UNIVERSITY OF SCIENCE

ADVANCED PROGRAM IN COMPUTER SCIENCE

ĐỖ TRÍ NHÂN - NGUYỄN MINH TRÍ

DEEPSPEECHVC: VOICE CLONING FRAMEWORK WITH SPEECH SYNTHESIS AND VOICE CONVERSION

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

HO CHI MINH CITY, 2021

UNIVERSITY OF SCIENCE

ADVANCED PROGRAM IN COMPUTER SCIENCE

ĐỖ TRÍ NHÂN - 1751087

NGUYỄN MINH TRÍ - 1751109

DEEPSPEECHVC: VOICE CLONING FRAMEWORK WITH SPEECH SYNTHESIS AND VOICE CONVERSION

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

THESIS ADVISOR:

ASSOC.PROF. VŨ HẢI QUÂN

MSC. CAO XUÂN NAM

HO CHI MINH CITY,2021

APCS DEEPSPEECHVC: VOICE CLONING FRAMEWORK 2021

UNIVERSITY OF SCIENCE

ADVANCED PROGRAM IN COMPUTER SCIENCE

COMMENTS OF THESIS'S ADVISOR

(Research)

Thesis title:

DEEPSPEECHVC: VOICE CLONING FRAMEWORK WITH
SPEECH SYNTHESIS AND VOICE CONVERSION

Students: **Đỗ Trí Nhân** (1751087) – **Nguyễn Minh Trí** (1751109)

Advisor: **Assoc. Prof. Vũ Hải Quân - MSc.Cao Xuân Nam**

1. Research Topics and Ideas:

The main objective of this thesis is to develop an effective solution to reconstruct the voice of a person who died or became suddenly mute after accident with only 3-5 seconds of voice. Then, the students applied this solution to build a pipeline framework for whole Vietnamese voice cloning system based on speech synthesis and voice conversion. In this thesis, the students were extremely active in suggesting research topic, conducting experiments on the state-of-the-art models, and coming up with practical solution.

2. Research Methods:

The students have taken the proper approach to the topic and used the appropriated research methods. They have studied very carefully and gained basic knowledge in this field. The students also spent a significant amount of time reading and experimenting with a huge number of scientific papers in order to get knowledge of both conventional and modern approaches.

3. Contributions:

With the voice conversion module, the students propose a new method - DeepSpeechVC. The students proved that this model helps to create cloning voices more efficiently and faster than the previous methods. This thesis can be considered as one of the first research on Voice Conversion for Vietnamese.

Moreover, in pipeline framework, speech synthesis models are experimented, evaluated and compared in detail, give an overview of end-to-end TTS models when applied to Vietnamese. The models have been compressed to be able to run on Android devices without internet and GPU. The voice output is natural and fluent, similar to the real voice used for training.

They also provide a library for text normalization – Vinorm and a library for Grapheme to Phoneme conversion - Viphoneme, which helps the speech synthesis process to converge faster.

4. **Report:**

The report is organized clearly, logically and the content is completely given on the research topic, knowledge foundation, and related methods. The students also give experimental and demo in depth.

5. **Presentation:**

The students give a concise presentation, concentrating on the key research content of the topic.

6. **Publications and/or realworld applications:**

The students have 02 published papers at NAFOSTED Conference and at MediaEval Workshop.

Rank: *Outstanding*

Ho Chi Minh city, August 29, 2021

Advisors

Vũ Hải Quân

Cao Xuân Nam

UNIVERSITY OF SCIENCE

ADVANCED PROGRAM IN COMPUTER SCIENCE

COMMENTS OF THESIS'S REVIEWER

(Research)

Thesis title: DEEPSPEECHVC: VOICE CLONING FRAMEWORK WITH SPEECH SYNTHESIS AND VOICE CONVERSION

Students: **Đỗ Trí Nhân** (1751087) – **Nguyễn Minh Trí** (1751109)

Advisor: **Assoc. Prof. Vũ Hải Quân**

MSc. Cao Xuân Nam

1. Research Topics and Ideas:

Voice cloning has meaningful applications in practice to create voice of people who are muted by accident or people who passed away. Voice cloning can be done by speech synthesis. However, traditional synthesis requires a large amount of data from a speaker to create his/her voice, while voice cloning assumes a very small piece of data is available, normally several seconds. With the development of deep learning, currently the most effective approach is end-to-end multi-speaker speech synthesis. This approach allows to create a voice with several seconds of reference speech data. However, it requires a large amount of training data from multiple speakers, which is not always satisfied in practice, especially for Vietnamese language as it is still a low-resource language at the moment.

This thesis proposes a voice cloning method for limited training data condition by first synthesizing temporary speech from text content, then using voice conversion to convert temporary speech to the target voice. Overall, the research topic of this thesis is important, and the approach is reasonable.

2. Research Methods:

This research is a quantitative research. Hypothesis was proved by objective evaluation on experimental results.

To synthesize a speech utterance from a text sentence and small reference speech, the author first convert text a temporary speech using speech synthesis and convert represent reference speech into speaker embedding vector. Then the temporary speech and the embedding are passed through a voice conversion module to yield output speech. To build this system, the authors had to investigate speech synthesis, speaker recognition and voice conversion.

The main contribution of this thesis is the improvement voice conversion method. The authors used AutoVC for voice conversion, which is considered as state of the art at the moment. However, AutoVC has a bottle problem that the AutoEncoder component, which is used to represent speech content, has to be adjusted manually in practical training. The authors proposed to use Deep Speech 2 to replace this component to better represent speech content and to be trained independently.

Experiment showed that this improvement provided better voice conversion performance in comparison with the original AutoVC method. Furthermore, the objective of voice cloning is achieved by providing relatively similar voice synthesis with the certain condition of available data.

3. Contributions:

This research suggested a new approach for voice cloning problem with limited speech data and applied into some public Vietnamese datasets. Through this research, the thesis additionally provided a survey and experimental results of various speech synthesis methods on Vietnamese language, which can be used as a reference for TTS.

The author also published the module of Normalization – Phonetization via Python package library for public use and deployed a demo on internet.

4. Report:

- The report was well-organized with 6 chapters: introduction, preliminaries, related work, proposed method, experiments and results, and conclusion.
- It further provided appendix for Vietnamese text normalization and related publication from the result of this thesis.

5. Presentation:

- Clear enough to understand the topic, method and results.
- All questions are answered.

6. Publications and/or realworld applications:

The authors have published one paper from the results of this thesis and another paper related to this thesis.

- D. T. Nhan, N. M. Tri and C. X. Nam, "Vietnamese Speech Synthesis with End-to-End Model and Text Normalization," 2020 7th NAFOSTED Conference on Information and Computer Science (NICS), 2020
- Tri-Nhan Do, Minh-Tri Nguyen, Hai-Dang Nguyen, Minh-Triet Tran and Xuan-Nam Cao: HCMUS at MediaEval 2020: Emotion Classification Using Wavenet Feature with SpecAugment and EfficientNet. Proc. of MediaEval 2020, 14-15 December 2020.

Rank: *Outstanding*

Ho Chi Minh city, August 29, 2021

Reviewer

Châu Thành Đức

ACKNOWLEDGEMENT

First and foremost, we would like to express our sincere gratitude to our advisor, Assoc. Prof. Vu Hai Quan. It was a blessing for us to have been a collaborator at AILAB since the early days of our sophomore year. We have the opportunity to approach and perform the most primitive problems in the field of speech processing. Lab meetings gave me an overview of the research and making applied AI products. That's a solid foundation, he has inspired me with passion to research this field of speech processing until doing this thesis and future research and work on speech. We promise to continue his great achievements in the field of speech research, explore new research directions and apply them to practice.

We would also like to thank MSc.Cao Xuan Nam, who spent the most time with us in this thesis. We really appreciate the continuous support of our study and research from him. Thanks to his patient, diligent guidance, we have acquired valuable knowledge in the fields of Phonology, Natural Language, Speech and Signal Processing. Had it not been for his constructive and sincere advice and encouragement, we would not have been able to complete this thesis.

We also would like to express our gratitude to all lecturers in the Faculty in Information and Technology, University of Science, VNU-HCM. Thanks to them, we were able to have a solid foundation in Computer Science, especially Artificial Intelligence, which helped us a lot in our in further research to complete our thesis.

In addition, this work is supported by AILAB for providing resources for experimentation and research, we are really thankful to our wonderful colleagues at AILAB and Robotics & IoT club, University of Science, including Huynh Hanh Thong, Pham Minh Nhut, Tran Minh Truong, Dang Hoai Thuong, and our colleagues at VinAI and VNG. They gave us valuable comments, discussions and always ready to help whenever we need.

Last but not least, we would like to express our deepest gratitude to our parents. Their moral support, caring and encouragement plays an important role in our success in education.

Authors
Đỗ Trí Nhân & Nguyễn Minh Trí

UNIVERSITY OF SCIENCE
ADVANCED PROGRAM IN COMPUTER SCIENCE

Thesis Proposal

Thesis title:

DEEPSPEECHVC: VOICE CLONING FRAMEWORK WITH SPEECH SYNTHESIS AND VOICE CONVERSION

Thesis advisor: Assoc. Prof. Vũ Hải Quân, MSc.Cao Xuân Nam
Students: Đỗ Trí Nhân (1751087) – Nguyễn Minh Trí (1751109)
Type of thesis: Research
Duration: From january 2021 to july 2021
Content of thesis:

1. Motivation:

With motivation of reconstructing voices for people who are mute after an accident or a person who has died and there is a small amount of data about their voices, this thesis aims to conduct experiments and apply new technologies, artificial neural networks to synthesize voices for Vietnamese. Different from speech synthesis models that require a one-speaker quality data set - traditional voice cloning, we suggest a new voice clone sytem, which is able to synthesize any person's voice with only a few sample audio input of that person's voice.

2. Problem Statement:

The voice cloning problem requires building a model with:

- Input: a text or sentence that needs to be converted (Z1) and a reference speaker audio (U2)
- Output: an audio that contains the content of the text (Z1) with the voice of the reference speaker (U2)

In other words, our task is to create an audio that speaks a given text in any reference voice.

3. Introduction:

One of the approaches to this problem is to build a unified speech synthesis model, as shown in Figure 1, which generates the resulted audio directly from the input text and reference audio without any intermediary step. This unified speech synthesis model is called MultiSpeaker Synthesiser.

In this approach, first the raw text will go through the Text Normalization and Phonetization module to be normalized and converted into phonetic representation features, in the other hand, the reference speaker audio will be passed through Speaker Embedding module to extract voice characteristic features.

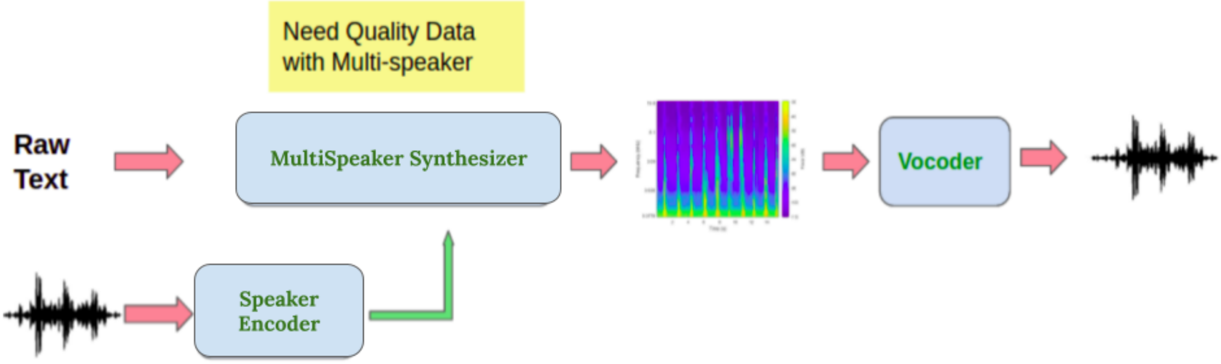


Figure 1: Tradional Voice Clone using MultiSpeaker Synthesis

The phonetic features and the characteristic features are passed to MultiSpeaker Synthesizer module to generate the Mel-Spectrogram, a kind of features generated from audio signal. Finally, the Vocoder will receive the generated Mel-Spectrogram to create the final audio result.

This approach is proved to be efficient, such as Baidu DeepVoice3 [3]. However,as we tried this method with Vietnamese dataset, the results are disappointing. We suspect that the total recoding hours, the number of speaker, the number of audio per speaker as well as the quality of Vietnamese dataset are not enough for this approach.

Therefore, in this thesis, we propose another approach that can be applied using smaller amount of dataset and can still get considerable result. We called this approach Voice Clone using SingleSpeaker Synthesis and Voice Conversion.

4. Method:

4.1. Model:

Voice Clone using SingleSpeaker Synthesis and Voice Conversion approach, as shown in Figure 2, is almost indentical to Traditional Voice Clone approach, the two diffirences are:

- The MultiSpeaker synthesizer in voice cloning is replaced by SingleSpeaker synthesizer, which is the synthesizer that is only able to generate audio in one speaker's voice.

- Voice Conversion module is added to convert the voice of the audio generated from SingleSpeaker synthesizer into any other voice.

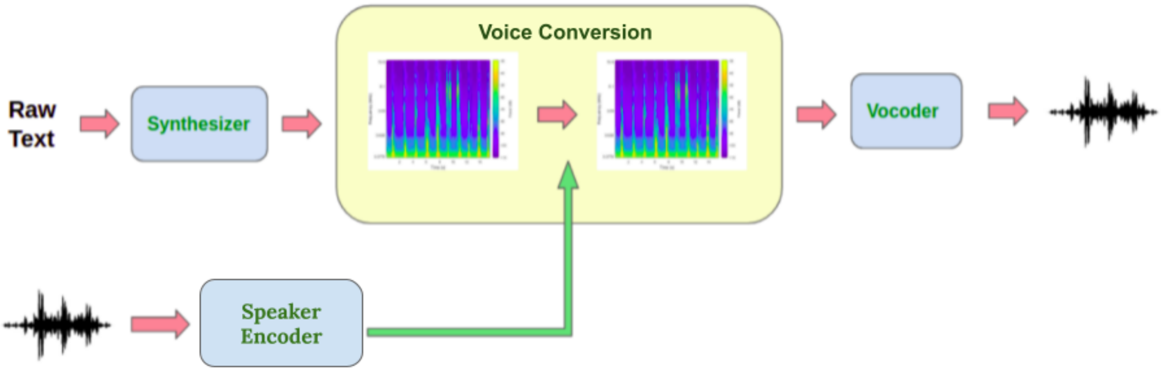


Figure 2: Voice Clone using SingleSpeaker Synthesis and Voice Conversion

With this approach, our mission only needs to build 5 separate module with data training requirements that are not too large at Conversion step:

- Text Normalization and Phonetization
- Synthesizer
- Vocoder
- Speaker Embedding
- Conversion Module

4.1.1. Text Normalization and Phonetization:

Each language has a different phonological and contex-tual characteristics, we have conducted experiments, statistics,and applied Vietnamese phonetics to improve speech synthesis systems. Our methods achieve the accuracy of 97% in text normalization tasks. We also provide a library for standardizing Vietnamese text called Vinorm[1] and a package that converts text into a phonetic format called Viphoneme[2], which is used as an input for end-to-end neural networks, making the synthesis process faster, more intelligent and natural than using character inputs. More infomations about our research about text normalization and phonetization in this paper [2].

4.1.2. Synthesizer:

Due to the development of Deep Neural Network learning method, many TTS systems moved to use end-to-end models and gain significantly improving results, such as Tacotron2 [6] and FastSpeech [5].

[1] Vinorm package link: https://pypi.org/project/vinorm

[2] Viphoneme package link: https://pypi.org/project/viphoneme

These systems do not use complex linguistic and acoustic features, they learn to produce audio directly from text, generate human- like speech using neural networks.

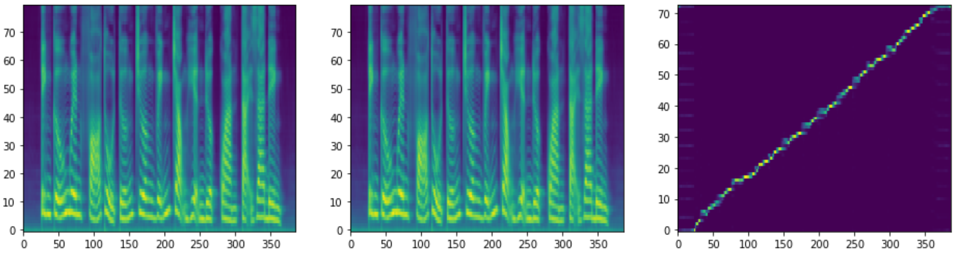


Figure 4: Tacotron 2 trainning alignment result

4.1.3. Vocoder:

After the Conversion module generates the mel-spectrograms result, we use a vocoder like WaveNet to convert the mel-spectrograms into audios. They are able to generate emotional, smooth and clean speech, work well on out-of-domain and complex words, learn pronunciations based on phrase semantics and are robust to spelling errors.

In this thesis, we will train the Tacotron2 and FastSpeech2 models for the Mel-generator module, and do transfer learning from the English pretrain on three vocoders: WaveGlow, Multiband-MelGan, and HifiGan.

Frontend	Mel-Generator	Vocoder
Viphoneme	FastSpeech2	MultibandMelGan
Viphoneme	Tacotron2	HifiGan
Viphoneme	Tacotron2	Waveglow
Viphoneme	GlowTTS	HifiGAN
Viphoneme	Tacotron2	Wavenet

Figure 5: Some pipelines of end2end speech syntheis for experiment

4.1.4. Speaker Embedding:

This module is used to extract speaker voice characteristics features to pass to the Voice Conversion module for trainning.

There are two ways for represents the speaker characteristics:

- Embed Speaker Identification: Domain Dependency using One-hot vector
- Embed Speaker Representation: derive a high-level representation of a voice summary vector of 256 values - characteristics of voice speaker

In the propose, we will do experiments with both representations. In the case of Embed Speaker Representation, we train the module according to paper [7].

4.1.5. Conversion Module:

This module is used to convert one speaker voice to another speaker voice. One of the most representative model of this type is the AutoVC model, as shown in Figure 5.

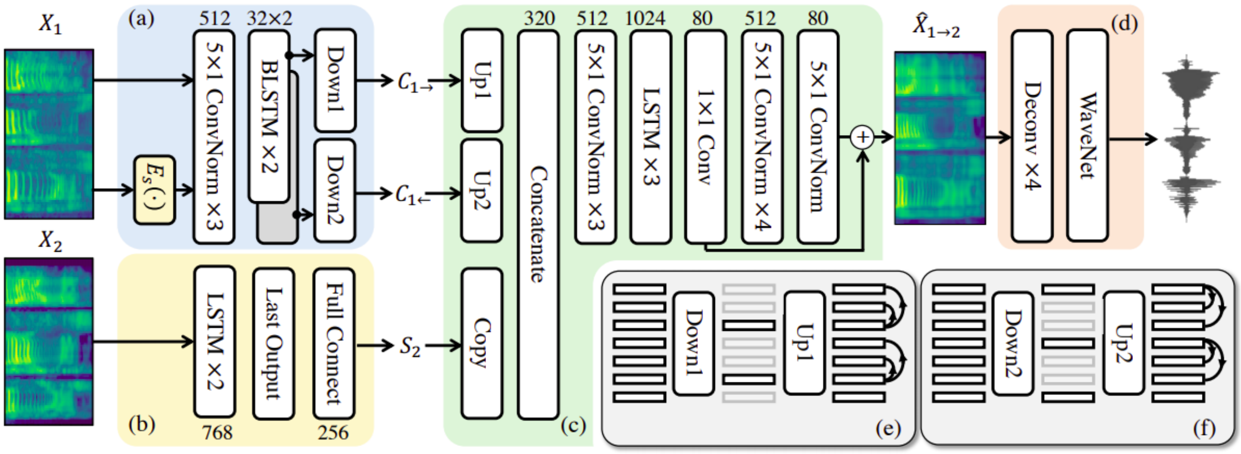


Figure 6: AutoVC architecture [4]

With the AutoVC approach, there are two ways that we can convert voice:

- One-hot Voice Conversion: U1 and U2 are both seen in the training set

- Zero-shot Voice Conversion: U1 or U2 is not included in the training set

with U1, U2 is the utterance of speaker

In our thesis, we focus on experimenting, analyzing and applying AutoVC model for Vietnamese dataset using zero-shot voice conversion so that the model can generate good result using unseen speaker audios.

4.2. Dataset:

In this thesis, we use 3 multispeaker datasets in Vietnamese:

- VIVOS: 15 hours, 46 speaker in total [3]

- VinBigdata ASR in VLSP 2020: 100 hours in total [4]

- Zalo AI Challenge: 400 speaker, each speaker speaker an avarage of 17 utterences [5]

Originally, the English dataset VCTK corpus has 44 hours of utterances from 109 speakers and the model requires at least 20 speakers. For the VIVOS dataset, number of difference speakers and number of utterances in each speaker set is small. The VinBigdata dataset is large but it is not classified into clusters of speakers.

The datasets with a single speaker of good quality are used for training two modules synthesizer and vocoder. We experiment on two sets of VLSP2019 and the TTS dataset of VinAI.

4.3. Experiments and Evaluation:

4.3.1. Data Preprocessing:

[3] VIVOS dataset link: https://ailab.hcmus.edu.vn/vivos

[4] VinBigdata VLSP dataset link: https://ailab.hcmus.edu.vn/vivos

[5] Zalo AI dataset link: https://challenge.zalo.ai/portal/news-summarization

Dataset	Module
ZaloAI	Training for Speaker Verification module
VinBigData ASR 100h	Testing for Speaker Verification modules
VLSP2019 TTS	Training for Speech Synthesis module
VinAI TTS	Training for Speech Synthesis module
VIVOS-AILAB	Training AutoVC and DeepSpeech-based Voice Conversion modules
VCTK	Training for DeepSpeech-based Voice Conversion in English voice conversion

List of datasets and the modules in which they are used for training and testing.

Since we don’t have ideal datasets for this problem, this thesis proposes some methods to clustering speakers in VinBigData dataset into clusters for training. There are many methods to clustering the speakers, we can use the spekaer verification model to create a specific d-vector for each speaker, then use K-mean to classify. Categorization can also be done based on open source code such as Kaldi M6, Pyanotate or x-vector methods. We will try each clustering method, compare them and then apply the best method that gets the best results.

4.3.2. Model Trainning:

The ZaloAI dataset will be used to train the Speaker Embedding module, both the Vivovs dataset and VinBigData will be used for Voice Conversion Module ’s trainning.

According to the original AutoVC paper, finding the right ”BottleNeck dimmension width” is important to obtain the best results, therefor we will conduct experiments relate to this parameters. At the same time, applying clustering methods to preprocess data is expected to improve the training process.

4.3.3. Evaluation Metrics:

To evaluate the results of Voice Conversion, we use two main metric:

- Mean opinion score (MOS): a measure method using real people, using the formula:

$$MOS = \frac{\Sigma_i^N Z_i}{N}$$

Where Z is rating of one person and N is the total number of rating people

- Mel cepstral distortion (MCD): a measure method which the result is obtained by compare two resulted audio mathematically[1], using the formula:

$$MCD(y, y') = \frac{10}{ln(10)}\sqrt{2\sum_{t=1}^{T}|y - y'|}$$

Where y and y' are the ground and the predicted mel-spectrogram respectively, T is the number of timesteps and t is the timestep slice.

