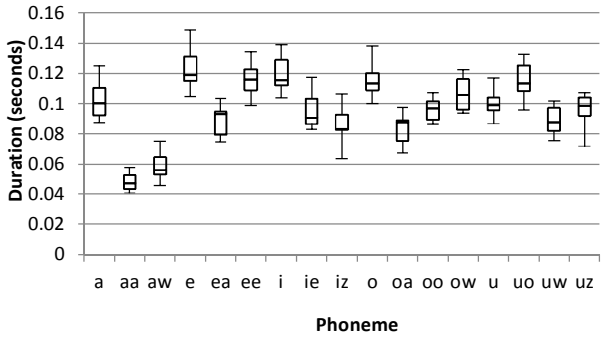


Figure 5. Duration distribution of vowels spoken by people with wide range of age and different gender

Figure 4 shows vowel duration distributions of 4 female speakers at the same age. As we can see, the range of all distributions are quite small, indicating that females at the same age tend to have similar reading speed. On the other hand, Figure 5 shows the duration distributions of speakers with different genders and wide range of age. We can clearly see that the distribution is larger than it is in Figure 4.

As one of the purpose of the corpus is to build speech synthesis adaptation systems. The result indicates an important clue that we should not use as many data as possible but instead only use data that have similar characteristic such as gender or age to achieve optimize results in term of duration adaptation.

5. CONCLUSION

In this paper, we have described a high quality speech corpus for Vietnamese that is suitable for data analysis and constructing speech synthesis systems. The work result is a high-quality data set that contains 5,400 utterances with the accompanied text which were recorded by variety gender and age speakers. This is the minimum data set that meets our target which guarantee that the amount of short sentences with phonetic balanced is suitable for speech synthesis adaptation. Future work will focus on expanding the data size in both term of speakers and accent, as well as, utilizing the corpus to construct Vietnamese speech synthesis adaptation systems.

6. REFERENCES

[1] J. Harrington, “Phonetic analysis of speech corpora”, John Wiley & Sons, 2010, pp. 4-5.

[2] S. Itahashi and K. Hasida, "Japanese Effort Toward Sharing Text and Speech Corpora," in *Proceeding of IJCNLP*, Hyderabad, India, 2008.

[3] L. C. Mai and D. N. Duc, "Design of Vietnamese Speech Corpus and Current Status," in *Proceeding of ISCSLP*, Kent Rigde, Singapore, 2006.

[4] L. V. Bac, T. D. Dat, E. Castelli and L. Besacier, "Spoken and written language resources for Vietnamese," in *Proceedings of LREC*, Lisbon, Portugal, 2004.

[5] L. T. Hieu and V. H. Quan, "A non-expert Kaldi recipe for Vietnamese Speech Recognition System," in *Proceedings WLSI-3 & OIAF4HLT-2*, Osaka, Japan, 2016.

[6] Q. T. Do and L. C. Mai, "VAIS-1000: a Vietnamese speech synthesis corpus" IEEE Dataport, 2017.

[7] J. Yamagishi, T. Kobayashi, Member, Y. Nakano and O. a. J. I. Katsumi, "Analysis of Speaker Adaptation Algorithms for HMM-Based Speech Synthesis and a Constrained SMAPLR Adaptation Algorithm," *IEEE Transactions on Audio, Speech, and Language Processing,* vol. 17, no. 1, p. 66, January, 2009.

[8] V. T. Thang, L. C. Mai and S. Nakamura, "An HMM-based Vietnamese Speech Synthesis System," in *Proceedings of Oriental-COCOSDA*, China, 10-12 August, 2009.

[9] M. N. Chừ, “Cơ sở ngôn ngữ học và tiếng Việt”, VietNam Education Publishing House, 1997.

[10] Đ. T. Thuật, "Ngữ âm tiếng Việt," VietNam National University Press, HaNoi, 2003.

[11] N. T. T. Trang, P. T. Thanh and T. D. Dat, "A method for Vietnamese Text Normalization to improve the quality of speech synthesis," in *Proceedings of SoICT*, Hanoi, Viet Nam, 2010.

[12] J. Zhang and S.Nakamura, "An efficient algorithm to search for a minimum sentence set for Collecting Speech Database," in *Proceeding of ICPhS*, Barcelona, 3-9 August 2003.

[13] P. Boersma and D. Weenink, "Praat: doing phonetics by computer, [Computer program]. Version 6.0. 19, 2016

[14] K.Genuit, "How to evaluate noise impact," in *Proceedings euronoise Naples,* Italy paper ID 347(2003*),* pp. 1-4.

[15] I. Mporas, G. Todor and N. Fakotakis, "A hybrid architecture for automatic segmentation of speech waveforms," in *In Proceeding of ICASSP*, Las Vegas,USA, 2008.

[16] K. T. Takayoshi Yoshimura, T. Masuko and T. K. a. T. Kitamura, "Duration modeling for hmm-based speech synthesis," in *Proceeding of ICSLP*, Australia, 1998.

[17] P. T. Son, V. T. Tat, D. T. Cuong and L. C. Mai, "A study in Vietnamese statistical parametric speech synthesis base on HMM," *International Journal of Advances in Computer Science and Technology,* Vols. Volume 2, No.1, pp. 1-6, January 2013.

BỘ VĂN HÓA, THỂ THAO VÀ DU LỊCH
CỤC BẢN QUYỀN TÁC GIẢ

GIẤY CHỨNG NHẬN
ĐĂNG KÝ QUYỀN TÁC GIẢ

CỤC BẢN QUYỀN TÁC GIẢ CHỨNG NHẬN

Tác phẩm: *Phần mềm Chuyển đổi văn bản thành giọng nói ADAPT - TTS*

Loại hình: *Chương trình máy tính (Không bao gồm dữ liệu)*

Tác giả: *Phạm Ngọc Phương*
Tổ 10, P. Gia Sàng, TP. Thái Nguyên, T. Thái Nguyên

Quốc tịch: *Việt Nam*
Số CCCD: *019084001343 10/04/2021*

Đã đăng ký quyền tác giả tại Cục Bản quyền Tác giả

Hà Nội, ngày 26 tháng 09 năm 2022
KT. CỤC TRƯỞNG
PHÓ CỤC TRƯỞNG

CỘNG HÒA X.H.C.N VIỆT NAM
CỤC BẢN QUYỀN TÁC GIẢ
BỘ VĂN HÓA THỂ THAO VÀ DU LỊCH

Số: 7590/2022/QTG
Cấp cho Chủ sở hữu

Phạm Thị Kim Oanh