5. Preliminary results:

We expect the Mean opinion score (MOS) to be as good as the result of our previous experiment on single speaker speech synthesis, which is around 3.97.

6. Ethic Problem:

AI Technology is growing rapidly, which has also raised many concerns about the danger of the development itself. In the past, there were also many synthesized voice systems doing voice cloning, but the most notable here is the ”one-minute” number, collecting a person's voice in a minute is a lot easier than collecting an hour's data set audio. This raises important questions if the system can become a tool for bad guys, can be used for tricking the verified identity of a software, and bring more unhappiness than happiness. Despite of that problem, we believe this technology can be used for creativity and entertainment, and make human's life more colorful. We can publicize the technology so that everyone will soon be aware that such technology exists. By that way, the damage will be a lesson, we agree with such a solution. In our opinion, this technology should be public and should be developed more to make it a safe tool for everyone to use.

7. References:

[1] Albert Haque, Michelle Guo, and Prateek Verma. *Conditional End-to-End Audio Transforms*. 2018. arXiv: 1804.00047 [cs.SD].

[2] D. T. Nhan, N. M. Tri, and C. X. Nam. “Vietnamese Speech Synthesis with End-to-End Model and Text Normalization”. In: *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*. 2020, pp. 179–184. DOI: 10.1109/NICS51282.2020.9335905.

[3] Wei Ping et al. *Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning*. 2018. arXiv: 1710.07654 [cs.SD].

[4] Kaizhi Qian et al. *AUTOVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss*. 2019. arXiv: 1905.05879 [eess.AS].

[5] Yi Ren et al. *FastSpeech 2: Fast and High-Quality End-to-End Text to Speech*. 2021. arXiv: 2006.04558 [eess.AS].

[6] Jonathan Shen et al. *Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions*. 2018. arXiv: 1712.05884 [cs.CL].

[7] Li Wan et al. *Generalized End-to-End Loss for Speaker Verification*. 2020. arXiv: 1710.10467 [eess.AS].

Research timelines:

Tasks
Build text Normalization package Vinorm
Build Phonetic Generate package Viphoneme
Train Mel-generator module (Tacotron2, FastSpeech2, GlowTTS)
Train Vocoder (Multiband-MelGan, HifiGan, Waveglow)
Train Vietnamese Speaker Verification model for generate Speaker Embedding
Research and propose new methods for Voice Conversion module: DeepSpeechVC
Apply Vietnamese Dataset for Conversion module by hyper-tuning the bottleNet
Combine all modules into end-to-end system
Conduct model evaluation by metrics
Write the thesis

Approved by the advisor	**Ho Chi Minh city, 18/03/2021**
Signature of advisor	***Signature(s) of student(s)***

TABLE OF CONTENTS

Page

Acknowledgement viii

Table of Contents xviii

List of Tables xxii

List of Figures xxiii

Abstract xxvi

CHAPTER 1 – INTRODUCTION

1.1 Overview of Voice Cloning 1

1.2 Motivation 5

1.2.1 Voice recreation with a audio of 3-5 seconds 5

1.2.2 Application for speech processing 5

1.3 Objectives 5

1.4 Project Content 6

CHAPTER 2 – PRELIMINARIES

2.1 Voice Formation 9

2.2 Acoustic feature representations 10

2.3 Deep learning for Language and Speech 14

2.3.1 Sequence to Sequence Models 14

2.3.2 Attention Mechanism 15

2.3.3 Variational Autoencoders 17

2.4 Loss function for Speech Processing.. 18

2.4.1 CTC 18

2.4.2 GE2E 19

2.5 Metric and evaluation in Speech Processing 20

2.5.1 Subjective measures... 20

2.5.2 Objective measures.. 20

CHAPTER 3 – RELATED WORK

3.1 Voice Cloning with Multi-speaker Speech Synthesis....................... 22

3.2 Voice Cloning with Voice Conversion 23

3.2.1 Parallel Voice Conversion 23

3.2.2 Non-Parallel Voice Conversion................................... 25

3.2.2.1 ASR & TTS Based Voice Conversion 25

3.2.2.2 VAE and Auto-Encoder Based Voice Conversion 26

3.2.2.3 GAN Based Voice Conversion 27

3.3 Speech Synthesis 28

3.3.1 Mel-spectrogram Generator 28

3.3.2 Vocoder.. 32

CHAPTER 4 – PROPOSED FRAMEWORK FOR VOICE CLONING BASED ON SPEECH SYNTHESIS AND VOICE CONVERSION

4.1 Voice Conversion.. 36

4.1.1 AutoVC 36

4.1.2 DeepSpeech2:........ 39

4.1.3 DeepSpeech-based voice conversion:........ 41

4.2 Speaker Encoder........ 44

4.3 Synthesizer........ 44

4.4 Vocoder........ 50

4.5 Voice Cloning System........ 51

CHAPTER 5 – EXPERIMENTS AND RESULTS

5.1 Dataset 54

5.1.1 Resource 55

5.1.2 Audio Pre-processing 56

5.2 Evaluation results........ 58

5.2.1 Voice Conversion........ 59

5.2.2 Speaker Encoder........ 61

5.2.3 Speech Synthesis 62

5.2.4 Vocoder........ 66

5.2.5 Voice Cloning Framework 67

CHAPTER 6 – CONCLUSION

6.1 Results........ 70

6.2 Ethic 71

6.3 Future works 71

References 74

Appendix

Appendix A – Text Normalization

A.1 Rules with Regular Expression 85

A.2 Dictionary Checking 88

A.3 Vietnamese Syllables Statistics 90

List of Publications

LIST OF TABLES

Table 5.1 List of datasets and the modules in which they are used for training and testing. 56

Table 5.2 Compare MCD result between AutoVC and DeepSpeech-based voice conversion on VCTK dataset. 60

Table 5.3 Compare MCD result between AutoVC-based VC System and DeepSpeechVC System on VCTK dataset. 69

Table A.1 Number of sentences from popular Vietnamese newspapers 90

Table A.2 Statistics on Tokens need to be standardized in Vietnamese 91

LIST OF FIGURES

Figure 1.1 Problem Statement of Voice Cloning. 2

Figure 1.2 Voice Cloning with Multi-speaker Speech Synthesis. 3

Figure 1.3 Voice Cloning with Speech Synthesis and Voice Conversion. 4

Figure 2.1 Descriptive Signal level analogy. [1] 10

Figure 2.2 The vocal organs, shown in side view. (Figure from OpenStax University Physics, CC BY 4.0) 11

Figure 2.3 Spectrogram visualization. 11

Figure 2.4 Mel-Spectrogram visualization. 12

Figure 2.5 Mel-scale formula. 13

Figure 2.6 Mel-scale conversion visualization. 13

Figure 2.7 MFCC visualization. 13

Figure 2.8 Sequence to sequence model for machine translation. (Figure from Stanford) 15

Figure 2.9 Attention architecture. (Figure from Stanford) 15

Figure 2.10 Attention score 16

Figure 2.11 Attention distribution 16

Figure 2.12 Attention output 16

Figure 2.13 Attention output and hidden state 17

Figure 2.14 Dot-product attention 17

Figure 2.15 Multiplicative attention 17

Figure 2.16 Additive attention 17

Figure 2.17 Variational Autoencoders architecture. 18

Figure 2.18 CTC Loss function. 19

Figure 2.19 GE2E loss function 19

Figure 3.1 Multispeaker-Tacotron2 Architecture [2] 22

Figure 3.2 Parrotron Architecture [3] 24

Figure 3.3 TTS-Skins Architecture [4]. 26

Figure 3.4 F0-AutoVC Architecture [5]. 27

Figure 3.5 CycleGan-VC Architecture [6]. 28

Figure 3.6 Tacotron2 Architecture [7]. 31

Figure 3.7 GlowTTS Architecture [8]. 32

Figure 4.1 An overview of voice cloning framework based on speech synthesis and voice conversion. 35

Figure 4.2 AutoVC Architecture [9]. 36

Figure 4.3 Bottleneck problem illustration [9]. 38

Figure 4.4 DeepSpeech2 pipeline 40

Figure 4.5 DeepSpeech2 Architecture 40

Figure 4.6 DeepSpeech-based voice conversion. 42

Figure 4.7 Conv2d output size calculation 43

Figure 4.8 Speaker Verification with Generalized End to End Loss [10]. 45

Figure 4.9 Pipeline for Text Normalization using Rule-based and Dictionary Checking. 46

Figure 4.10 Tacotron2 Architecture. 47

Figure 4.11 Duration Predictor of Fastspeech and GlowTTS. 50

Figure 4.12 Generator Architecture of HifiGAN [11]. 51

Figure 4.13 Voice cloning based on Voice Conversion with AutoVC. 52

Figure 4.14 Voice cloning based on Voice Conversion with DeepSpeechVC. 53

Figure 5.1 An overview of voice cloning framework based on speech synthesis and voice conversion. 54

Figure 5.2 Silent trimming for training set. 58

Figure 5.3 Histogram of the duration of audio in training set. 59

Figure 5.4 Clustering for D-Vector of Speakers with PCA algorithmn. 61

Figure 5.5 Compare EER result between Speaker Encoder trained on VCTK dataset (English) and ZaloAI dataset (Vietnamese). 62

Figure 5.6 Melspectrogram and Alignment Generated by Tacotron2. 63

Figure 5.7 Ground truth (WaveGlow) generating process. 64

Figure 5.8 Training Loss of GlowTTS for Vietnamese. 65

Figure 5.9 Evaluation on Synthesizers for Vietnamese. 66

Figure 5.10 Evaluation on Vocoders for Vietnamese. 67

Figure 5.11 Evaluation Process for Voice Cloning framework. 68

Figure A.1 Word Cloud for Vietnamese Abbreviation from Statistics on News. 88

Figure A.2 Statistics on Tokens need to be standardized in Vietnamese. 90

ABSTRACT

Speech processing is an important research area in the field of artificial intelligence, alongside computer vision and natural language processing. Speech synthesis has been studied since the late 1950s, through the stages, the voice becomes more and more intelligent and natural like a human and can run in real-time and on devices. However, there are still many problems that need to be solved before they can be put into practice. One of them is the problem of voice cloning, which helps to synthesize a person's voice with only an audio sample, in addition, we can extract the speaker's features to be able to switch that person's language.

In this thesis, we proceed to build a voice cloning framework for Vietnamese. The framework will be a pipeline that includes many speech processing modules such as voice conversion, mel-generator, vocoder and speaker encoder. We experimented on SOTA models of the E2E Speech Synthesis, including mel-spectrogram generators such as Tacotron2, Fastspeech2, MelGAN, GlowTTS, and vocoders including Waveglow, HifiGAN, then applied these models to Vietnamese. These models have been optimized to be able to run on Android devices without internet and GPU. For the voice conversion module, we have proposed a new method based on AutoVC and Deepspeech2, then conducted an evaluation to compare with international models in English, the result is much improved compared to the AutoVC model and therefore, this new method is also integrated into this framework.

For each module, we choose the best method based on training time, model size, MOS score, real-time performance, GPU utils, then combine with other modules of the framework to form an end-to-end pipeline that can perform voice cloning. To evaluate the quality of cloned voice, we compared the pipeline using our method, which is voice cloning with voice conversion based on Deepspeech2, and the pipeline with voice conversion module using AutoVC. Our DeepSpeechVC framework results have a mel cepstral distortion of 11.90, better than the one using AutoVC with a result of

13.67.

The new proposed voice conversion model helps to create the foundation for the study of Vietnamese voice cloning, the phoneme conversion and text standardization libraries are provided, these libraries can run on cross-platform (C++, Python, Android NDK) and have been practically applied in industrial products.

CHAPTER 1

INTRODUCTION

This chapter provides an overview of the development of Voice Cloning. We show practical applications of the problem, the latest approaches, analyze the strengths and weaknesses of each method as well as the direction we use. Then We explain the motivation which makes us choose this topic to build a framework for voice cloning. At the end of the chapter is the outline of this thesis.

1.1 Overview of Voice Cloning

The demand for personal voice is increasing to make devices, apps or virtual assistants more familiar to users. There are a number of solutions such as using transfer learning, which requires 5 to 30 minutes of voice data, and then retrains the model to adapt to the new voice. The quality of the voice is proportional to the number of audio samples used for the transfer. For voice cloning based on speaker characteristics, we only need one or several audio samples to extract speaker embedding, and use it to create cloned voices. This method does not require retraining the whole speech synthesis model.

Voice cloning can be used for prosody transfer, our voice can be transformed to speak different tones, timbers based on the expressive characteristics of other speakers. Further application is to help us sing with the tone and voice of our favorite singer. And the most practical application currently is voice cloning to help solve the pain of speech synthesis systems when put into practice, the problem of mix-code and cross language. Voice cloning has a potential future application for creating cross-lingual cloned voices. Most speech synthesis systems today only support one language, voice cloning helps us to speak any foreign language with our personal voice, whether we know that language or not. This application can be used in cross-country meetings.

Voice cloning is the task of generating a voice from another voice using only one or a

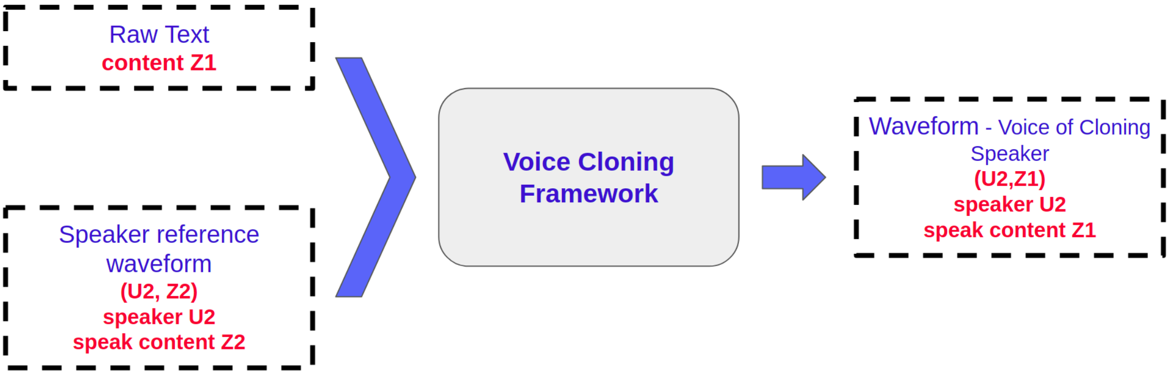


Figure 1.1: Problem Statement of Voice Cloning.

few voice samples.

The voice cloning problem requires building a model with:

- Input: a text or sentence that needs to be converted (Z1) and a reference speaker audio (U2)

- Output: an audio that contains the content of the text (Z1) with the voice of the reference speaker (U2)

In other words, this task is to create an audio that speaks a given text in any reference voice.

One of the approaches to this problem is to build a unified speech synthesis model, which generates the resulted audio directly from the input text and reference audio without any intermediary step. This unified speech synthesis model is called Multi-speaker Synthesiser.

In this approach, first the raw text will go through the Text Normalization and Phonetization module to be normalized and converted into phonetic representation features, in the other hand, the reference speaker audio will be passed through Speaker Embedding module to extract voice characteristic features. The phonetic features and the characteristic features are passed to Multispeaker Synthesizer module to generate the

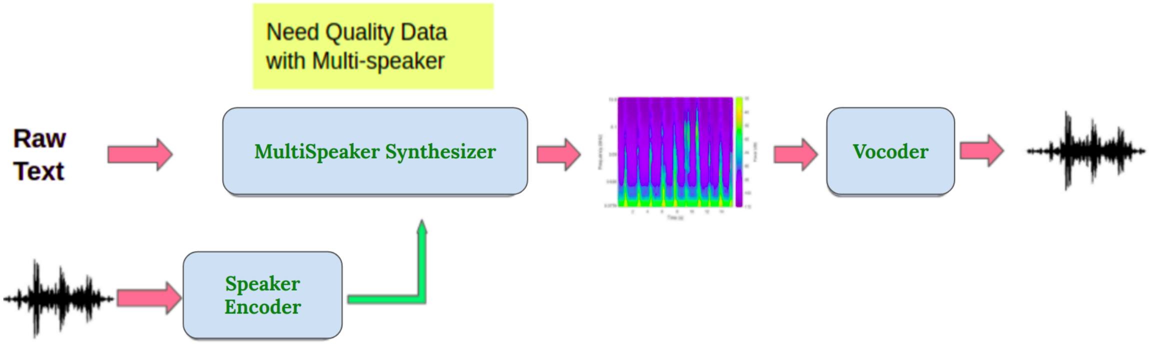


Figure 1.2: Voice Cloning with Multi-speaker Speech Synthesis.

mel-spectrogram, a kind of features generated from audio signal. Finally, the Vocoder will receive the generated mel-spectrogram to create the final audio result. This approach is proved to be efficient, such as Baidu DeepVoice3 [12]. However, as we tried this method with Vietnamese dataset, the results are disappointing. We suspect that the total recording hours, the number of speaker, the number of audio per speaker as well as the quality of Vietnamese datasets are not enough for this approach. Therefore, in this thesis, we propose another approach for voice cloning based on Speech-to-speech that can be applied using smaller amount of dataset and can still get considerable result.

Voice conversion techniques are used to modify the signal which is the output of speech synthesis to produce voice that includes variety of emotional expressions, duration, rhythm and multiple speakers. In this study we use it as a module that transforms speech characteristics between speakers rather than just modifying speaker prosody.

With Voice Cloning based on Voice Conversion, we need to build 4 separate module with data training requirements that are not too large at Speech-to-speech step: Synthesizer, Vocoder, Speaker Encoder and Voice Conversion.

Speaker Encoder is also used to extract speaker features, playing an important role in generating high-quality voice similar to the original voice. Contrary to the approach mentioned above, the Multispeaker Synthesizer in voice cloning is replaced by a Sin-

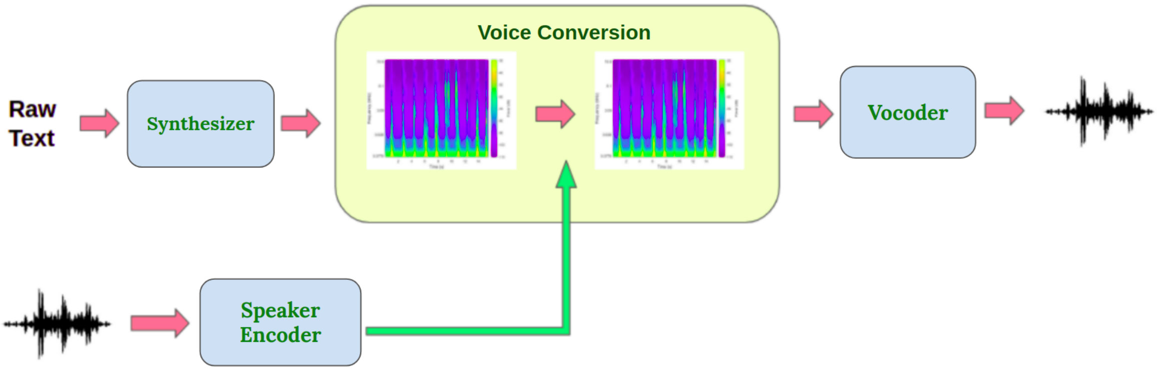


Figure 1.3: Voice Cloning with Speech Synthesis and Voice Conversion.

gle Speaker synthesizer that can only produce audio results with one speaker's voice. This voice will be considered as the original voice, we need to transform the voice based on the reference voice but make sure to keep the content of this original voice. Voice Conversion will take care of this task, it will modify the speech of a source speaker to make it sound like it was spoken by another target speaker while preserving the content of the speech, and the process is text-independent.

In this thesis, we will build an end-to-end framework based on the structure of Voice Cloning with Speech Synthesis and Voice Conversion, we conduct research, experiment and evaluation on each small module of the framework, then suggest improvements to each of those modules.

The two most important tasks are to build a voice conversion module and a robust speech synthesis system to produce quality source audio. The current voice conversion model used is AutoVC, but this method has many problems with sound quality and especially a bottleneck problem, which takes a lot of effort and time to tune. We propose a new voice conversion model based on auto-encoder and features extracted from automatic speech recognition, we call this method is DeepSpeechVC.

1.2 Motivation

1.2.1 Voice recreation with a audio of 3-5 seconds

With the motivation of reconstructing voices for people who are mute after an accident and there is a small amount of data about their voices, this thesis aims to conduct experiments and apply new technologies, artificial neural networks to synthesize voices for Vietnamese. Different from speech synthesis models that require a one-speaker quality data set - traditional voice cloning, we suggest a new voice clone system, which can synthesize any person's voice with only a few sample audio inputs of that person's voice

1.2.2 Application for speech processing

Speech translation is a system with the goal of helping people who speak different languages to communicate with each other, the task is to generate speech in the target language from the translated text. By applying the advantages of Voice Conversion, we can make it straightforward to retain the voice of the original speaker after translation.

If we can build a good Voice Cloning pipeline, we can apply it to data augmentation, the voice of an original speaker will be modified based on different reference voices, creating multi-speaker data, used to support training automatic speech recognition models.

1.3 Objectives

The main objectives of this project are:

1. Suggest new method to improve current voice conversion, AutoVC, improve the bottleneck limitations. Create a premise for researches on the voice conversion in English and Vietnamese.

2. Experiment on all the newest approaches of Speech Synthesizer, Vocoder then apply to Vietnamese. Use these models as modules that generate input and output for voice conversion.

3. Build a framework for creating a cloning voice which does not appear in the training data set, the input is text we want to say and the voice audio of the person to be synthesized.

1.4 Project Content

Our project report is structured into 6 chapters:

Chapter 1

Chapter 1 provides an overview of the development of Voice Cloning. We show practical applications of the problem, the latest approaches, analyze the strengths and weaknesses of each method as well as the direction we use. Then We explain the motivation which makes us choose this topic to build a framework for voice cloning. At the end of the chapter is the outline of this thesis.

Chapter 2

In chapter 2, we present the basic knowledge of speech and signal processing, common deep learning techniques used for signal processing problems are also explained, we introduce an overview of speech synthesis and the metrics used to evaluate the quality of speech processing problems.

Chapter 3

In chapter 3, the most popular and latest approaches to the voice cloning problem are briefly mentioned to get an overview of this problem. Two approaches to voice cloning are based on multi-speaker speech synthesis and based on voice conversion. For voice conversion, depending on the dataset, we use parallel or non-parallel approach. Recent voice conversion studies are all based on non-parallel because it does not require pairs of two-person utterances saying the same content. Prominent methods can be mentioned as Gan-based, VAE-based, ASR-based Voice Conversion. Outstanding methods of related tasks such as speech synthesizer, vocoder are also presented to combine with voice conversion to build a cloning framework.

Chapter 4

In chapter 4, we introduce Voice Conversion, which is the most important module in Voice Cloning. The architecture of AutoVC, one of the state-of-the-art techniques of Voice Conversion, is described in detail to analyze its advantages and limitations. To overcome these limitations, we briefly describe an automatic speech recognition model called DeepSpeech2, which is used as a content encoder. By applying this idea to AutoVC, features are efficiently extracted and help to improve bottleneck problem of the old model. To build a cloning pipeline based on Voice Conversion, we conduct research and test satellite modules, support input and output for voice conversion module. Speaker Encoder is used to extract speaker features, Speech Synthesizer plays the role of generating melspectrogram source as input for Voice Conversion's content encoder. The output of the decoder part of the Voice Conversion is fed into a Vocoder to generate the waveform. We evaluates on each module of the framework, then combine them to build demos and perform evaluations on the entire system.

Chapter 5

In chapter 5, we show the data that we use and the experiment results. To be able to compare and evaluate the quality of the solutions that we have proposed in chapters 4 and 5, we select suitable datasets for each problem, perform training and fine tuning for the models. Then we use the appropriate metrics, give analysis and evaluation, trade-off considerations to choose the best model to build an end to end framework.

Chapter 6

In chapter 6, we highlight the work involved in building a voice cloning based on speech synthesis and voice conversion. Then we summarize what knowledge we have research, experiments, evaluation results on models and solutions that we propose. Ethical issues of science and potential future research problems are also discussed.

CHAPTER 2

PRELIMINARIES

In this chapter, we present the basic knowledge of speech and signal processing, common deep learning techniques used for signal processing problems are also explained, we introduce an overview of speech synthesis and the metrics used to evaluate the quality of speech processing problems.

2.1 Voice Formation

To better understand the speech signal, we show how the human organs make the sound depicted in Figure 2.1.

The lungs have the function of creating an air stream, which puts pressure on the larynx (vocal folds) causing it to open and close periodically. This helps to generate sound wave frequencies with a fundamental frequency of about 125Hz for men, 210Hz for women, which is called F0 frequency. Different people will have different F0. In speech processing problems that have speaker independence, such as speech synthesis or automatic speech recognition, this frequency F0 will be eliminated. In contrast, in problems such as speaker verification, voice conversion, F0 is considered as an effective feature to distinguish speakers.

In order to form a voice, other organs are also needed such as: palate, tongue, teeth, lips, nose (Figure 2.2). These organs have the role of a ”resonator” like a guitar box, they have can change shape flexibly. This resonator has the effect of amplifying some frequencies, suppressing others to create sound.

Therefore the source spectrum will be transformed by the filter functions to form the voice. Output spectrum will have peaks called formants. The value, position, and time variation of these peaks produce diverse features for each phoneme.

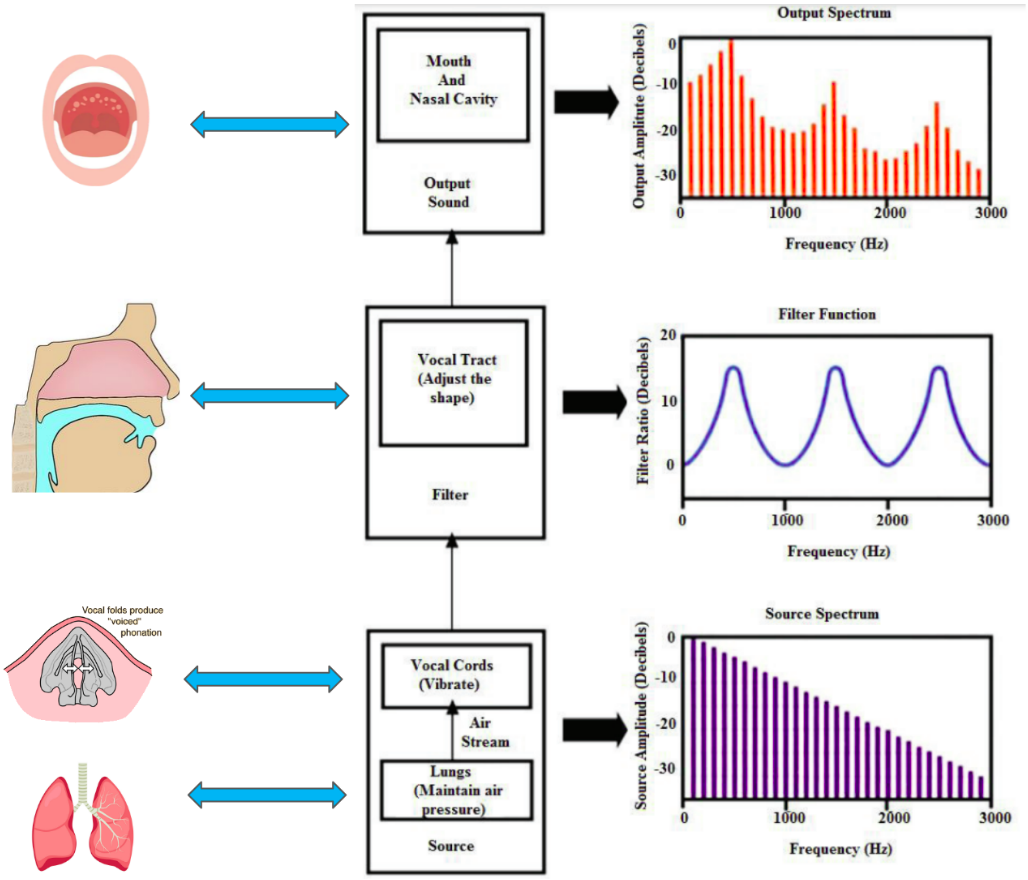


Figure 2.1: Descriptive Signal level analogy. [1]

2.2 Acoustic feature representations

Feature extraction is an important step in speech processing. Raw audio input is usually difficult to understand and hard to use. Therefore, to be able to train deep machine learning effectively, we need to extract more features from the initial audio input. There are many ways to extract features from audio and extract by signal domain is one of the most popular ways.

An audio signal contains three domains: time domain, frequency domain and amplitude domain. The waveform, which is extracted directly from audio, only contains the changing of amplitude with respect to the time domain. To include the frequency domain into extracted features, we use time-frequency representation of audio features, which can be obtained into using Short-Time Fourier Transform (STFT) on the raw

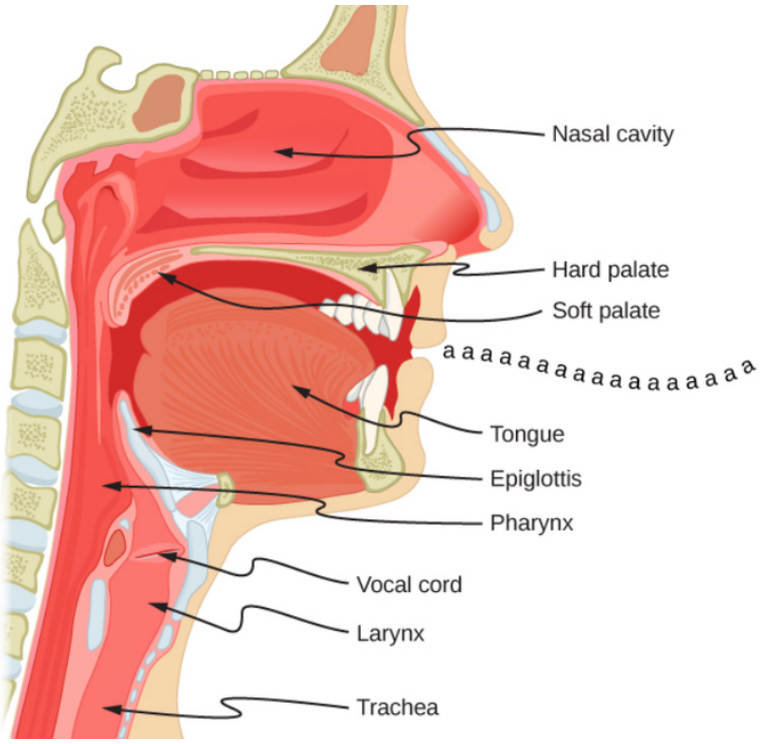


Figure 2.2: The vocal organs, shown in side view. (Figure from OpenStax University Physics, CC BY 4.0)

audio signal. After this, we will get Spectrogram, one of the common features to fed into a deep neural network.

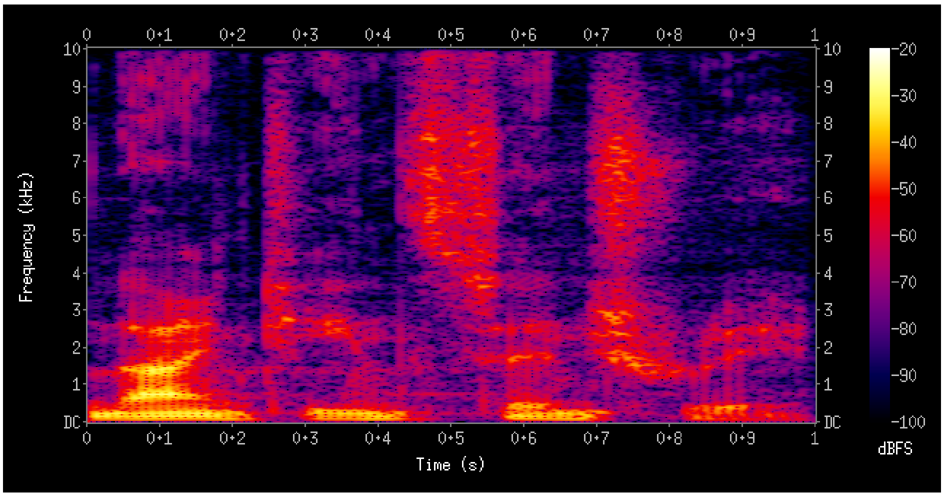


Figure 2.3: Spectrogram visualization.

As mentioned above, a spectrogram is a representation of features extracted that contain both time and frequency features of an audio signal by applying SFTF. In other words, we divide the audio signal into many small segments, then compute each signal using Fast Fourier Transform (FFT), and then concatenate all the outputs, we will get spectrogram representation. Apply this spectrogram with mel-scale will result in mel-spectrogram, another popular feature in signal processing.

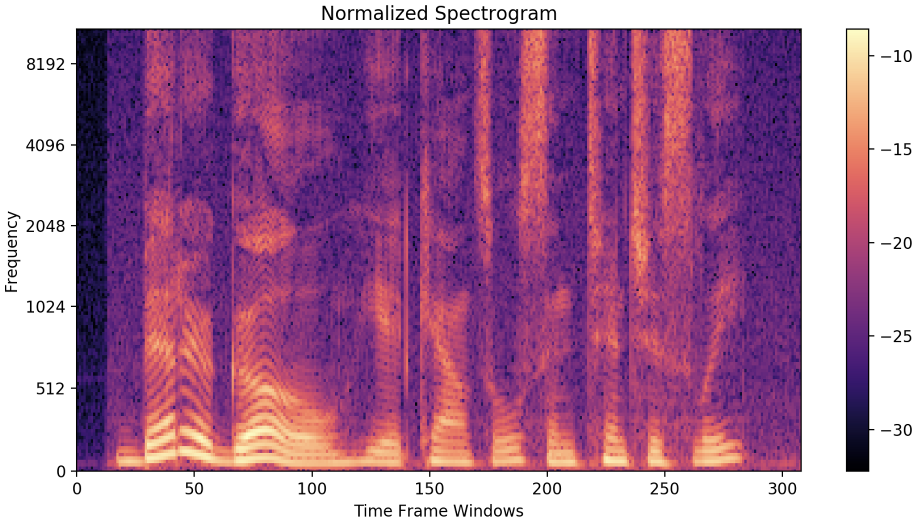


Figure 2.4: Mel-Spectrogram visualization.

Mel-spectrograms are spectrograms whose frequency is converted from current scale to mel scale. This feature extraction comes from the fact that the human ear recognizes the difference between low-frequency signals better than high-frequency signals. For example, we can recognize the difference between 200Hz and 300Hz sound signals better than the difference between 2000Hz and 2100Hz sound signals, despite having the same difference of 100Hz. This conversion from normal scale to mel scale can be computed using the formula shown in figure 2.5 and the visualization of mel-scale conversion results is shown in figure 2.6.

As the plot in figure 2.6 shows, the higher frequency the narrower the difference between nearby frequencies. And here we can go further in signal feature extraction by

$$m = 1127 \times log(1 + \frac{f}{700})$$

Figure 2.5: Mel-scale formula.

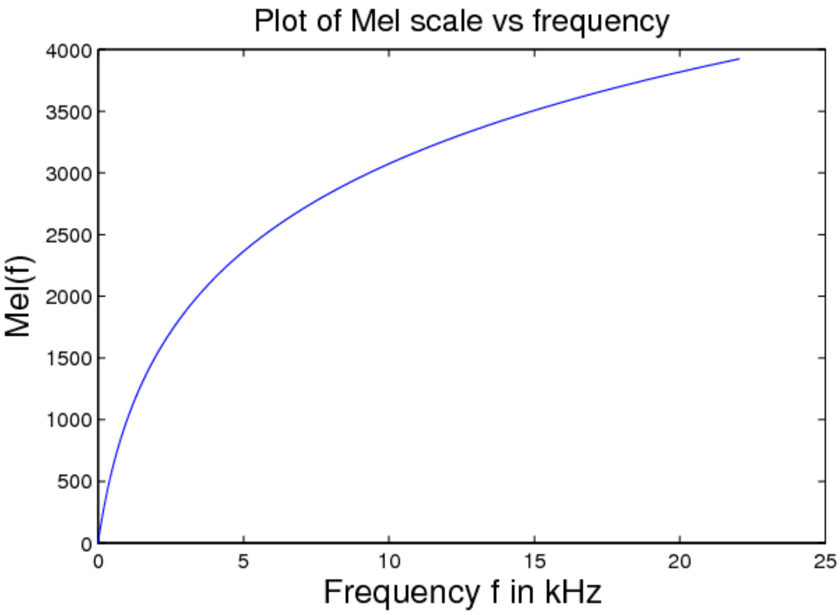


Figure 2.6: Mel-scale conversion visualization.

converting mel-spectrogram into MFCC (mel-frequency cepstrum coefficients).

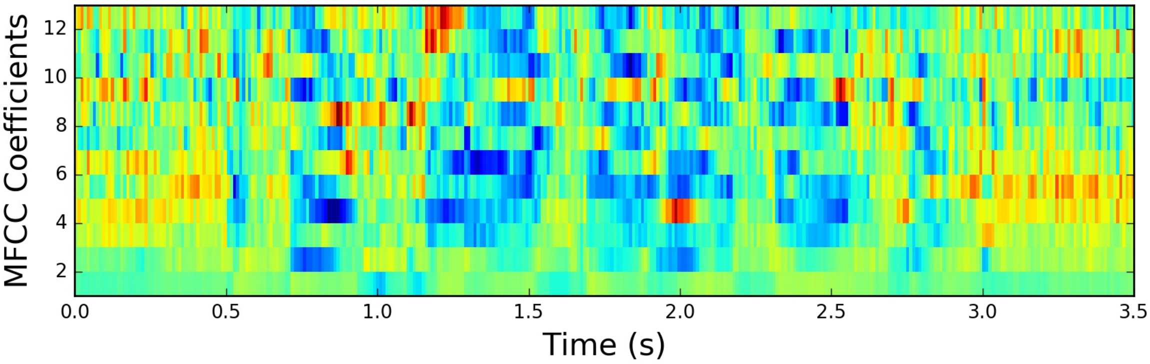


Figure 2.7: MFCC visualization.

MFCC is converted from mel-spectrogram by using DCT (discrete cosine transform).

In signal processing, MFCC is often used to describe the timbre of an audio signal.

Spectrogram, mel-spectrogram and MFCC, are all commonly used in signal processing. In our thesis, we use mel-spectrogram for speech representation, which is the kind of feature not only contains both time and frequency information but is also practical to use because of its relation to human hearing ability.

2.3 Deep learning for Language and Speech

In recent years, Deep Learning method become more popular and powerful in all aspect of Language and Speech processing. Here we will introduce some of the most important works to both our thesis and speech processing.

2.3.1 Sequence to Sequence Models

The sequence-to-sequence model consists of two recurrent neural networks called encoder and decoder [13]. This RNN model can be gated recurrent neural networks [14], long short-term memory network [15], bidirectional long short-term memory [16] and many other variants of RNN. The inputs and outputs of this model are sequences that can be an encoded text, a sequence of audio samples, a sequence of frames. The encoder part has the function to produce an encoding of the input, provide initial hidden state for the decoder part. The decoder part is based on the information extracted from the encoder in the form of hidden state, creating the target sequence output.

One of the biggest problems with sequence to sequence models is the bottleneck in the hidden state output of the encoder. The task of this hidden state is to capture information from the input input. Therefore, if the input sequence is too long, the information in the hidden state is lost. To overcome this shortcoming, a direct connection is connected between the decoder and the encoder to focus on important relevant information at each step of the decoder. That is the idea of the attention method.

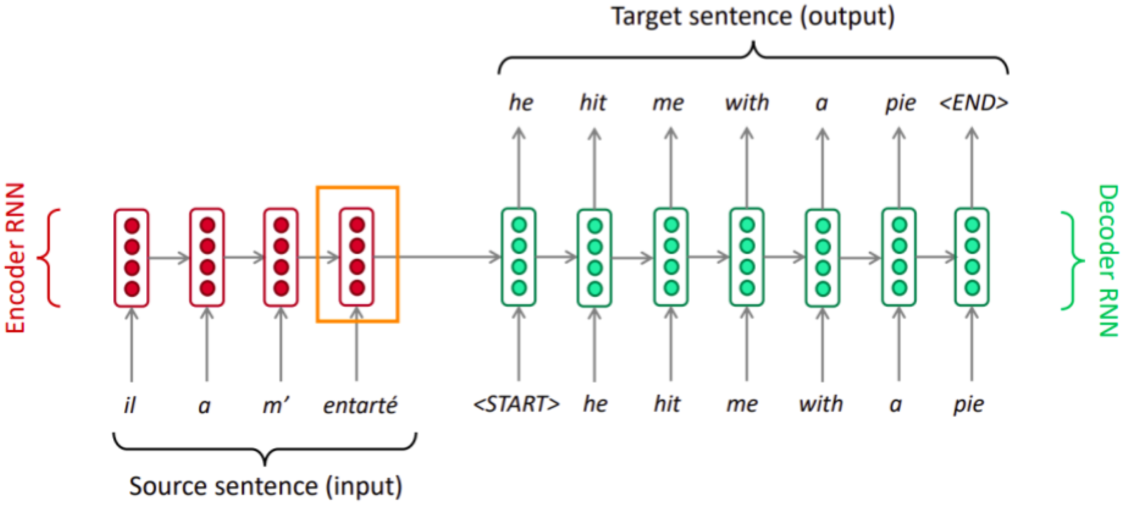


Figure 2.8: Sequence to sequence model for machine translation. (Figure from Stanford)

2.3.2 Attention Mechanism

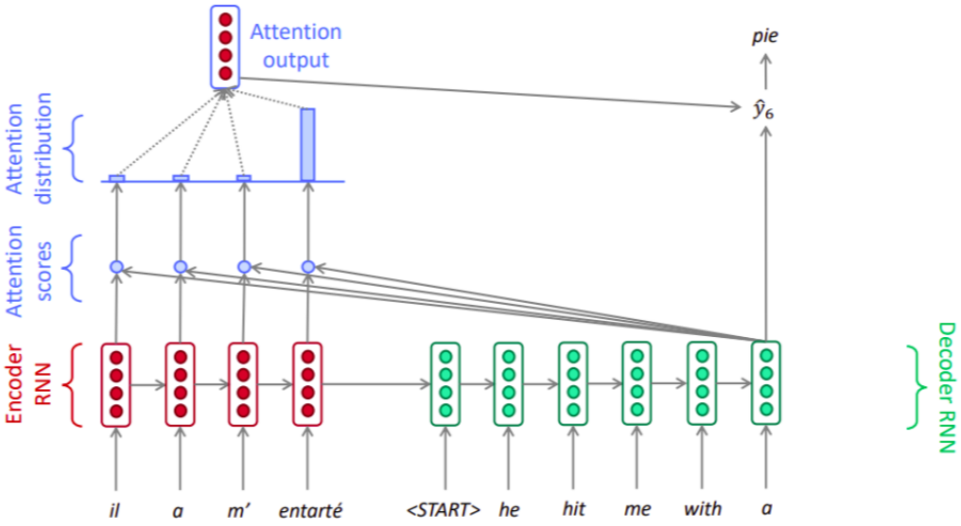


Figure 2.9: Attention architecture. (Figure from Stanford)

In order that the context information is not lost during the decode process of the sequence to sequence model, at each step of the decoder, the hidden layers will be mathematically performed with each state of the encoder. The most commonly used math operator is dot as in figure 2.9. As a result, we get the attention score. (Figure 2.10)

$$e^t = [s_t^T h_1, ..., s_t^T h_n] \in \mathbb{R}^n$$

Figure 2.10: Attention score

Then we take softmax with the attention scores, to form the attention distribution. This distribution tells us what parts of the encoder are important to the decoder's current state. (Figure 2.11)

$$\alpha^t = softmax(e^t) \in \mathbb{R}^n$$

Figure 2.11: Attention distribution

Attention output is performed by taking weighted sum of the hidden states of the encoder with the newly created attention distribution. This attention output will contain the information of the entire encoder, moreover, the important information in the current state will have a larger role thanks to distribution attention. (Figure 2.12)

$$a^t = \sum_{i=1}^{N} \alpha_i^t h_i \in \mathbb{R}^h$$

Figure 2.12: Attention output

There are many ways to use the attention output for the next states, the easiest one that can be used is to concatenate the decoder's hidden state with the generated attention output, consider it as more information about the encoder context, to decoder model can learn better. (Figure 2.13)

By using attention as additional information for the model, the decoder can access all

$$[a_t; s_t] \in \mathbb{R}^{2h}$$

Figure 2.13: Attention output and hidden state

the encoder information, the information will not be lost in the bottleneck part of the encoder's output layer. In addition, attention also helps to solve the vanishing gradient problem by allowing all the states in too long sequences to have direct access with the decoder.

In addition to dot attention as illustrated above (Figure 2.14), we also have other attention variants that are applied in many other sequence to sequence problems. Some variants can be mentioned such as multiplicative attention (Figure 2.15), additive attention. (Figure 2.16)

$$e^t = s^T h_i \in \mathbb{R}$$

Figure 2.14: Dot-product attention

$$e^t = s^T W h_i \in \mathbb{R}$$

Figure 2.15: Multiplicative attention

$$e^t = v^t anh(W_1 h_i + W_2 s) \in \mathbb{R}$$

Figure 2.16: Additive attention

2.3.3 Variational Autoencoders

An autoencoder is a neural network that is learned by using an unsupervised learning method. This model learns to encode the information from the input by attempting to reconstruct it in the training step. This information is visualized as a latent space,

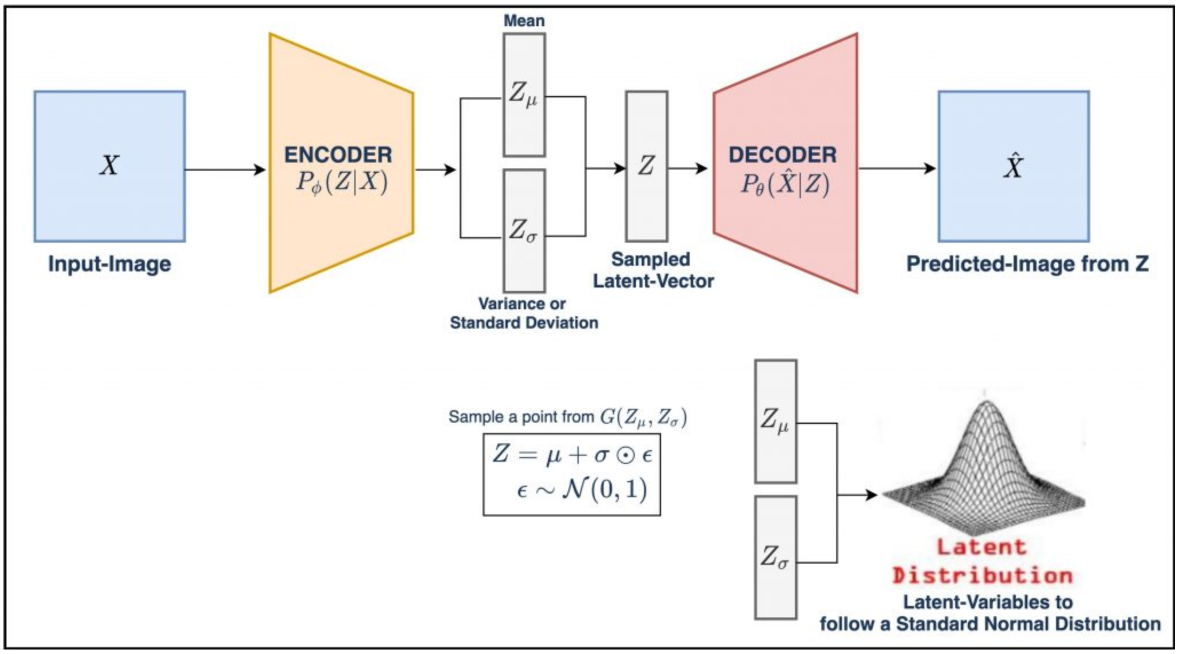


Figure 2.17: Variational Autoencoders architecture.

which is a collection of similar data that are close to each other.

A variational autoencoder is a kind of autoencoder. It addresses the problem of the vanilla autoencoder, which is the result latent space may not be continuous. By encoding the inputs as a distribution over latent space instead of as a single point, the model regularizes the latent space resulted in more accurate results.

2.4 Loss function for Speech Processing

2.4.1 CTC

CTC (Connectionist temporal classification) calculates the sum of all alignments between the input and label, as shown in figure Figure 2.18. It is usually used for training the sequence-to-sequence model.

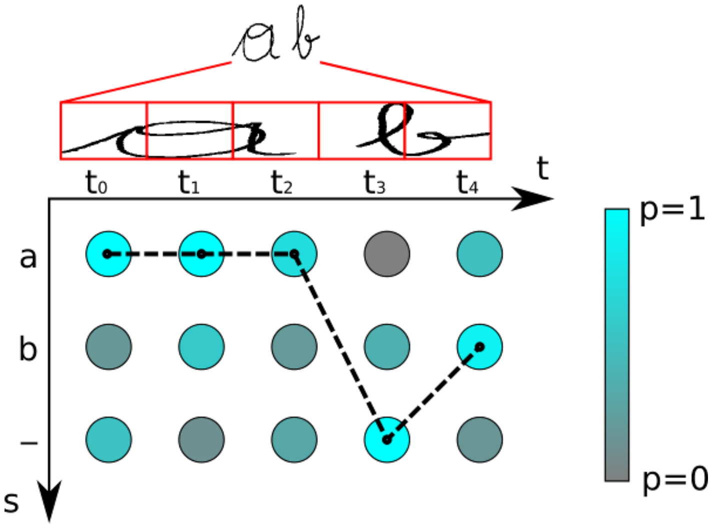


Figure 2.18: CTC Loss function.

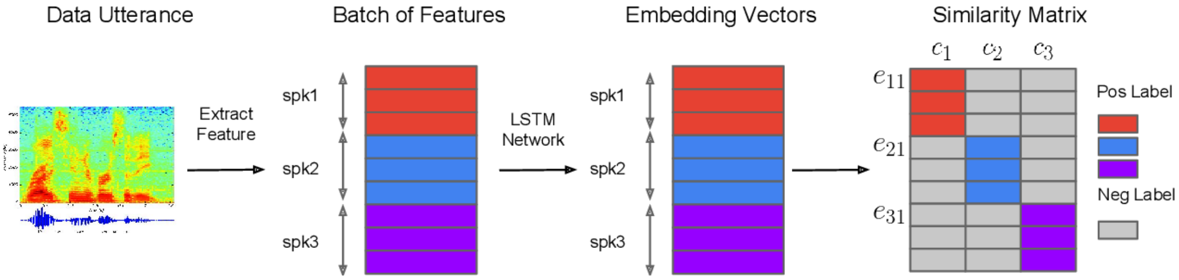


Figure 2.19: GE2E loss function

2.4.2 GE2E

GE2E (Generalized End-to-End) loss function uses the training approach that processes many data utterances from multi-speaker with a significant amount of utterances per speaker, as shown in figure Figure 2.19. This approach is shown to be able to update the model in a way that focuses on utterance examples that are difficult to verify.

2.5 Metric and evaluation in Speech Processing

2.5.1 Subjective measures

Subjective measures are evaluated by humans by having test participants listen to a random piece of audio synthesized by the computer and the audio read by humans, and then giving a score based on intelligence and naturalness. The systems used for evaluation are usually Amazon Mechanical Turk, in Vietnamese, the most popular way to measure is through the VLSP contest. [17]

MOS is the average score calculated through a test based on the listener's perception of the audios generated from the model. The scores are scored from 1 to 5 corresponding to the quality of the sentence (Bad, Poor, Fair, Good, Excellent). This method is done with many different people and many different sentences. The final result will be averaged from the score given by the evaluator

There is a commonly used method for collecting listener reviews called MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA). Listeners were asked to compare mixed signals between ground truth speech which is spoken by humans and generated signals and they were asked to assign a score on a scale from 0 to 100.

2.5.2 Objective measures

In order to make the results independent of the assessor, objective measures are introduced.

In 2019, Binkowski et al. introduce a quantitative automatic test called the Fréchet DeepSpeech Distance (FDSD) which is an adaptation of the Fréchet distance applied to the calculation of the distance between 2 speech files. This test allows having a distance score between the generated signal and the original file.[18]

Mel cepstral distortion is a measure that indicates the match between the actual speech and the sentence generated by the model. First, we have to extract the MFCC feature from the two audio files of the speaker and the generated model and split it into frames. Then, at each frame of the MFCC, calculate the distance of each pair of MFCC features of the natural voice and the synthesized voice. The smaller the MCD score, the closer the synthetic voice is to the natural voice.

Root mean squared error (RMSE) is a fast and simple measure commonly used, unlike MCD which only calculates the difference on the mel-spectrogram, the RMSE is performed on the waveform. RMSE is the square root of the mean of the square of all of the errors, which is the difference between two waveforms generated by the vocoder model and the original voice of the speaker.

For the speaker verification problem, Equal error rate (EER) is used to measure the system. This metric is usually used to compare devices and biometric technologies. The lower the measurement, the more reliable and accurate the system is. This measure is based on false positive rate and false negative rate, EER is the point on the graph with two equal FNR and FPR values.

CHAPTER 3

RELATED WORK

In this chapter, the most popular and latest approaches to the voice cloning problem are briefly mentioned to get an overview of this problem. Two approaches to voice cloning are based on multi-speaker speech synthesis and based on voice conversion. For voice conversion, depending on the dataset, we use parallel or non-parallel approach. Recent voice conversion studies are all based on non-parallel because it does not require pairs of two-person utterances saying the same content. Prominent methods can be mentioned as Gan-based, VAE-based, ASR-based Voice Conversion. Outstanding methods of related tasks such as speech synthesizer, vocoder are also presented to combine with voice conversion to build a cloning framework.

3.1 Voice Cloning with Multi-speaker Speech Synthesis

Multi-speaker Speech Synthesis is a voice cloning approach that adapts a TTS model into a model that can clone many different kinds of voice. There are many models using this approach, includes MultiSpeech [19], Baizhu's DeepVoice models [20], and one of the most representative method is the Multispeaker-Tacotron2 with speaker verification transfer learning method.

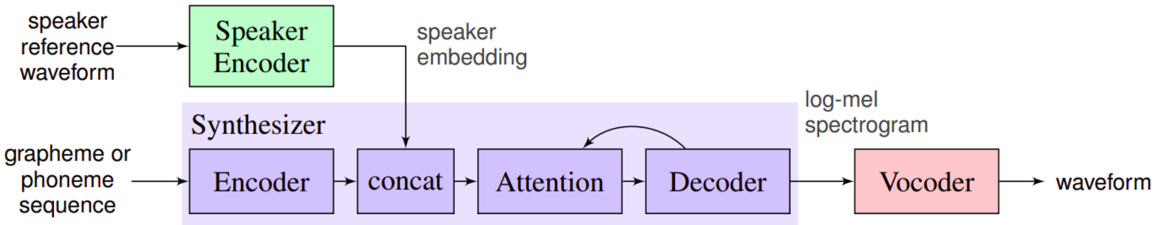


Figure 3.1: Multispeaker-Tacotron2 Architecture [2]

This method includes three components that are trained independently: a speaker encoder module that is trained for verification task, which is used to extract speaker voice characteristics from the reference audio; a tacotron2-based speech synthesis module used to generate output mel-spectrogram based on the input text and speaker encoder a vocoder module used to convert mel-spectrogram into waveform. This model is able to generate substantial audio results. However, when we experiment with this method to clone Vietnamese voices, the results are significantly worse due to the lack of quality and amount of audio samples in Vietnamse datasets.

3.2 Voice Cloning with Voice Conversion

Voice Conversion is the process of converting from an original speech into new speech with the voice of the speaker in the reference speech and the content of the orignal speech. Voice Conversion includes two main approaches: Parallel Voice Conversion and Non-Parallel Voice Conversion.

3.2.1 Parallel Voice Conversion

Parallel Voice Conversion includes all the methods that are trained by using parallel multi-speaker datasets - datasets that contain audio samples spoken by the same person. In other words, we need a corpus containing speech pairs where the two speakers utter the same sentences. Some solutions to this problem are based on Gaussian mixture models such as Voice conversion based on maximum likelihood estimation of spectral parameter trajectory [21], using partial least squares regression [22] or continuous probabilistic transform [23]. Some other voice cloning methods that use non-negative matrix factorization such as exemplar-based voice conversion using sparse representation in noisy environments [24] or with residual compensation for voice conversion [25].

With the advent of deep learning models, many methods have been proposed based on sequence to sequence [26], such as Voice conversion using deep neural networks with speaker-independent pre-training [27], others using highway networks [28], bidirectional long short-term memory based recurrent neural networks [16]. There is also a method to use the advantages of GAN [29].

Due to the inconvenience and limitation in using the dataset, only a few methods using this approach and one of the most representative methods in this approach is Parrotron [3].

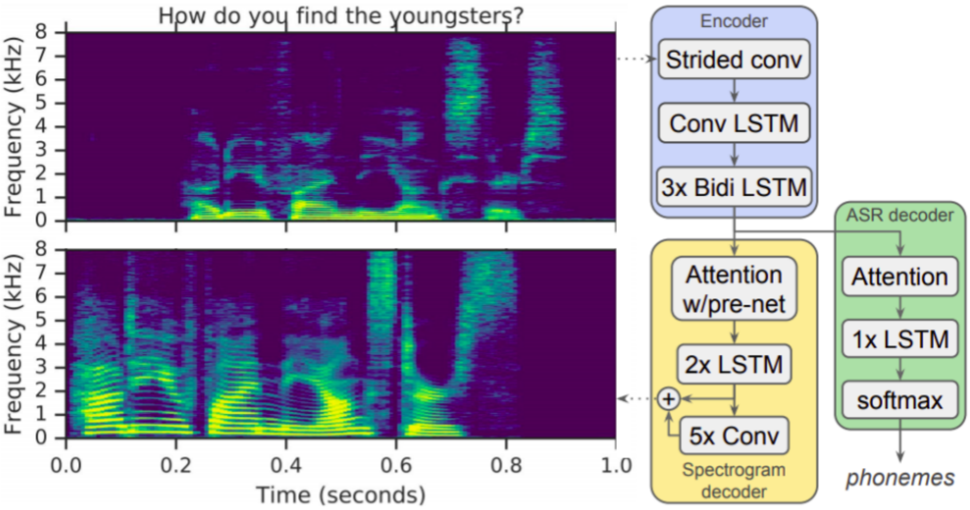


Figure 3.2: Parrotron Architecture [3]

Parrotron is a voice conversion model that converts the input mel-spectrogram into the output mel-spectrogram without any reference mel-spectrogram. The model can be used for normalization, which is the process of converting speech from any voice into a consistent speaker voice. Furthermore, it can be used to perform speech separation. This model uses an ASR decoder to able to produce more robust results. The method is trained using a parallel dataset, which was produced using a TTS system. The dataset is not public and the method is mainly used to convert into a consistent voice, not just any voice, therefore this method is not suitable for our objective. However, the use of

ASR decoder inspires and leads to our method.

3.2.2 Non-Parallel Voice Conversion

Non-Parallel Voice Conversion approach only needs datasets that contain audio samples from multiple speakers. This approach is more versatile, therefore, there are many methods using this approach. The quality and conversion effect obtained with non-parallel methods are usually limited compared with methods using parallel data due to the disadvantage related to the training condition.

3.2.2.1 ASR & TTS Based Voice Conversion

ASR (Automatic Speech Recognition) & TTS (Text to speech) Based Voice Conversion includes methods that use ASR and TTS as modules to extract content features from speech. Those modules are mostly trained beforehand to learn how to convert speech into text and then its encoder part will be used to extract content features.

We list several approaches for voice conversion based on ASR & TTS like FragmentVC which fuses fine-grained voice fragments with attention [30], Cotatron uses transcription-guided speech encoder [31]. Some use phonetic posteriorgrams [32], disentangling speaker and content representations with instance normalization [33] or cascading ASR and TTS with Prosody Transfer [34] [35]. One of the most representative methods is TTS-Skins.

TTS-Skins is a convolutional model that is used to convert voices. The model includes two modules: the encoder is a pre-trained ASR model, which is used to recognize voices, and the WaveNet-based decoder, which is used to convert into speech. This conversion method is close to our method, the difference is the number of used modules.

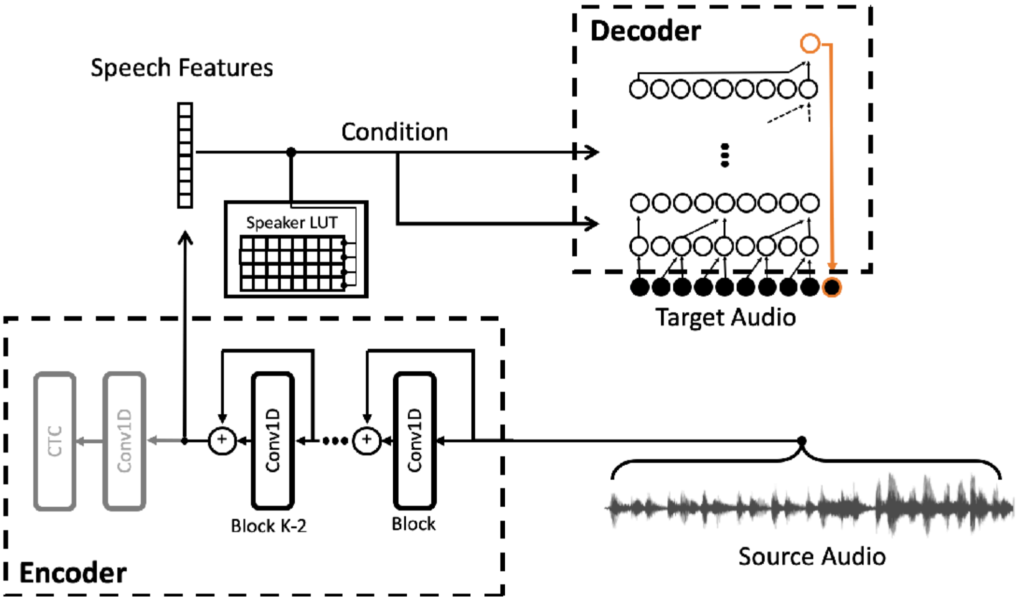


Figure 3.3: TTS-Skins Architecture [4].

3.2.2.2 VAE and Auto-Encoder Based Voice Conversion

VAE and Auto-Encoder Based Voice Conversion are methods using Auto-Encoder module that can be trained using non-parallel dataset and still get meaningful results. Models using VAE and Auto-Encoder Based methods are Unsupervised Learning [36]. Some methods use Triple Information Bottleneck [37], Vector Quantization [38] or WaveNet autoencoders [39]. ACVAE-VC is a voice conversion model that uses auxiliary classifier variational autoencoder [40] and Blow use single-scale hyperconditioned flow [41]. There are many methods using this approach and two of the most representative method are Autovc and F0-Autovc [5].

AutoVC is one of the important models relate to our thesis. It includes three modules: the Content Encoder to extract content features, the Speaker Encoder to extract the voice characteristic of the speaker, and the Decoder to decode the concatenation of the outputs of Content Encoder and Speaker Encoder. Although this method is able to generate SOTA results, the F0 of results are not consistent due to BOTTLENECK problem, which we will introduce later in our thesis.

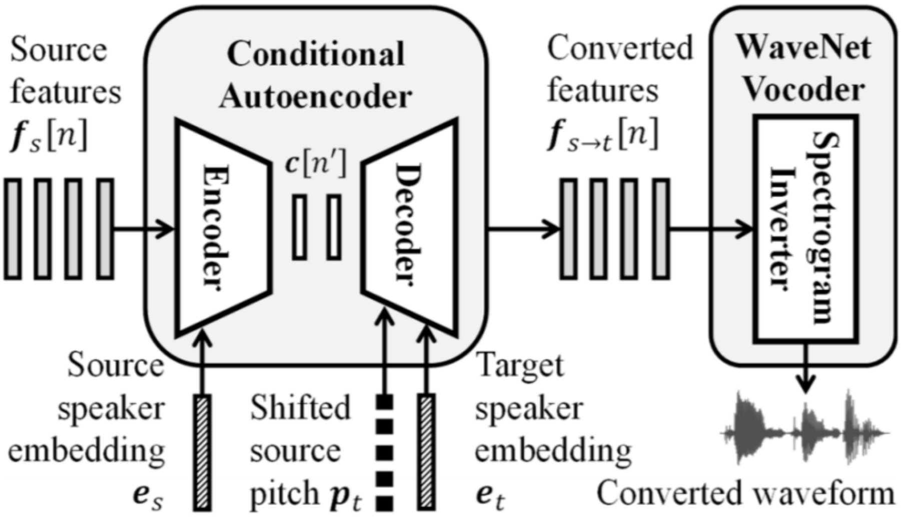


Figure 3.4: F0-AutoVC Architecture [5].

F0-Autovc is the next generation of AutoVC, which is able to generate more F0-consistent results. To solve the F0-inconsistent problem in AutoVC model, they extract the F0 features from the original audio and then pass through the Decoder module for training.

3.2.2.3 GAN Based Voice Conversion

GAN (Generative adversarial networks), which was introduced in 2014, is one of the classes of machine learning frameworks. This method includes two modules: Generator and Discriminator. The Generator module will generate the result based on the inputs and the Discriminator module will check if the result is fake or not. The Generation will be trained to generate good enough results to confuse the Discriminator. Therefore, the main results will generate by the Generator module, while the Discriminator module mostly used to train the model indirectly in an unsupervised manner.

There is GAN-based voice conversion method, in which the Generator will generate good enough speech results as real as possible to fool the Discriminator. Some of the most representative models in this approach is CycleGAN-VC [6] and StarGAN-VC which use star generative adversarial networks [39].

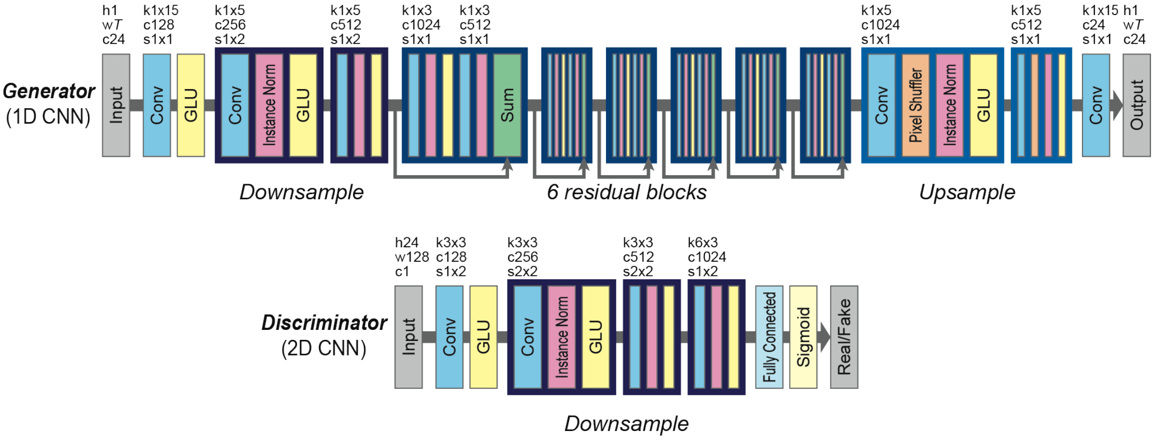


Figure 3.5: CycleGan-VC Architecture [6].

CycleGAN-VC was adapted into voice conversion from the CycleGan model, which is used in image conversions. Its generator module is 1D gated convolution neural network (Gated CNN) while the discriminator is a 2D Gated CNN. The model receives Mel-cepstral coefficients as inputs.

3.3 Speech Synthesis

The modern Speech Synthesis system consists of two main steps: generating frequency representation for input text, then a second module called vocoder used to generate the speaker's voice in waveform form.

3.3.1 Mel-spectrogram Generator

The Mel-spectrogram Generator is a Feature Prediction module, which converts strings of characters into speech representations necessary for the finalization of the generation, and the feature used here is the mel-spectrogram.

With the mel-spectrogram generator, there are methods such as Feed-Forward Transformer [42] or Attention based [43], in which the most typical is Tacotron2 [7] and

FastSpeech2 [44].

Since the input of this generator is a sequence of encoded characters or phonemes and its output is a sequence of frames of mel-spectrogram, therefore, the mel-spectrogram generator is a time series model.

There are two types of mel-generator models: Autoregressive and Non-autoregeressive. Autoregressive (AR) models are models that generate mel-spectrogram by using the results from previous mel-spectrogram's frame to predict the next mel-spectrogram's frame. The other type is Non-autoregressive (NAR), which are models that generate all mel-spectrogram's frames in parallel. Non-autoregeressive models have better generation speed but lower accuracy compare to Autoregressive models [45].

Auto Regressive

Mel-Generator models using autoregressive can be mentioned as TransformerTTS [46], Deep voice [47], Durian [48], Flowtron [49], Non-attentive tacotron [50], Robutrans [51], and the most prominent is the Tacotron2 model.

Tacotron2 includes a recurrent sequence-to-sequence feature prediction network that maps input text to mel-scale spectrograms, with a highlight that is the attention mechanism. The input text can be split into a list of unicode characters, a list of graphemes or a phoneme sequence. The representation for the input text plays an important role for training, the faster the model converges when there is a large correlation between the represented text and the audio bands. In addition, a suitable phoneme code will help in the inference process, the output will be flexible with many different vocabularies. We will detail methods to represent phoneme, and suggest ways to apply them to Vietnamese in the proposed method section.

The list of text representations is then encoded forms a vector, each phonetic unit corresponding to a one-hot vector whose length is the size of the declared character set. Tacotron2 model contains three parts: an encoder, a decoder and the most important one is attention. The encoder extracts content features vectors from the input text and then passes them to the decoder. The generated vector will be extracted features through three layers of Convolution Layer, these layers aim to filter the features showing the correlation between phonemes in the same context. Tacotron 2 also use Local Sensitive Attention, which extends the additive attention mechanism, as a step between the encoder and the decoder.

At the Decoder step, frames are generated at each state of two LSTM layers with the input information being the previous frame after passing through two fully connected layers, the Encoder vector and the context vector obtained from the Attention mechanism. These frames are then passed through five post-net convolution layers and added to themselves to generate the mel spectrogram. The error function used is negative log-likelihood to maximize the probability of each frame. The generated mel-spectrogram will be the input to a vocoder model to generate speech or a voice conversion model to transform the characteristics of this mel-spectrogram before generating the speech signal. We will describe in detail each layer of tacotron2 when applying for Vietnamese speech synthesis in the proposed method.

Models that use autoregressive are often difficult to deploy to run in real-time because the prediction of mel-spectrogram frames must depend on previous frames. To use these models we can perform optimization for it.

Non-Auto Regressive

In order to improve the limitations of Autoregressive models during inference, several techniques including knowledge distillation and source-target alignment have been proposed to bridge the gap between AR and NAR models.

Mel-Generator models using non-autoregressive can be mentioned as Fastpitch [52], Fastspeech2 [44], FlowTTS [53], Speedyspeech [54], Parallel Tacotron [55], Wave-Tacotron [56], JDI-T[57], EATS [58].

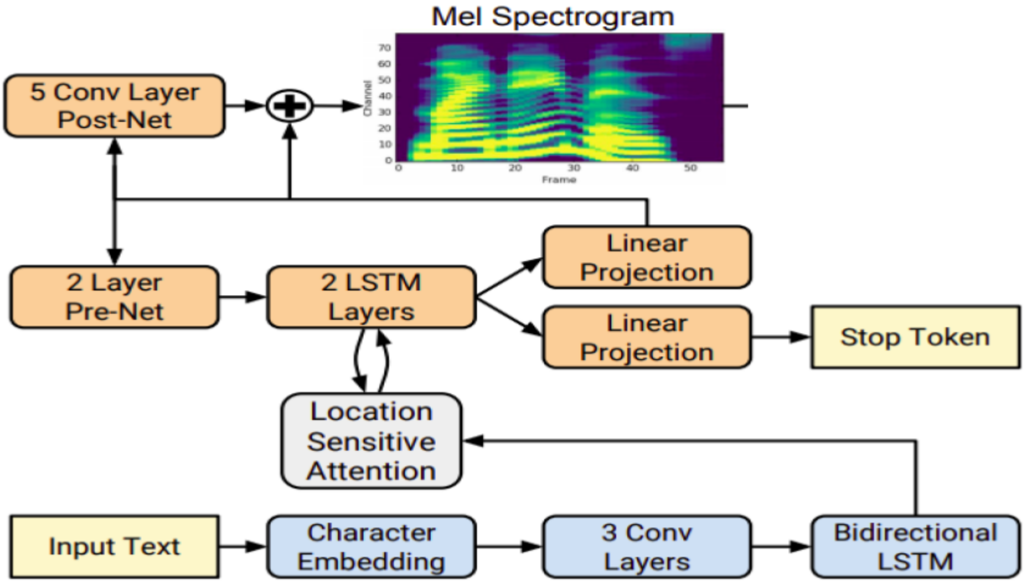
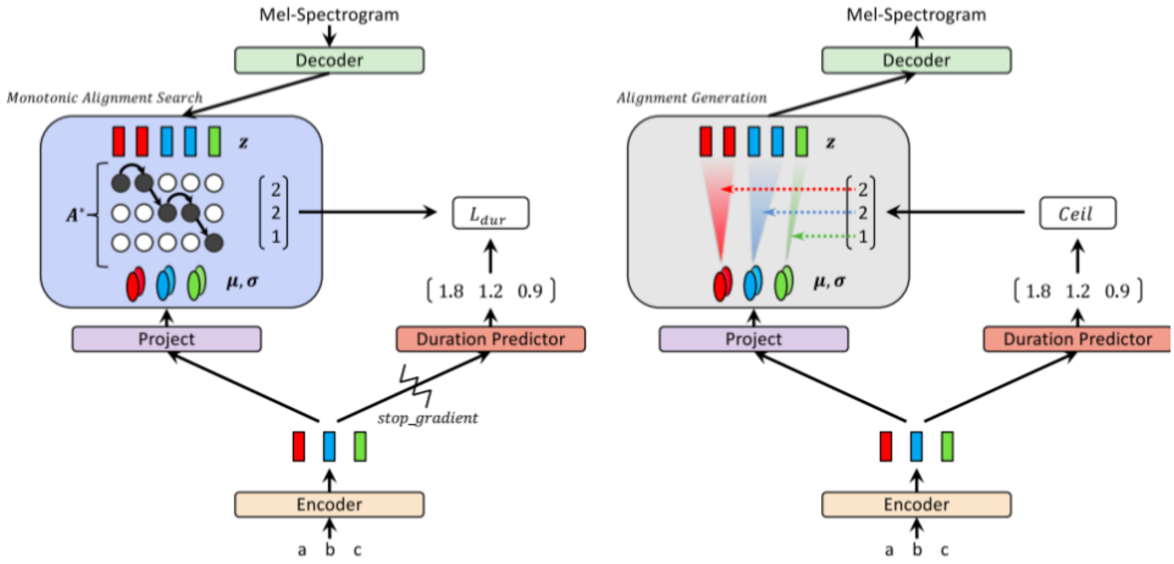


Figure 3.6: Tacotron2 Architecture [7].

We will briefly present one of these models, which is GlowTTS [8]. GlowTTS finds the suitable alignment between text features and speech features based on the idea of the hidden Markov model. By applying the invertible transformations, it can generate the resulted mel-spectrogram base on the input text in parallel. In other words, GlowTTS is based on AlignTTS and Flowtron, and then proceeds to improve the disadvantages of these two models.

Thanks to the above advantages, GlowTTS can control the speaking rate or pitch of synthesized speech, which we can use for prosody transfer or style modeling problems, and GlowTTS can synthesize parallel mel-spectrograms faster than Tacotron 2 15.7 times.

Besides, if we compare GlowTTS with other non-autoregressive models, it doesn't need extra external aligners. Specifically, for Fastspeech2, which is also a NAR model, the training process needs to have information such as duration of phonemes and attention maps pre-extracted from external aligners, such as Tacotron2. Therefore we need



(a) An abstract diagram of the training procedure. (b) An abstract diagram of the inference procedure.

Figure 3.7: GlowTTS Architecture [8].

to train a Tacotron2 model before we want to train the Fastspeech2 model, these parallel models critically depend on the autoregressive model. And as a NAR, GlowTTS also has limitations on the quality of the synthesized voice.

3.3.2 Vocoder

The input of vocoder models is acoustic parameters, for today's end-to-end models, these input features are mel-spectrogram, the vocoder's task is to synthesize speech signals close to the human voice.

In 2016, WaveNet was introduced, which is a combination of wavelet and neural networks, this technique estimates waveform samples from given input feature vectors - mel-spectrogram in speech synthesis[59]. WaveNet is a vocoder, which improves the synthesis process better than previous techniques, but the weakness of Wavenet is that sequential generation is too slow for production environments, leading to the introduction of CNN-based such as Multi-Scale Convolutional Neural Network for spectrogram inversion [60], Flow-based and GAN-TTS approaches. Flow-based approaches can be mentioned as Parallel WaveNet[61], ClariNet[62], FloWaveNet[63], the most typical of which is WaveGlow[64].

Griffin-Lim

The Griffin-Lim Algorithm (GLA) is a reconstruction algorithm used to convert mel-spectrogram into waveform. It is based on the redundancy of Short-Time Fourier transform (STFT). This algorithm generates the final resulted waveform by keeping estimating the target waveform's missing informations between two domains: time and frequency. It can generate consistent results without the knowledge of the target waveform.

WaveGlow

WaveGlow, which is a combination of two models WaveNet and Glow, is a generative model that uses flow architecture to estimate parameters to build a probability density function for data so that it can generate audio by sampling from a distribution. This conditional probability distribution is modeled against a class of convolution layers. In this model, the CNN layers do not have a pooling layer to produce output in the same dimension as the input. The model generates a discrete distribution on the next value based on a softmax function. This function is used to optimize the Maximize Likelihood Estimator problem and find the best set of parameters for data.

MelGAN

MelGAN is a vocoder model that generates speech audio based on generative adversarial networks. Compared to WaveNet model, Melgan is a non-autoregressive model with less number of parameters. The Melgan model contains many fully convolutional layers, so the advantage of this model is that the inference speed is greatly improved, the training process requires less GPU resources, which can be used to run on real-time applications.

HifiGAN

By using many patterns of different lengths, HifiGAN can learn various periods of voice audio. Each ResBlock will have a different kernel size and dilation. As a result, the generated voice will be of high quality and have a much faster synthesis time than models using auto regressive. It can run real time on CPU 13.4 times faster than models like Tacotron2.

CHAPTER 4

PROPOSED FRAMEWORK FOR VOICE CLONING BASED ON SPEECH SYNTHESIS AND VOICE CONVERSION

In chapter 4, we introduce Voice Conversion, which is the most important module in Voice Cloning. The architecture of AutoVC, one of the state-of-the-art techniques of Voice Conversion, is described in detail to analyze its advantages and limitations. To overcome these limitations, we briefly describe an automatic speech recognition model called DeepSpeech2, which is used as a content encoder. By applying this idea to AutoVC, features are efficiently extracted and help to improve bottleneck problem of the old model. To build a cloning pipeline based on Voice Conversion, we conduct research and test satellite modules, support input and output for voice conversion module. Speaker Encoder is used to extract speaker features, Speech Synthesizer plays the role of generating melspectrogram source as input for Voice Conversion's content encoder. The output of the decoder part of the Voice Conversion is fed into a Vocoder to generate the waveform. We evaluates on each module of the framework, then combine them to build demos and perform evaluations on the entire system.

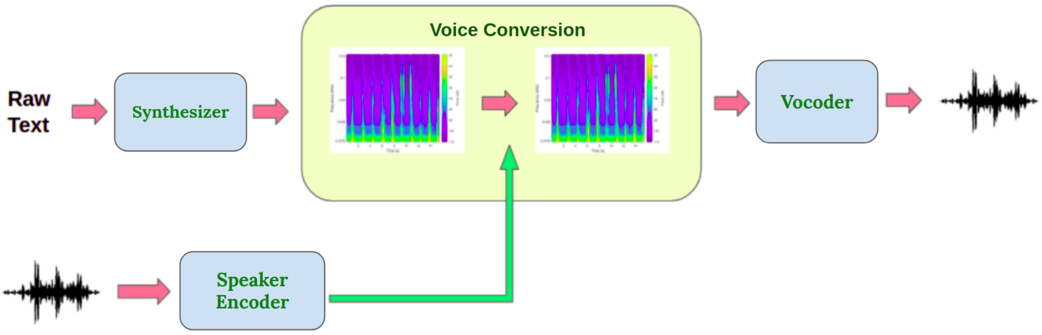


Figure 4.1: An overview of voice cloning framework based on speech synthesis and voice conversion.

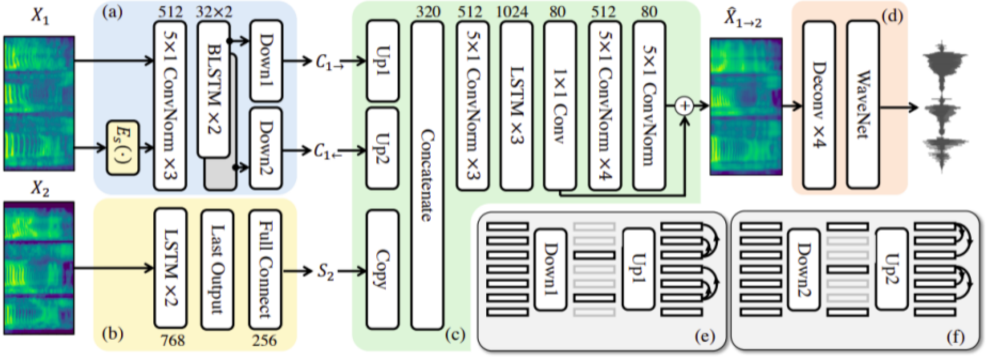


Figure 4.2: AutoVC Architecture [9].

4.1 Voice Conversion

Voice Conversion is the most important module in our Voice Cloning approach. We introduce one of the state-of-the-art techniques of Voice Conversion in 2019, AutoVC. The architecture of AutoVC is described in detail to analyze its advantages and limitations. To overcome these limitations, we briefly describe an automatic speech recognition model called DeepSpeech2, which is used as a content encoder. By applying this idea to AutoVC, features are efficiently extracted and help to impove bottlenet problem of the old model. This proposed voice conversion model is called DeepSpeech-based Voice Conversion.

4.1.1 AutoVC

AutoVC is a speaker conversion model used to convert speech to speech. This model was introduced in 2019 and is able to generate better results and perform “zero-shot” voice conversion. By using the auto-encoder loss function, this method is able to be trained using non-parallel datasets - datasets that do not contain sentences that are spoken by two different speakers.

In this paper, we use this model to convert speech from the single-speaker synthesizer to other speech results that contain the reference speaker voice. The AutoVC model consists of four modules: speaker encoder, content encoder, decoder and vocoder.

- With the speaker encoder, we use a pre-trained speaker embedding module, which we describe in chapter 5, to extract the d-vector. According to the AutoVC method, we will use the method to extract d-vectors from both the original audio input (X1) and the reference audio input (X2).

- The encoder module is used to extract features from each frame of the input mel-spectrogram. Its input include the 80 channels mel-spectrogram of the original audio and the d-vector of the original audio. The d-vector is concatenated with each time step of the mel-spectrogram beforehand. After that, the input is passed to those layers:

+ Three 512 channels convolutional layers with batch normalization and ReLU activation.

+ Stack of two bidirectional LSTM layers, the cell size of those LSTM will be called bottleneck size.

+ A downsampling module, which reduces the number of spatial dimensions.

- The decoder module then uses the feature output of the encoder module to convert into the mel-spectrogram features. It consists of:

+ An upsampling module, which increases the number of parietal dimensions to the same of the dimensions before the upsampling module.

+ Three 5×1 convolutional layers with 512 channels, each followed by batch normalization and ReLu three LSTM layers with cell dimension 1024.

+ A Projection to dimension 80 with a 1 × 1 convolutional layer

+ Five 5×1 convolutional layers, where batch normalization and hyperbolic tangent are applied to the first four layers. The channel dimension for the first four layers is 512 and goes down to 80 in the final layer.

Finally, the vocoder module uses the output mel-spectrogram from the decoder module

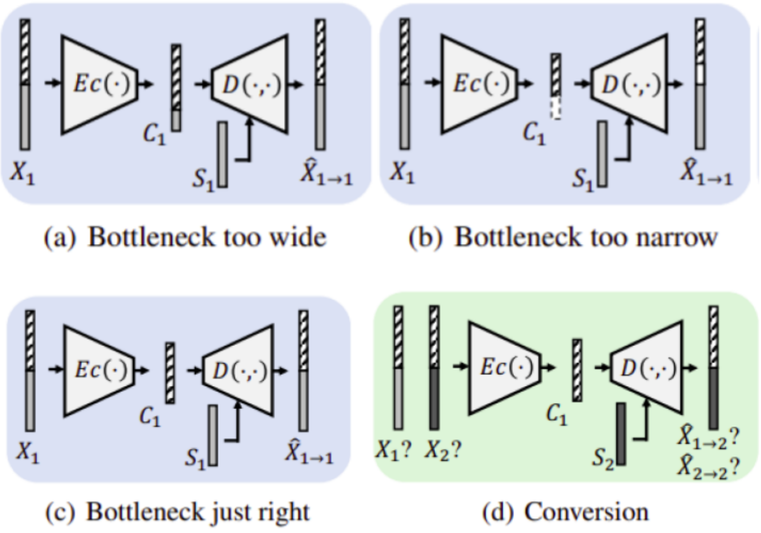


Figure 4.3: Bottleneck problem illustration [9].

to convert it into speech. The author apply the WaveNet vocoder as introduced in Van Den Oord et al. (2016), which consists of four deconvolution layers.

The size of the output feature must be chosen by hand and can lead to the bottleneck problem.

BOTTLENECK PROBLEM:

This problem happens when the output's size of the content encoder is too small or too big:

- When the size is too small, the encoder can not extract enough information from the speech, resulting in bad audio output.

- When the size is too big, the encoder will contain both the context of the speech and the characteristic of the speaker, resulting in the resulting audio containing the original speaker voice.

To prevent the bottleneck problem, we need to make sure that the bottleneck parameter is big enough to contain the context of the original audio and small enough to not contain the speaker characteristic of the original speaker.

The author's experiments show that the "small bottleneck" model contains the characteristic of the speaker from the reference audio but lacks context in the original audio while the "large bottleneck" model has the context fully but contains the voice of the original speaker. Therefore, it must have a bottleneck value so that the results contain both the reference speaker and the context from the original audio.

Between the encoder and decoder step, the author also performs the downsampling of the encoder output and then upsampling the output again the pass to the decoder. The author downsample the encoder output by the factor of 32, which means the only keep the 1st, 32st, 64st frame of the encoder output, and then upsampling the results again by multiplying each frame by 32. The aim of this is to generalize the training result, help the model gain better results.

AutoVC uses a reconstructed loss function to train the model. The reconstruction loss is applied to both the initial and final reconstruction results. As the author of the paper suggests, if we set the "bottleneck" parameter at a right amount, then the model will learn to extract the context feature without learning the characteristic of the original speaker.

4.1.2 DeepSpeech2:

Automatic Speech Recognition is one of the fields of Computational Linguistics that studies the translation of speech into text. Speech Recognition models also can be used as modules to extract content features from input speech audio. One of the most representative models is DeepSpeech2.

DeepSpeech2 is an ASR system that was introduced in 2015. It includes the Deep Neural Network (DNN) part and the Decoder/Inference part. The DNN part is used to generate feature vector which is probability of each character over time period and this part is trained using CTC loss function. The Decoder will generate the final text result through the Greedy or Beam Search algorithms. Due to the simplicity and efficiency of

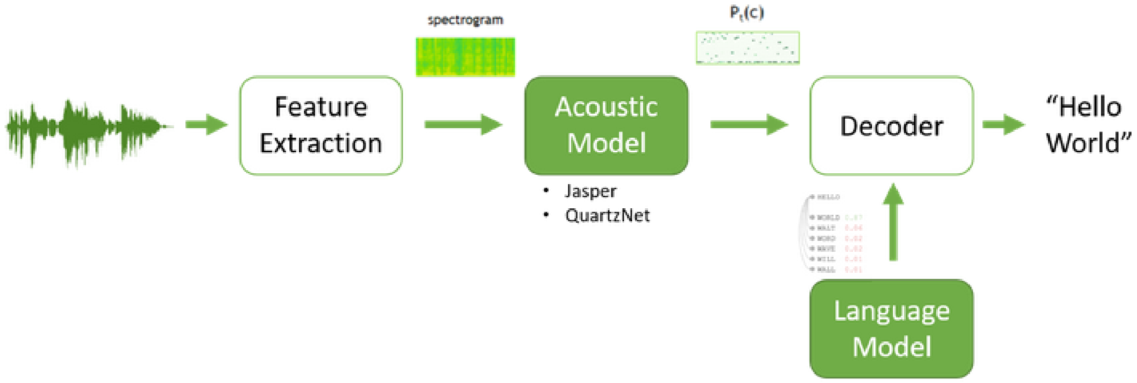


Figure 4.4: DeepSpeech2 pipeline

this method, we will integrate this model into our framework and evaluate the results.

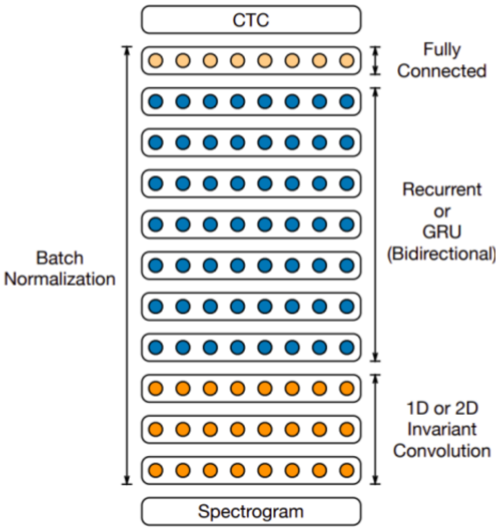

CTC
Fully Connected
Recurrent or GRU (Bidirectional)
Batch Normalization
1D or 2D Invariant Convolution
Spectrogram

Figure 4.5: DeepSpeech2 Architecture

We use the model configuration which is shown to be the best on LibriSpeech. It consists of:

- A convolutional layer with 32 channels, kernel size is (11, 41), stride size is (2, 2).

- A convolutional layer with 32 channels, kernel size is (11, 21), stride size is (1, 2).

- Five bidirectional GRU layers with the size of 800.

- One fully connected layer with the size of 1600.

- One projection layer with the size is the number of characters plus 1 for CTC blank symbol, which is 29 in English Voice Conversion, and 146 in Vietnamese Voice Conversion using phonemes.

4.1.3 DeepSpeech-based voice conversion:

One of the limitations of AutoVC is the training process requires configuring the parameter bottleneck to gain the best result possible. The parameter bottleneck is different depending on the overall quality and quantity of the dataset. If the bottleneck size is chosen correctly, then the training will be easier, otherwise, this may results in a long training time if the correct bottleneck value is not close enough. Although the result's quality is good enough, we want to find a way to train the model so that we don't need to configure the "bottleneck" parameter randomly at first and still generate meaningful results. Therefore, we propose a new method that can generate the result as good as AutoVC and does not require configuring the parameter bottleneck. This method is based on AutoVC and DeepSpeech2 models, which we call DeepSpeech-based Voice Conversion.

The DeepSpeech-based Voice Conversion method consists of four modules: a content encoder used to convert speech to text feature, a speaker encoder, a decoder to convert encoder output into text and a vocoder.

The content encoder will be based on DeepSpeech2, an ASR model used to convert speech into text. Normally, DeepSpeech2 consists of an Encoder module used to convert speech into character features and a Decoder module used to predict characters into text, as shown in figure Figure 4.4. Here in this method, we will train the DeepSpeech2 with CTC loss function and then only make use of the encoder module of DeepSpeech2 to extract content features from the original audio.

The decoder was the same as the decoder of the AutoVC model. The decoder will

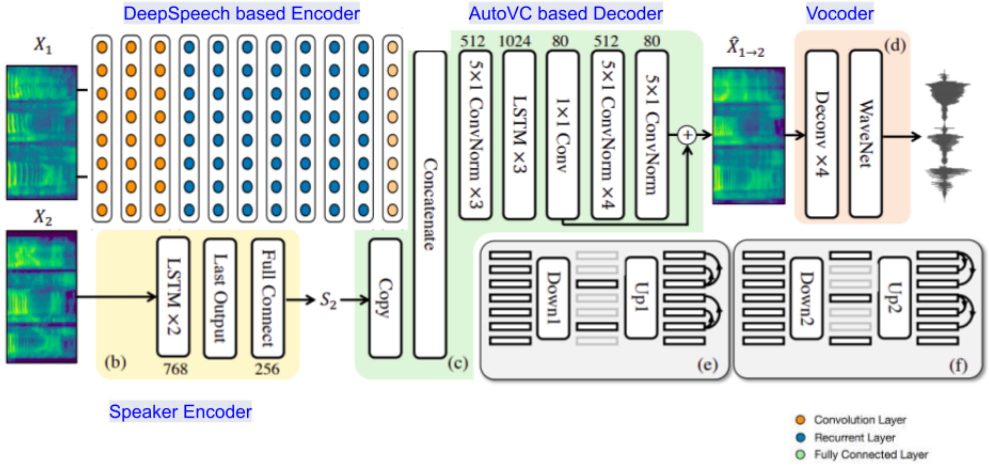


Figure 4.6: DeepSpeech-based voice conversion.

receive the output feature from the DeepSpeech module to generate the audio. The DeepVC decoder is inspired by AutoVC, but without the up-down sampling factor. The Vocoder module is also the same as the Vocoder module of AutoVC model.

In this method, we need to train three modules: the content encoder, the decoder and the speaker encoder separately. The training of the content encoder requires the multi-speaker datasets that contain text data. Luckily, most multi-speaker datasets contain text data. The size of the output depends on the number of characters or phonemes in the dictionary. In our experiment with the use of the character in the English dataset, the size of the output is 29. In our experiment with the use of phonemes in the Vietnamese dataset, the size of the output is 95, while the use of characters in the Vietnamese dataset requires the output's size of 146. The output dimension size, which is also the bottleneck size in the AutoVC model, is static and just needs to be configured one time depending on the dataset and the use of phonemes or characters.

To train the decoder module, we need the input is the content features extracted from the input mel-spectrogram using the encoder module. However, because the length of the content features are not the same as the length of the input mel-spectrograms, therefore, we need to recalculate the length of the ground truth mel-spectrogram to match the length of the content features.

Calculate for ground truth mel-spectrogram output length:

In DeepSpeech2, the length of the input change when passed through each Convolutional 2D layer and it is shown by this formula 4.7. Using this formula, we can calculate the length of the output content features, which is also the length of the ground truth mel-spectrogram.

$$w_{out} = \frac{w_{in} + 2 \times padding[1] - dilation[1] \times (kernel\_size[1] - 1) - 1}{stride[1]}$$

Figure 4.7: Conv2d output size calculation

If the input mel-spectrogram X1 has hop-length H1 with sequence length S1, then ground truth mel-spectrogram X2 with sequence length S2 needs to have hop-length H2. This is shown as formula:

$$\frac{H2}{H1} = \frac{S2}{S1}$$

We can calculate S2 following S1 by using the formula in figure Figure 4.7, therefore we can calculate hop-length H2 needed to generate ground-truth mel-spectrogram.

In our experiments, using the configuration as we mentioned above, we calculate the number of output frames is half of the initial mel-spectrogram frames. Therefore, to match the number of frames from the encoder output, we increase the hop length by 2 when converting the speech to mel-spectrogram. Then, we use the mel-spectrogram results to train the decoder module.

In theory, using this method, we can make sure that the content feature extracted from the content encoder does not contain the speaker feature of the original speaker, therefore removing the use of bottlenecks parameters. Also, we make use of the text provided by the dataset.

4.2 Speaker Encoder

This module is used to extract speaker voice characteristics features to pass to the Voice Conversion module for training.

There are two ways for represents the speaker characteristics:

- Embed Speaker Identification: Domain Dependency using One-hot vector

- Embed Speaker Representation: derive a high-level representation of a voice summary vector of 256 values - characteristics of voice speaker

We experiment with both representations. In the Embed Speaker Identification approach, we use a one-hot vector with the length of 65 corresponding with 65 speakers of vivos dataset. In the Embed Speaker Representation, we use the speaker verification model proposed with the use of GE2E loss function [10].

Our LSTM model consists of:

- 3 layers of LSTM with the hidden size of 768

- One projective layer to project the number of hidden size of output vector into d-vector (vector with size of 256).

4.3 Synthesizer

Text Normalization

Text Normalization is an important step in Text-to-Speech systems, helping to filter noise and making the input to be consistent with only Vietnamese syllables. The main task of the front-end of the TTS system is to standardize the text for the back-end sys-

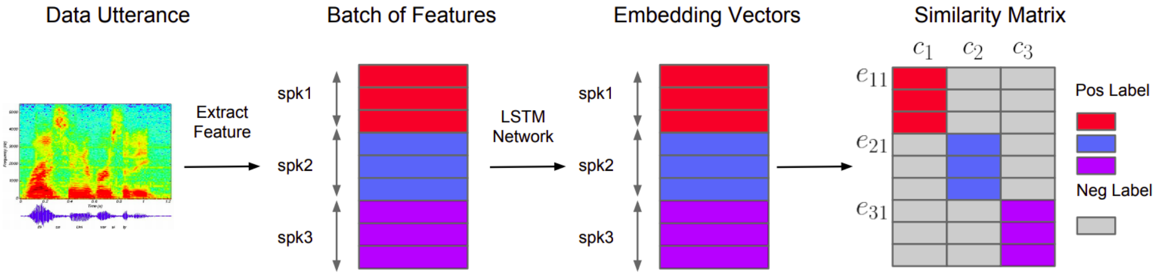


Figure 4.8: Speaker Verification with Generalized End to End Loss [10].

tem, the input is the raw text, we need to decide how to verbalize non-standard words, convert numbers, abbreviations, and words that cannot be pronounced into syllables, including dots, commas[65]. Every language needs different normalization processing methods because this problem is language-dependent[66]. It is impossible to build a complete text normalization because the language is ambiguous and evolves over time[67]. Text Normalization of Vietnamese Speech Synthesis today is still building grammars by hand instead of using automatic inference from large corpora because it has been the lack of annotated data[68]. To standardize text into readable words, the TTS system process through two steps, Rule-based and Dictionary-Checking. Specific methods for handling different cases of text normalization are shown in Appendix A. The text normalization process is illustrated in Figure 4.9

Phonetization

In order to apply these advanced models to Vietnamese, we need to standardize the data as well as propose using phonetic-based instead of character-based approach as an input of the neural network for taking the advantages of the Vietnamese language. In order to synthesize words that have never appeared in the train set or out of vocabulary words (OOV), we now use the grapheme instead of the character as the input for the end-to-end model. This makes the model converges faster. Because IPA is for describing sound, we not only create general lexicon for Vietnamese but it also depends on the speaker's own region in the train (dialect). The presentation of IPA for

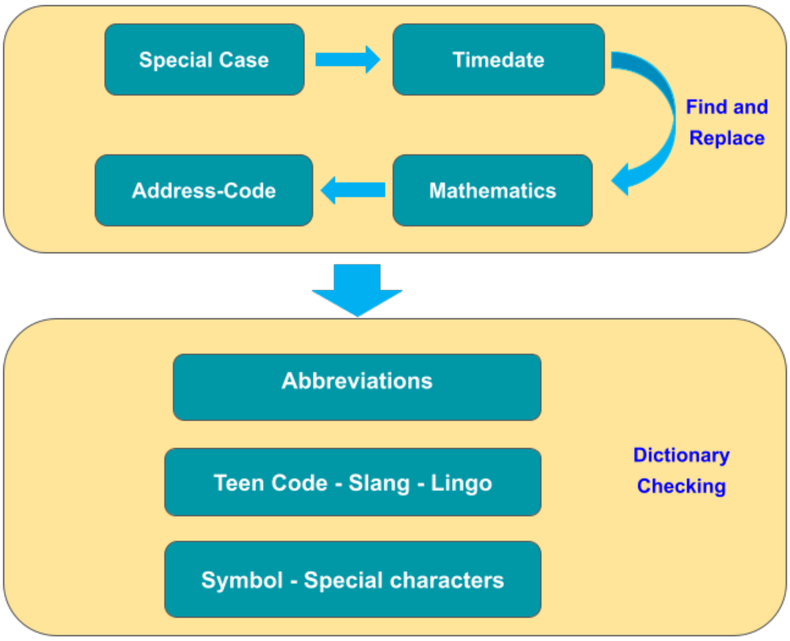


Figure 4.9: Pipeline for Text Normalization using Rule-based and Dictionary Checking.

Vietnamese has many ways and is still not unified. We refer to the method of Pham 2006 [69], customize the way some phonemes represent and some labiovelar on-glide. Because IPA does not display tones, we have signed the blanks, grave, acute, hook, tilde, dot accents with the numbers from 1 to 6. The output will be in the following format:

$$(C1)(w)V(G|C2) + T$$

There are a total of 144 IPA characters including tones, Vietnamese phoneme, English phoneme, dot, comma, and other special characters, each IPA character will be mapped corresponding to a number. Therefore the input text will be converted to a sequence of numbers and this sequence will be the input for the embedding layer. If using a dictionary with a large number of phonemes, the model will converge faster, the phoneme sets will be suitable if the training data set has few samples, we use many phonemes to control the model quality. However, in order for the generated voice to be diverse with many different contexts, we can reduce the number of phonemes so that the model can learn different readings of the phoneme, which requires a consistent voice dataset.

Mel-Generator

For the Mel-spectrogram Generator, we will apply Vietnamese phonemes to two main models, which are Tacotron2 and GlowTTS.

Tacotron2 for Vietnamese

The following are details of the tacotron2 model we used. To be able to apply to Vietnamese, in addition to the layers as described by the author, we replace the character embedding layer with the grapheme embedding layer. The tacotron2 model consists of three main parts encoder, attention, decoder.

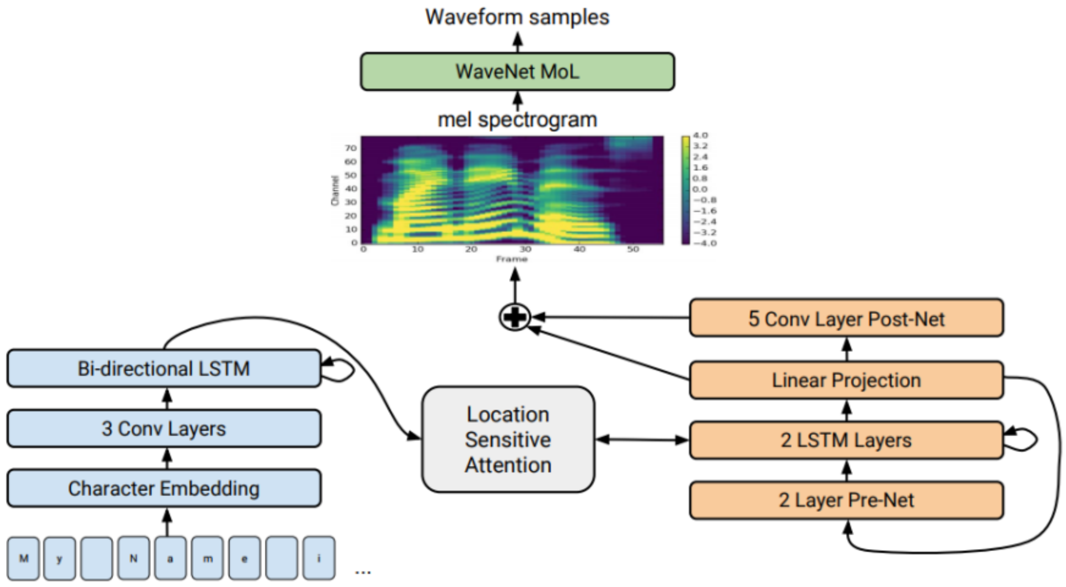


Figure 4.10: Tacotron2 Architecture.

The first part is an Encoder that converts the phonemic string into a word embed vector. These features are then used for the Decoder to predict the frames of the mel-spectrogram. The encoder includes the following neural networks:

- The network Grapheme Embedding uses to encode characters, the size of this network depends on the number of words that we configure in the dictionary.

- The 3 Conv networks after the output of the embed network will be put into 3 Convolution 1D layers and each of them contains 512 5x1 filters and finally the Batch Normalization layer and the ReLU activation function.

- The output of the final convolutional layer is fed into a bidirectional LSTM network containing 512 units (256 units in each direction) to generate encoded features.

In the article, the author proposes a new Attention mechanism for the speech recognition problem which extends the additive attention mechanism as we mentioned in the preliminaries chapter. The purpose of the Attention layer is to help the model focus on not only the features in the previous steps but also the features at the current position. The Local Sensitive Attention mechanism proved to be very effective in distinguishing between phonemes with similar pronunciations such as /kcl/ and /k/ in English. This will help a lot in clearly distinguishing the tones with additional tones in Vietnamese. We can see this clearly when we compare the voices generated by GlowTTS in our demo.

The purpose of the decoder network is to generate a mel-spectrogram from the output of the previous step. We first consider the pre-net with two fully connected layers of 256 units and the ReLU activation function. The output of the pre-net is concatenated with the output of the attention network and passed through two layers of LSTM with 1024 units. Finally, to predict the mel-spectrogram, the output vector is passed through 5 convolution layers called post-net.

GlowTTS for Vietnamese

The GlowTTS model consists of three main parts: encoder, decoder and duration predictor.

The GlowTTS encoder is the same as the encoder structure of Transformer TTS, however in the self attention section, the GlowTTS model will replace the positional encoding with relative position representations. A residual connection is also added to the pre-net encoder. A linear projection layer at the end of the encoder to estimate the statistics of the prior distribution. When comparing the encoder of GlowTTS and TransformerTTS with the encoder of Tacotron2, the bi-directional RNN was replaced with Transformer encoder. The advantages of attention in transformers over seq-to-seq networks have been shown in the preliminaries chapter. Multi-head attention will include the attention generated from the initial input, in order to help the model learn many different aspects of the data randomly, this helps to increase the number of features and improve the training time. In addition, the frames of the melspectrogram will have long-time dependency information, from which context-dependent features will be learned more efficiently.

Obviously, unlike Tacotron2, GlowTTS allows parallel computations, thereby improving training time. The duration predictor consists of convolutional layers, like other CNN models, the predictor also uses the activation function ReLU and normalization layers. The architecture of predictor GlowTTS is exactly the same as the one of FastSpeech. Therefore, the loss function of the GlowTTS model is equal to the sum of the maximum likelihood estimation loss on the frames of the mel-spectrogram and the duration loss which is an MSE loss between Predicted and extracted duration from the training audio.

The most important part of GlowTTS is the flow-based decoder. At the training step, the mel-spectrograms are transformed into the latent representation, then these inputs will be used for maximum likelihood estimation and internal alignment search. The decoder will include main layers like the invertible 1x1 convolution layer, activation normalization layer, and affine coupling layer.

Because the Monotonic Alignment Search algorithm cannot run in parallel on the

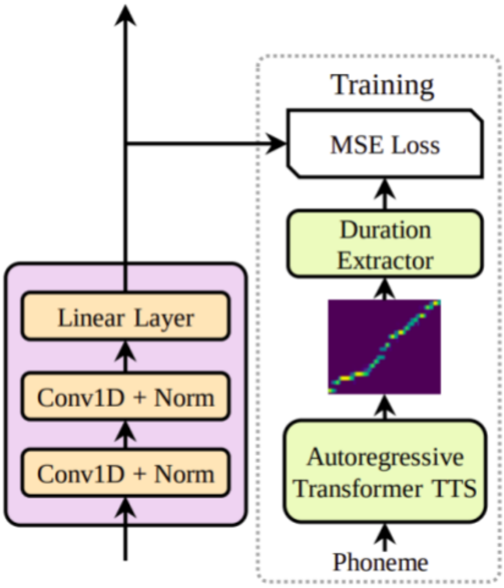


Figure 4.11: Duration Predictor of Fastspeech and GlowTTS.

GPU, so during the experiment, we found that if we set the batch size too large, it slows down the training process due to the delay from the CPU executions. During inference, we do not need MAS as the duration predictor will estimate the alignment.

4.4 Vocoder

Vocoder models help to generate voice waveforms from acoustic features or mel-spectrograms. Besides Griffin-Lim Algorithm, we study on different models such as Waveglow, Melgan and especially the HifiGAN model that can create fast, quality voice output. Like other GAN-based models, HifiGAN consists of a generator and discriminators that are trained adversarially.

- The generator is a fully convolutional neural network. Mel-spectrogram will be up-sampled through transposed convolutions until matches the temporal resolution of raw waveforms. Next are the multi-receptive field fusion (MRF) modules to observe patterns of various lengths in parallel. MRF consists of multiple residual blocks, each using different kernel sizes and dilations. Each phoneme can correspond to 2000 samples in the speech waveform, so the convolutional layers must be capable of long-range memorization.

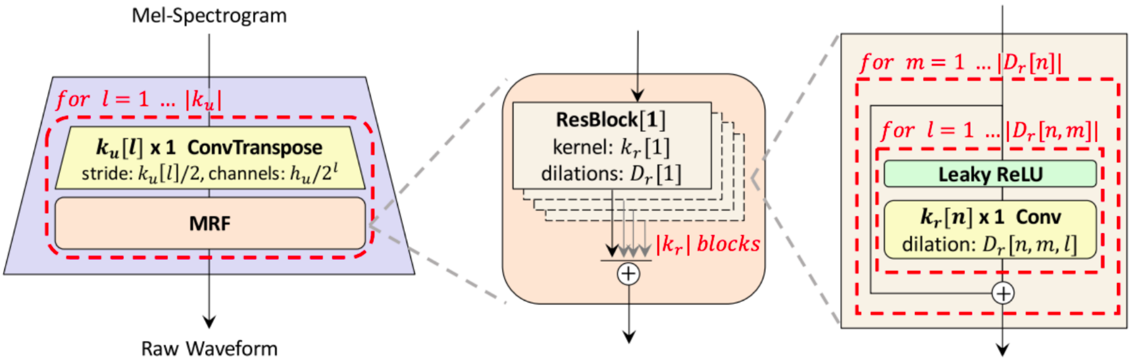

Mel-Spectrogram
$for\ \ l = 1\ ...\ |k_u|$
$k_u[l]$ x 1 ConvTranspose
stride: $k_u[l]/2$, channels: $h_u/2^l$
MRF
Raw Waveform
ResBlock[1]
kernel: $k_r[1]$
dilations: $D_r[1]$
$|k_r|\ blocks$
$for\ \ m = 1\ ...\ |D_r[n]|$
$for\ \ l = 1\ ...\ |D_r[n,m]|$
Leaky ReLU
$k_r[n]$ x 1 Conv
dilation: $D_r[n,m,l]$

Figure 4.12: Generator Architecture of HifiGAN [11].

- Discriminator of HifiGAN includes 2 sub-modules, Multi-Period Discriminator to handle portions of periodic signals of input audio and Multi-Scale Discriminator to capture consecutive patterns and long-term dependencies. The MPD will consist of several sub-discriminators, each of which is a stack of stridden convolutional layers with leaky rectified linear unit, tasked with looking at different parts of input audio. Unlike MPD which learns on disjoint samples, MSD works on smoothed waveforms using average pooling. Both discriminators are responsible for determining whether the input waveform is real or fake. Hifigan uses 3 types of loss functions for training: GAN loss, Mel-Spectrogram Loss, Feature Matching Loss to improve stability and quality of the output voice.

4.5 Voice Cloning System

After evaluating different methods in each module, we selected the best neural network models and proposals of each module, forming a pipeline based on the voice cloning framework that is based on voice conversion as described in the introduction chapter.

For the speech synthesizer, we choose Tacotron2 model along with vinorm normalizer and phoneme conversion library - viphoneme. Although the voice generated by the autoregressive Tacotron2 model has a natural quality and is more like a human voice, the inference time is quite long when compared to glowTTS. To optimize the runtime and not depend on the deep learning framework, the Tacotron2 model is converted to ONNX, an Open Neural Network Exchange that establishes standards for representing machine learning algorithms. With this format, we can easily convert the model to TFlite which supports running artificial intelligence models on web platforms and mobile devices.

For the speaker encoder module, we use the trained speaker vector extractor to adapt on the Vietnamese voice dataset. Compared with using the checkpoint pretrained of the speaker verification model used for English, after the adaptation process, the characteristics of the speakers in the training domain as well as the Vietnamese speakers are more clearly distinguished, especially the characteristics of gender and pitch of the speaker.

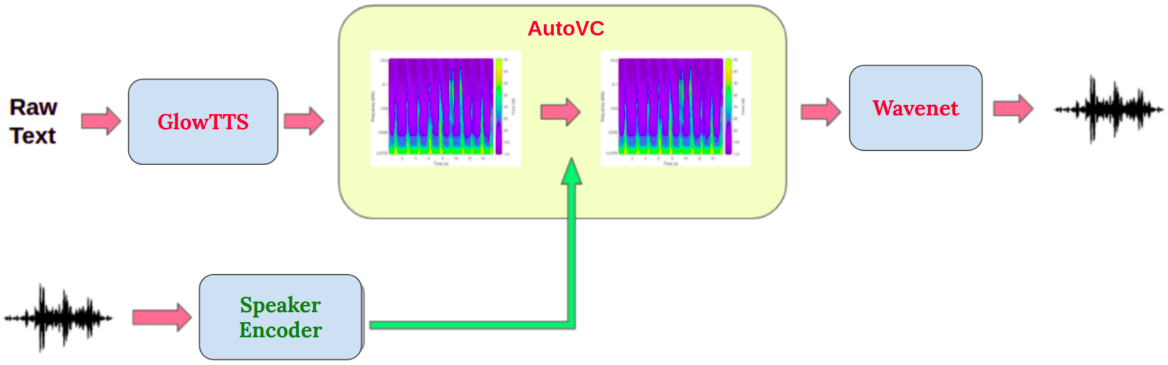


Figure 4.13: Voice cloning based on Voice Conversion with AutoVC.

With the vocoder module, we consider choosing between WaveNet and HifiGan models. Although HifiGan's inference time is very fast and can be applied to real-time running because it is non-autoregressive, the quality of the generated waveform cannot be better than that of WaveNet. Furthermore, if we look at the pipeline of the voice cloning framework, the vocoder is placed after the voice conversion module, the mel-spectrogram generated by the voice conversion model has been transformed a lot compared to the mel-spectrogram generated by the synthesizer, the quality of the input for vocoder will no longer be preserved, we need a stable and efficient vocoder model, so WaveNet model is chosen for quality cloning speech, HifiGan is suitable for speech synthesis and real-time demos.

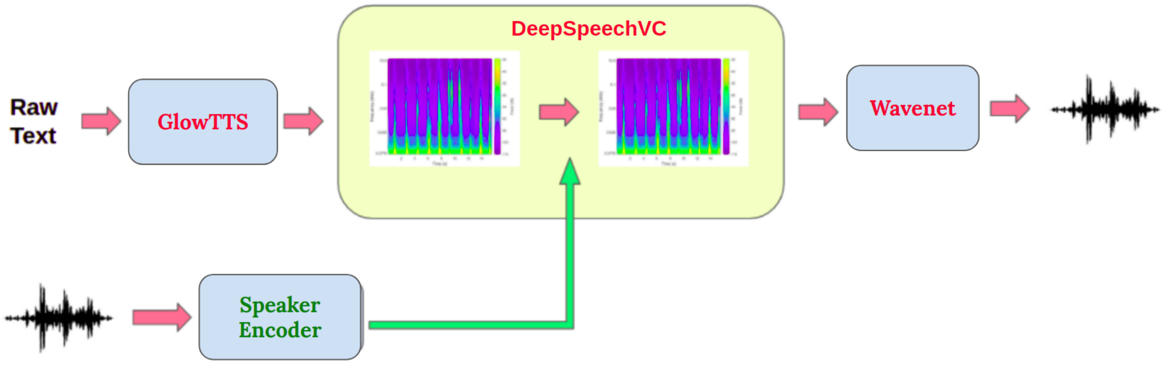


Figure 4.14: Voice cloning based on Voice Conversion with DeepSpeechVC.

The most important module for the voice cloning framework is the voice conversion, which plays the role of converting the input voice based on the speaker information taken from the speaker encoder. We have proposed a new voice conversion model based on deepspeech2, which is used in ASR problems, to improve the limitations of AutoVC. We attached two VC models including our solution and the AutoVC model to the voice cloning framework, and then we evaluated the entire system.

CHAPTER 5

EXPERIMENTS AND RESULTS

To be able to compare and evaluate the quality of the solutions that we have proposed in chapters 4 and 5, we select suitable datasets for each problem, perform training and fine tuning for the models. Then we use the appropriate metrics, give analysis and evaluation, trade-off considerations to choose the best model to build an end to end framework.

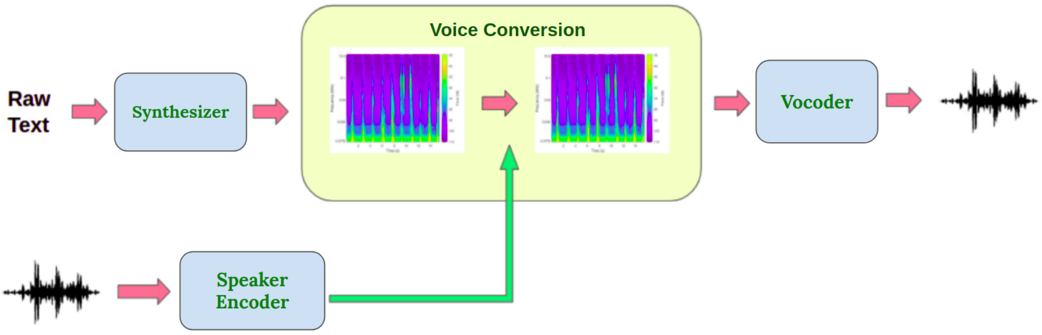


Figure 5.1: An overview of voice cloning framework based on speech synthesis and voice conversion.

5.1 Dataset

Each module of voice cloning will have different requirements on the dataset, such as the requirement for the number of speakers, the quality of the audio, and whether the language is Vietnamese or English. Before putting data into training, we need to perform preprocessing to reduce complexity and ambiguity, so that we can help the model learn effectively and improve the training process.

5.1.1 Resource

The studies of this thesis include many different artificial neural model modules, so many different datasets have been used for each specific problem.

For the speaker encoder module, we use multi-speaker data that was published in the ZaloAI challenge. The dataset includes 400 speakers with an average of 26 audios per speaker, which we use to train and test the speaker verification model for Vietnamese. In addition, we also use ASR data VinBigdata including 100h of data, we use this dataset to try the method of clustering speakers, serving the training voice conversion process.

Speech synthesizer and vocoder modules require quality data to synthesize human-like voices, we use two datasets with single speakers from VinAI and the dataset is provided by InfoRe Jsc, which is also the Big Corpus set in International Workshop on Vietnamese Language and Speech Processing(VLSP) 2019 [17]. The dataset included about 22 hours with 13,462 utterances of north-accent female Vietnamese. Because the data set contains lots of noisy audio, we filtered out and removed more than 2000 samples, many of the samples that the reader stopped in the wrong place also affect the training process.

For the voice conversion problem, because we have proposed a new speech2speech solution, to re-evaluate the quality of our solution compared to VC models, in the world, we experiment on VCTK, an English multispeaker dataset. More specifically, we use Device Recorded VCTK, a small subset extracted from the VCTK dataset. This small subset contains 30 speakers and is used to train the voice conversion model based on deepspeech2.

After evaluating the model in English, we adapted the language then training AutoVC and DeepSpeechVC Voice Conversion Module to Vietnamese using AILAB's current high-quality, multi-speaker dataset, the VIVOS dataset [70]. VIVOS contains 65 speakers and almost 16 hours.

Table 5.1: List of datasets and the modules in which they are used for training and testing.

Dataset	Module
ZaloAI	Training for Speaker Encoder module
VinBigData ASR 100h	Experiment for Speaker Encoder module
VLSP2019 TTS	Training for Speech Synthesis module
VinAI TTS	Training for Speech Synthesis module
VIVOS-AILAB	Training AutoVC and DeepSpeechVC modules Testing for Speaker Encoder module
VCTK	Training for DeepSpeechVC in English

5.1.2 Audio Pre-processing

Since the collected audio data is not completely accurate, we need to proceed to filter out the non-standard audio. The process of normalizing training data and evaluation criteria will be presented in turn according to the following steps:

Transcript and audio must match

We use a 3rd party Speech2Text like Google and run through all the audio files and generate the transcript. If the generated and given transcripts differ by more than a threshold, the sentence should be discarded.

Reading speed and standard deviation

Normally, a good reading voice will keep a stable reading rate. That will be measured by the standard deviation of the time it takes to read the syllable or word. Speaker rate can be measured in syllables/seconds. We will calculate the time of each syllable/seconds. If it is greater than a threshold, it should be prompted to remove. It is possible to calculate the average speed for sentences, with sentences with a slow rate below a certain threshold also consider removing.

The standard deviation of F0

With the voices for Speech Synthesis, there is no need to press too hard, it will create outliers during training. It can also be caused by the inaccuracy of the extract tool F0. So after extracting F0, calculate the standard deviation F0 of each sentence, then plot the whole dataset and check if any sentences have abnormally high Std(F0). If so, it should be removed from the dataset.

Speaker fluency

Usually if in a sentence, the fluent reader will not stop in the middle of the sentence. Readers who do not read a sentence first will easily be interrupted in the middle of the sentence because of the surprise about the content of the sentence. To assess the fluency of the reader we will rely on the ratio: The maximum silence in the middle of sentences (excluding beginning and end) divided by the average length of reading time of syllable. If this ratio is high, it means that the user is resting too much in the sentence.

To be able to generate speech as closely as possible to human voices and match Vocoder's input, we reduce the sampling rate of each input audios data from 44100 Hz to 22050 Hz by using FFmpeg library to ensure the audio sample rate changes but

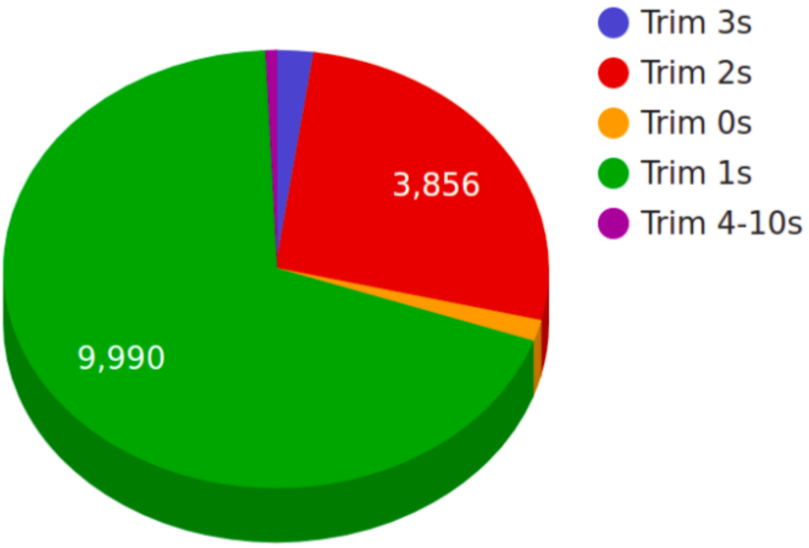


Figure 5.2: Silent trimming for training set.

keeps the speech rate unchanged.

We remove silence at the start and the end, then adding one second silence to the end of each audio in order to help the model recognize the end of the sentence better. We trim both sides of audios which have silent < 20dB. The cut length of each audio is shown in the figure 5.2.

Audios that are too short often cause a difference in padding length, reducing the efficiency of the training process. We choose 2% of the audios that are too short in the training set, then merge them to form a longer audio, the text of the two corresponding audios is also combined by commas. We've made statistics that audios longer than 3.21 seconds will be preserved as shown in the figure 5.5.

5.2 Evaluation results

The detailed evaluation results of the models are clearly listed in this section. At the same time we also analyze the pros and cons of the models.

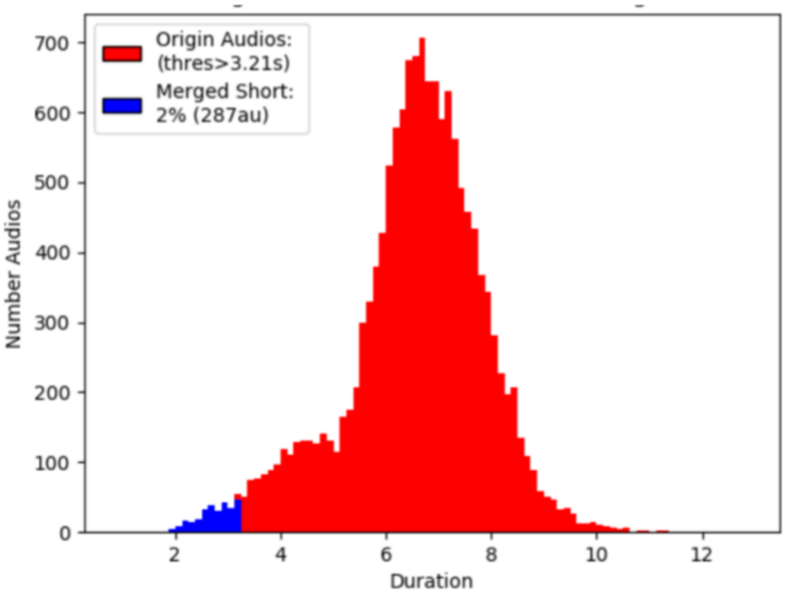


Figure 5.3: Histogram of the duration of audio in training set.

5.2.1 Voice Conversion

In AutoVC approach, we train the autovc on vivos dataset with the batch size of 2 and the learning rate of 0.0001. The training time is 48 hours and the model takes about 31 hours to converge. However, we also must fine-tune the bottleneck value two times, from 16 to 32, to get the best results possible, therefore, increasing the total training time to 96 hours.

In DeepSpeech2-based voice conversion approach, for Vietnamese voice conversion, we use the Vivos dataset to train both the encoder module and the decoder module, while for the English voice conversion, we only use the VCTK dataset to train the decoder module. The training on the encoder module took 48 hours to finish while the training on the decoder module only took 12 hours to converge, make the total training time 60 hours.

Since VIVOS dataset does not provide parallel audio samples, in Vietnamese voice conversion, we will evaluate this module as a part of the total system. The figure 5.3

is our evaluation on the VCTK dataset and using the MCD metric:

Table 5.2: Compare MCD result between AutoVC and DeepSpeech-based voice conversion on VCTK dataset.

	AutoVC	DeepSpeech2-based VC
MCD	16.387	14.755
Training time (hours)	96	60
Number of parameters	28M	111M
Average Runtime	0.056s	0.021s

The evaluation shows that the DeepSpeech-based Voice Conversion model performs better than AutoVC model in both quality and training time. When comparing the two models with each other and the evaluation results, we can see that the biggest difference between the proposed model and AutoVC is the Content Encoder.

DeepSpeech-based Voice Conversion solves the bottleneck problem as mentioned in the thesis, we no longer spend too much time tuning. This helps the training process become easier since we only need to train the DeepSpeech2 module and the Decoder module one time. With the AutoVC, we may need to train the model many times to find the suitable bottleneck with the best result. With our solution, Content Encoder, Speaker Encoder, and Decoder can be trained independently.

Finally and most importantly, DeepSpeech2 is able to generate the result mel-spectrogram better than the AutoVC model because it is able to extract the content better than the AutoVC encoder module since it is the ARS model, which generates the content feature exactly (frames of the phoneme probability). For AutoVC, it must control the features generated by the content encoder by a loss function with reconstructed mel-

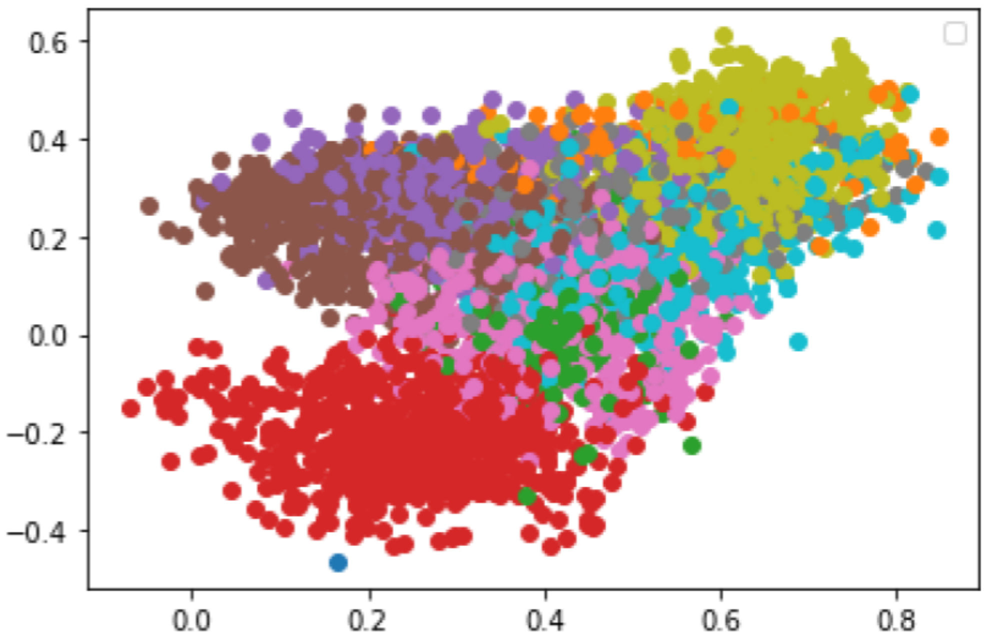


Figure 5.4: Clustering for D-Vector of Speakers with PCA algorithmn.

spectrogram. The backpropagation process takes place in parallel with the training of both encoder and decoder.

5.2.2 Speaker Encoder

We trained the model on the training dataset provided by ZaloAI. We evaluate the result by using two metrics, accuracy and EER, on the test set of VIVOS dataset.

The ZaloAI dataset contains a total of 400 speakers with average of 26 audios per speaker.

- In the accuracy evaluation approach, we generate 1000 pairs of speaker speech, then into each pair, we use our method to generate the d-vector of each audio and calculate the cousin similarity between them, if the cousin similarity is bigger than the threshold (default is 0.5) then the result according the method is that two speech belong to the same speaker, otherwise they do not belong to the same speaker. Compare that to the actual result (ground truth) and then average all the results, we gain an accuracy of

89.8%, which is pretty impressive.

- With the Equal Error Rate measure, we evaluate the model ten times by inference the validation set over 10 epochs, then take the average EER result. The evaluation result on the English dataset VCTK is 0.0355. For Vietnamese, after the process of adapting to Vietnamese on the ZaloAI Challenge set, we conducted an evaluation on the VIVOS validation set, the EER result was 0.0747. In other words, the model can give an accurate result of 92.53%, from which the speakers will have more easily distinguishable features.

VCTK	ZaloAI
0.0355	0.0747

Figure 5.5: Compare EER result between Speaker Encoder trained on VCTK dataset (English) and ZaloAI dataset (Vietnamese).

We also evaluate the results on the VinBigData dataset. As we mentioned above, this dataset contains a total of 100 hours of speech, each sample has no label to indicate the speaker. We use our method as a speaker encoder model to extract the d-vector feature from each sample. Then we cluster these d-vector features using a k-means clustering algorithm with many different k values. Figure 5.4 shows our clustering result, reducing to the two most important dimensions using PCA algorithms.

The overall quality of the dataset is average, but good enough for training our model. This speaker verification model is essential to extract speaker characteristic feature from audio and allow zero-shot voice cloning.

5.2.3 Speech Synthesis

Vinorm

We provide a python package on Ubuntu 18.04 that can be installed at the Python Package Index called ViNorm.

From the data collected, we extracted 100 tricky need-normalized cases[1] to use as the baseline for improvement, 500 random cases in practical contexts for testing our proposal. These test cases do not include normal sentences, foreign words and proper nouns. With the 500 test cases, we improved the frontend of VOS from 60% to 97%.

Method	Accuracy
Front-end VOS 2.0	60%
Updated Front-end VOS	97%

Tacotron2

For the Tacotron2 model, we train the model with two Nvidia Tesla V100 and use a batch size of 32, with a learning rate is 10^{-5}. We run the model with 200 epochs and it converges after the 82000^{th} iteration. The total amount of training is about 120 hours, approximately 5 days. Some audio parameters for training models such as filter length are 1024, hop length is 256, window length is 1024, the number of mel channels is 80.

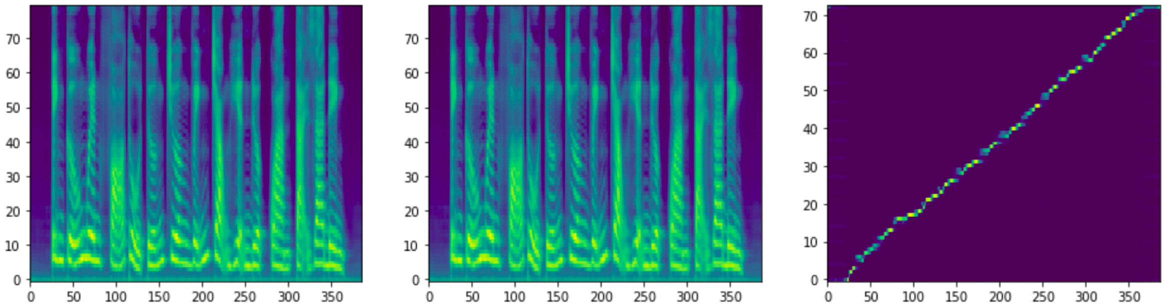


Figure 5.6: Melspectrogram and Alignment Generated by Tacotron2.

Figure 5.6 is the output generated during the inference process. The first two images are the mel-spectrogram generated by the model and the one generated by the

[1] https://github.com/NoahDrisort/NICS_Appendix

speaker's voice, respectively.

To evaluate the Tacotron2 model when applied to Vietnamese, without being dependent on the WaveGlow vocoder, ground truth audios for testing are converted into mel-spectrograms and then converted these mel-spectrograms back to audios by using pre-trained WaveGlow as shown in 5.7. This processed ground truth is called Groundtruth (Mel + WaveGlow), they will be compared with voices synthesized and standardized by our model.

Model	MOS
Tacotron2 (WaveGlow)	3.97
Groundtruth (Mel + WaveGlow)	4.43

To evaluate the result, we choose the MOS (mean opinion score) scoring system on the test set to check the quality of audios[71]. Each person who joins the survey listens to 40 audios, which include 20 audios generated from Tacotron2 and 20 corresponding ground truth audio generate from the process. They were asked to grade from 5 to 1, based on how natural and smooth those speeches compare to real human speeches. The final score of each audio type will be equal to the total score divides by the number of survey participants. Below is the MOS result gain from the survey of at least 20 people.

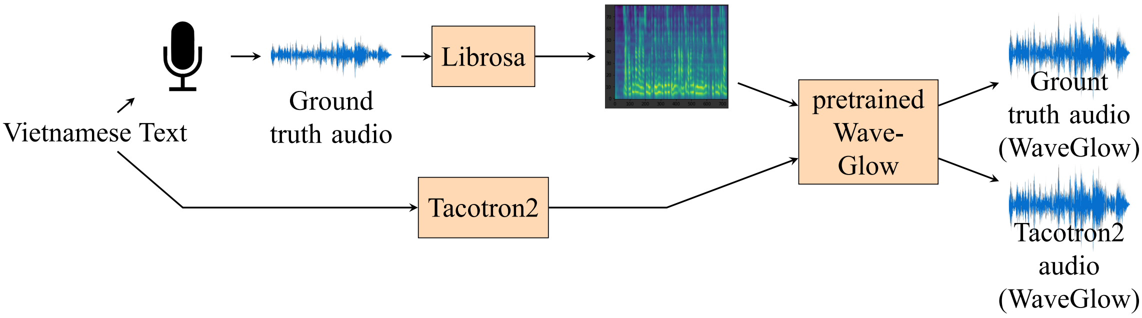


Figure 5.7: Ground truth (WaveGlow) generating process.

GlowTTS

We trained the glowTTS model with RTX 2080 and the model converged after about 3 days with 1200 epoch. The loss function of the training process is illustrated in the figure 5.8, We can see that the model is overfitting when looking at the total loss, however, the total loss is affected by the duration loss because the duration predictor model converges early because it's pretty simple. Therefore we only need to care about the mel-spectrogram loss.

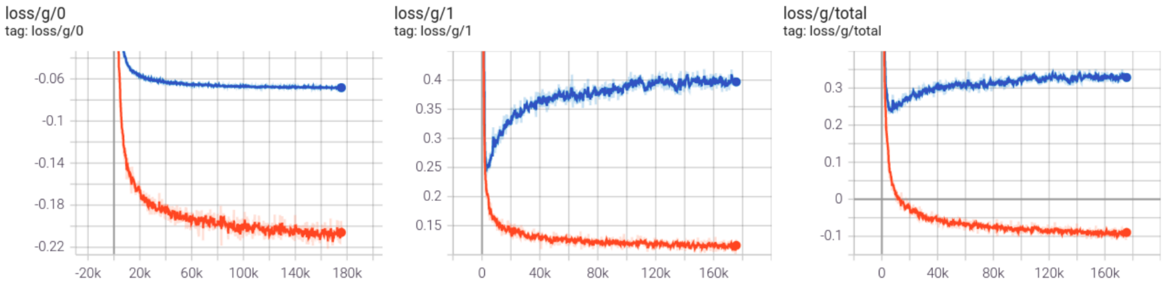
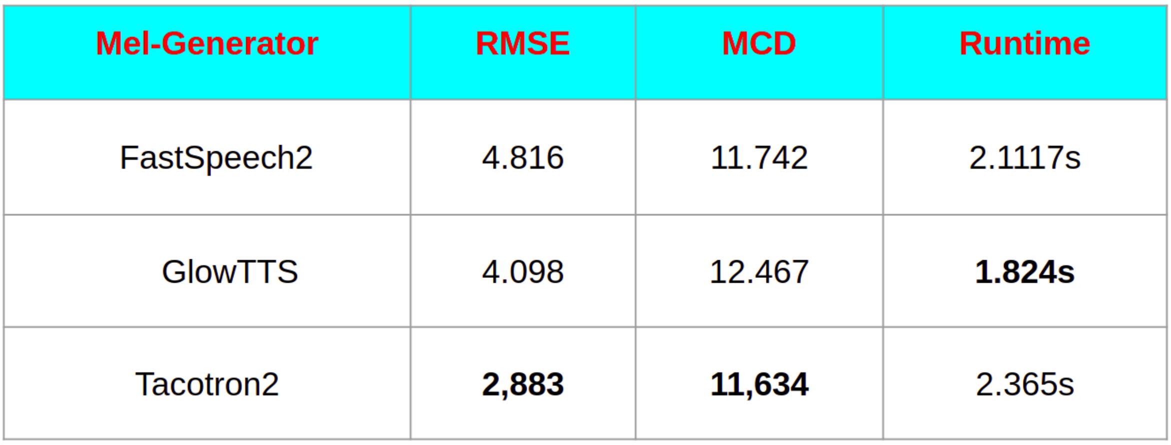


Figure 5.8: Training Loss of GlowTTS for Vietnamese.

In addition, we also tested on other mel-generator models such as Fastspeech2, MelGan. We have trained these models on Vietnamese dataset. The fastspeech2 model is trained with 200k steps, using many different loss functions such as F0 loss, duration loss, energy loss. Since it is a non-autoregressive model, fastspeech2 needs to depend on the training results and the quality of Tacotron2, however, the inference time will be improved compared to Tacotron2.

We evaluate with objective metrics including RMSE, MCD and also compare the inference time of each model. The time to generate the mel-spectrogram is averaged over 100 samples, which does not include time for initializing and writing audio. The inference process is performed on the Tesla V100 GPU.

The results of the evaluation of the mel-generator models are shown in the table 5.9. We can see that the voice audio quality produced by the Tacotron2 model is better than the rest, with the lowest RMSE and MCD. However, GlowTTS's inference time is the fastest as it takes less than two seconds to generate the mel-spectrogram. The reason is that the Tacotron2 model is an autoregressive model, so the quality is better than GlowTTS, but the inference time of GlowTTS will be faster because it is a non-autoregressive model.



Mel-Generator	RMSE	MCD	Runtime
FastSpeech2	4.816	11.742	2.1117s
GlowTTS	4.098	12.467	**1.824s**
Tacotron2	**2,883**	**11,634**	2.365s

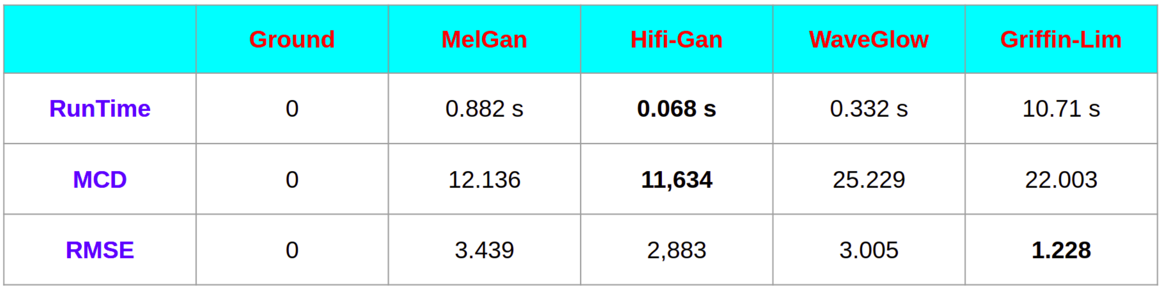
Figure 5.9: Evaluation on Synthesizers for Vietnamese.

5.2.4 Vocoder

The WaveGlow model's English pretrained is available from NVIDIA. This model was trained on a studio-quality single female speaker dataset with about 20 hours of speech. They trained the model for 1.5M iterations, it takes about 1 day to train 30k iterations, so we need about 50 days to train this vocoder. To train this vocoder model, we need the input mel-spectrogram and the output waveform to be generated accordingly, so the model is text-independent. Due to time constraints and GPU resources, we use English pretrained, then train for Vietnamese data instead of training from scratch. It took about 78k iterations to train the Vietnamese data.

To evaluate and compare between different vocoders, we use the Tacotron2 model to generate the mel-spectrogram, the vocoders will use this mel-spectrogram to generate the voice waveform. The generated waveform is converted back to the mel-spectrogram for evaluation by the MCD measure.

From the comparison table between vocoder models 5.10, we can see that HifiGAN excels in both inference time and voice quality in MCD measure. The results are calculated and averaged on a test set of 100 samples.



	Ground	MelGan	Hifi-Gan	WaveGlow	Griffin-Lim
RunTime	0	0.882 s	**0.068 s**	0.332 s	10.71 s
MCD	0	12.136	**11,634**	25.229	22.003
RMSE	0	3.439	2,883	3.005	**1.228**

Figure 5.10: Evaluation on Vocoders for Vietnamese.

5.2.5 Voice Cloning Framework

After creating a framework that combines voice conversion with supporting modules such as speaker encoder, synthesizer and vocoder, we performed an inference process to evaluate the two voice cloning systems. The first one has a voice conversion part which is AutoVC trained in Vietnamese, the second one is a voice cloning system with a voice conversion part which is our proposed model.

We choose a Vietnamese multi-speaker dataset of AILAB, VIVOS, with clear and quality voices. We choose the objective measure, mel cepstral distortion, to compare the cloned voice with the real voice.

For each voice cloning system, we conduct an evaluation on a training set with known in-domain speakers, and a test set with unknown speakers. For the training set, we selected 46 pair-samples from 46 speakers, similar to the testing set, we selected 19 pair-samples corresponding to 19 speakers.

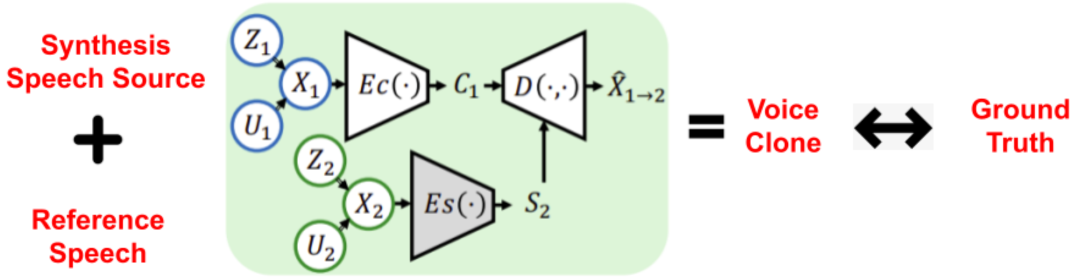


Figure 5.11: Evaluation Process for Voice Cloning framework.

- Each pair-sample mentioned above is used for one voice cloning turn. A pair-sample consists of 2 audios of the same speaker U2, that is (Z1 - U2) and (Z2 - U2)

- The first audio sample (Z1 - U2) is used as the ground truth, which the voice cloning framework needs to generate.

- The second audio sample (Z2 - U2) is used as an audio reference. The speaker encoder module is responsible for extracting information from this audio to create S2 features.

- The speak synthesis model will generate audio (Z1 - U1) called origin audio, containing the content that the cloned voice needs to say.

- The Voice Conversion module will extract the content feature C1 from the origin audio (Z1 - U1), from the two features S2 and C1 obtained, the decoder part of the voice conversion model will generate a mel-spectrogram for the new voice.

- Vocoder will synthesize and convert cloning mel-spectrogram into waveform form cloning voice (Z1 - U2)

- We compare this cloned voice with the first audio ground truth of the pair-sample mentioned above. Measure the distance of two audio using MCD.

This process is repeated for each pair-sample of each speaker. The average MCD results of two voice cloning systems based on speech synthesis and voice conversion, AutoVC and DeepSpeechVC are shown in the table below.

Table 5.3: Compare MCD result between AutoVC-based VC System and DeepSpeechVC System on VCTK dataset.

	AutoVC system	DeepSpeechVC system
MCD	13.675	11.900

The speech synthesizer, vocoder and speaker encoder modules are all preferred to use models that produce good quality instead of optimizing runtime. Our proposed Voice Conversion method performs better than the current AutoVC method by using DeepSpeech2 as a content encoder, resulting in improved overall results of the Voice Cloning framework.

CHAPTER 6

CONCLUSION

In this chapter, we highlight the work involved in building a voice cloning based on speech synthesis and voice conversion. Then we summarize what knowledge we have research, experiments, evaluation results on models and solutions that we propose. Ethical issues of science and potential future research problems are also discussed.

6.1 Results

In this thesis, we have researched and proposed solutions to build a pipeline framework for Vietnamese voice cloning problem based on speech synthesis and voice conversion. We break down the pipeline into small problem modules, conduct experiments on state-of-the-art models, suggest improvements and applications for Vietnamese.

Our main contribution can be mentioned as one of the first studies on Voice Cloning for Vietnamese which has the ability to synthesize speech with only an input audio sample. We have proposed a new Voice Conversion model based on AutoVC and ARS model - DeepSpeechVC, then conducted an evaluation to compare with international models in English. At the same time, for the Speech Synthesis module, we experimented on SOTA models of the E2E Speech Synthesis problem for Vietnamese, the models have been compressed to be able to run on Android devices without internet and GPU, the output has a natural and fluent voice, quite similar to the real voice used for training with a MOS score of 3.97. We have provided a set of libraries for text normalization - Text Normalization on cross-platform (C++, Python, Android NDK) and a Grapheme to IPA conversion library, which helps the speech synthesis process to converge faster when performing Vietnamese phonetics as well as handle cross-language cases. These libraries have direct practical applications in industry, solving the most ambiguous problems encountered in real life.

The objectives stated at the beginning of the thesis have been implemented and achieved good results when the proposed models have higher results than the baseline model. A part of this thesis has been selected and further developed into scientific articles. Several potential studies are being evaluated and compared in more detail. We also provide a demo website including a TTS real-time demo along with samples of synthesized voices based on Voice Cloning.

6.2 Ethic

AI Technology is growing rapidly, which has also raised many concerns about the danger of the development itself. In the past, there were also many synthesized voice systems doing voice cloning, but the most notable here is the ”one-minute” number, collecting a person’s voice in a minute is a lot easier than collecting an hour’s data set audio. This raises important questions if the system can become a tool for bad guys, can be used for tricking the verified identity of software, and bring more unhappiness than happiness. Despite that problem, we believe this technology can be used for creativity and entertainment, and make human life more colorful. We can publicize the technology so that everyone will soon be aware that such technology exists. In that way, the damage will be a lesson, we agree with such a solution. In our opinion, this technology should be public and should be developed more to make it a safe tool for everyone to use.

6.3 Future works

With the development of deep learning, modern speech models can produce quality, natural voices like humans, but there are still many challenges when we apply these technologies in practice.

Training and evaluation of the models were performed with ideal text and audio. The audio is recorded in studios with quality microphones, when putting speech problems in a noisy environment, using the microphone of phone or computer devices, the quality will be reduced, even the domain voice is different. completely.

Similarly, for text inputs, the syllables used for training are completely normalized, including only words that appear in the Vietnamese dictionary, not numbers. However, for documents that appear on the Internet, there will be many cases of abbreviations, even newly appearing, the model will not be updated in time. We need a frontend module to standardize these cases. The methods used so far are to use the rule-based method used in this thesis to handle each case individually. The rule-based approach requires ongoing maintenance as the language evolves. There are many methods of normalizing text input using neural networks such as a sequence to sequence model [72], or using pre-trained for text representations such as BERT [73] [74], and the most noticeable approach is the hybrid model, which combines rule-based and attention-based models [75].

The second problem we need to deal with is that in documents containing many cross languages, we can use a transliteration dictionary to map words into Vietnamese. Another solution is to convert the phonemes of other languages to the phonemes of Vietnamese, but this makes pronunciation unnatural. Some solutions use Finite-state transducers (FST) to build a grapheme to phoneme conversion model based on the available phoneme table [76]. And the approach for the future when we have enough data and resources, we can train a multilingual speech synthesis model which needs a multilingual dataset, quality and a general phonemic input representation for all languages [77].

One of the common challenges of artificial intelligence models is getting applications to run in real-time and beyond, on devices. In this thesis, we have compressed the speech synthesis model to be able to run in real-time on devices by using distillation, pruning and quantization methods, but the trade-off is that the quality of the model will be reduced, the voice will be unnatural.

As mentioned throughout this thesis, the voice cloning problem will be a trend and a step forward in further research in the field of speech processing because of its applications. We can create voice avatars with just an audio sample, we can even clone cross-lingual, speak any language with our own voice. In the next studies, we will collect more data, aiming to build a multi-speaker speech synthesis model from which we can perform voice cloning better than the voice conversion method used in this thesis.

REFERENCES

[1] MD Singh et al. *Fatigue Detection Using Voice Analysis*. PhD thesis, 2015.

[2] Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, and Yonghui Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis, 2019.

[3] Fadi Biadsy, Ron J. Weiss, Pedro J. Moreno, Dimitri Kanevsky, and Ye Jia. Parrotron: An end-to-end speech-to-speech conversion model and its applications to hearing-impaired speech and speech separation, 2019.

[4] Adam Polyak, Lior Wolf, and Yaniv Taigman. TTS skins: Speaker conversion via ASR. *CoRR*, abs/1904.08983, 2019. URL `http://arxiv.org/abs/1904.08983`.

[5] Kaizhi Qian, Zeyu Jin, Mark Hasegawa-Johnson, and Gautham J. Mysore. F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020. doi: 10.1109/icassp40776.2020.9054734. URL `http://dx.doi.org/10.1109/ICASSP40776.2020.9054734`.

[6] Takuhiro Kaneko and Hirokazu Kameoka. Cyclegan-vc: Non-parallel voice conversion using cycle-consistent adversarial networks. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2100–2104, 2018. doi: 10.23919/EUSIPCO.2018.8553236.

[7] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly,

Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions, 2017.

[8] 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.

[9] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. Autovc: Zero-shot voice style transfer with only autoencoder loss, 2019.

[10] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification, 2020.

[11] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *arXiv preprint arXiv:2010.05646*, 2020.

[12] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv preprint arXiv:1710.07654*, 2017.

[13] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[15] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*, 2016.

[16] Lifa Sun, Shiyin Kang, Kun Li, and Helen Meng. Voice conversion using deep bidirectional long short-term memory based recurrent neural networks. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4869–4873. IEEE, 2015.

[17] NGUYEN Thi Thu Trang and NGUYEN Xuan Tung. Text-to-speech shared task in vlsp campaign 2019: Evaluating vietnamese speech synthesis on common datasets. *Vietnamese Language Signal Processing. VLSP*, 2019.

[18] Mikołaj Bińkowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande, Luis C Cobo, and Karen Simonyan. High fidelity speech synthesis with adversarial networks. *arXiv preprint arXiv:1909.11646*, 2019.

[19] Mingjian Chen, Xu Tan, Yi Ren, Jin Xu, Hao Sun, Sheng Zhao, Tao Qin, and Tie-Yan Liu. Multispeech: Multi-speaker text to speech with transformer, 2020.

[20] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O. Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning, 2018.

[21] Tomoki Toda, Alan W Black, and Keiichi Tokuda. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2222–2235, 2007.

[22] Elina Helander, Tuomas Virtanen, Jani Nurminen, and Moncef Gabbouj. Voice conversion using partial least squares regression. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(5):912–921, 2010.

[23] Yannis Stylianou, Olivier Cappé, and Eric Moulines. Continuous probabilistic transform for voice conversion. *IEEE Transactions on speech and audio processing*, 6(2):131–142, 1998.

[24] Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. Exemplar-based voice conversion using sparse representation in noisy environments. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 96(10):1946–1953, 2013.

[25] Zhizheng Wu, Tuomas Virtanen, Eng Siong Chng, and Haizhou Li. Exemplar-based sparse representation with residual compensation for voice conversion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10): 1506–1521, 2014.

[26] Srinivas Desai, Alan W Black, B Yegnanarayana, and Kishore Prahallad. Spectral mapping using artificial neural networks for voice conversion. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(5):954–964, 2010.

[27] Seyed Hamidreza Mohammadi and Alexander Kain. Voice conversion using deep neural networks with speaker-independent pre-training. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 19–23. IEEE, 2014.

[28] Yuki Saito, Shinnosuke Takamichi, and Hiroshi Saruwatari. Voice conversion using input-to-output highway networks. *IEICE Transactions on Information and Systems*, 100(8):1925–1928, 2017.

[29] Takuhiro Kaneko, Hirokazu Kameoka, Kaoru Hiramatsu, and Kunio Kashino. Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks. In *INTERSPEECH*, volume 2017, pages 1283–1287, 2017.

[30] Yist Y Lin, Chung-Ming Chien, Jheng-Hao Lin, Hung-yi Lee, and Lin-shan Lee. Fragmentvc: Any-to-any voice conversion by end-to-end extracting and fusing fine-grained voice fragments with attention. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5939–5943. IEEE, 2021.

[31] Seung-won Park, Doo-young Kim, and Myun-chul Joe. Cotatron: Transcription-guided speech encoder for any-to-many voice conversion without parallel data. *arXiv preprint arXiv:2005.03295*, 2020.

[32] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In *2016 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2016.

[33] Ju-chieh Chou, Cheng-chieh Yeh, and Hung-yi Lee. One-shot voice conversion by separating speaker and content representations with instance normalization. *arXiv preprint arXiv:1904.05742*, 2019.

[34] Jing-Xuan Zhang, Li-Juan Liu, Yan-Nian Chen, Ya-Jun Hu, Yuan Jiang, Zhen-Hua Ling, and Li-Rong Dai. Voice conversion by cascading automatic speech recognition and text-to-speech synthesis with prosody transfer. *arXiv preprint arXiv:2009.01475*, 2020.

[35] Xiaohai Tian, Eng Siong Chng, and Haizhou Li. A vocoder-free wavenet voice conversion with non-parallel data. *arXiv preprint arXiv:1902.03705*, 2019.

[36] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. Voice conversion from non-parallel corpora using variational auto-encoder. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–6. IEEE, 2016.

[37] Kaizhi Qian, Yang Zhang, Shiyu Chang, Mark Hasegawa-Johnson, and David Cox. Unsupervised speech decomposition via triple information bottleneck. In *International Conference on Machine Learning*, pages 7836–7846. PMLR, 2020.

[38] Da-Yi Wu and Hung-yi Lee. One-shot voice conversion by vector quantization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7734–7738. IEEE, 2020.

[39] Jan Chorowski, Ron J Weiss, Samy Bengio, and Aäron van den Oord. Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM transactions on audio, speech, and language processing*, 27(12):2041–2053, 2019.

[40] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo. Acvae-vc: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder. *arXiv preprint arXiv:1808.05092*, 2018.

[41] Joan Serrà, Santiago Pascual, and Carlos Segura. Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. *arXiv preprint arXiv:1906.00794*, 2019.

[42] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. Neural speech synthesis with transformer network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:6706–6713, 07 2019. doi: 10.1609/aaai.v33i01.33016706.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[44] Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech, 2020.

[45] Yi Ren, Jinglin Liu, Xu Tan, Zhou Zhao, Sheng Zhao, and Tie-Yan Liu. A study of non-autoregressive model for sequence generation. *arXiv preprint arXiv:2004.10454*, 2020.

[46] 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*, volume 33, pages 6706–6713, 2019.

[47] Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep voice: Real-time neural text-to-speech. In *International Conference on Machine Learning*, pages 195–204. PMLR, 2017.

[48] Chengzhu Yu, Heng Lu, Na Hu, Meng Yu, Chao Weng, Kun Xu, Peng Liu, Deyi Tuo, Shiyin Kang, Guangzhi Lei, et al. Durian: Duration informed attention network for multimodal synthesis. *arXiv preprint arXiv:1909.01700*, 2019.

[49] Rafael Valle, Kevin Shih, Ryan Prenger, and Bryan Catanzaro. Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. *arXiv preprint arXiv:2005.05957*, 2020.

[50] Jonathan Shen, Ye Jia, Mike Chrzanowski, Yu Zhang, Isaac Elias, Heiga Zen, and Yonghui Wu. Non-attentive tacotron: Robust and controllable neural tts synthesis including unsupervised duration modeling. *arXiv preprint arXiv:2010.04301*, 2020.

[51] Naihan Li, Yanqing Liu, Yu Wu, Shujie Liu, Sheng Zhao, and Ming Liu. Robutrans: A robust transformer-based text-to-speech model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8228–8235, 2020.

[52] Adrian Łańcucki. Fastpitch: Parallel text-to-speech with pitch prediction. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6588–6592. IEEE, 2021.

[53] 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)*, pages 7209–7213. IEEE, 2020.

[54] Jan Vainer and Ondřej Dušek. Speedyspeech: Efficient neural speech synthesis. *arXiv preprint arXiv:2008.03802*, 2020.

[55] Isaac Elias, Heiga Zen, Jonathan Shen, Yu Zhang, Ye Jia, Ron J Weiss, and Yonghui Wu. Parallel tacotron: Non-autoregressive and controllable tts. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5709–5713. IEEE, 2021.

[56] Ron J Weiss, RJ Skerry-Ryan, Eric Battenberg, Soroosh Mariooryad, and Diederik P Kingma. Wave-tacotron: Spectrogram-free end-to-end text-to-speech synthesis. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5679–5683. IEEE, 2021.

[57] Dan Lim, Won Jang, Heayoung Park, Bongwan Kim, Jaesam Yoon, et al. Jdi-t: Jointly trained duration informed transformer for text-to-speech without explicit alignment. *arXiv preprint arXiv:2005.07799*, 2020.

[58] Jeff Donahue, Sander Dieleman, Mikołaj Bińkowski, Erich Elsen, and Karen Simonyan. End-to-end adversarial text-to-speech. *arXiv preprint arXiv:2006.03575*, 2020.

[59] 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, 2016.

[60] Sercan Ö Arık, Heewoo Jun, and Gregory Diamos. Fast spectrogram inversion using multi-head convolutional neural networks. *IEEE Signal Processing Letters*, 26(1):94–98, 2018.

[61] Aaron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. Parallel wavenet: Fast high-fidelity speech synthesis, 2017.

[62] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech, 2018.

[63] Sungwon Kim, Sang gil Lee, Jongyoon Song, Jaehyeon Kim, and Sungroh Yoon. Flowavenet : A generative flow for raw audio, 2018.

[64] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis, 2018.

[65] Richard Sproat, Alan W Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards. Normalization of non-standard words, 2001.

[66] Min Chu, Hu Peng, and Yong Zhao. Front-end architecture for a multi-lingual text-to-speech system, February 24 2009. US Patent 7,496,498.

[67] David Yarowsky. Text normalization and ambiguity resolution in speech synthesis, 1993.

[68] Richard Sproat. Lightly supervised learning of text normalization: Russian number names, 2010.

[69] Andrea Hoa Pham. Vietnamese rhyme. *Southwest Journal of Linguistics*, 25: 107–142, 01 2006.

[70] Hieu-Thi Luong and Hai-Quan Vu. A non-expert kaldi recipe for vietnamese speech recognition system. In *Proceedings of the Third International Workshop on Worldwide Language Service Infrastructure and Second Workshop on Open Infrastructures and Analysis Frameworks for Human Language Technologies (WLSI/OIAF4HLT2016)*, pages 51–55, 2016.

[71] Robert C Streijl, Stefan Winkler, and David S Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227, 2016.

[72] Junjie Pan, Xiang Yin, Zhiling Zhang, Shichao Liu, Yang Zhang, Zejun Ma, and Yuxuan Wang. A unified sequence-to-sequence front-end model for mandarin text-to-speech synthesis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6689–6693. IEEE, 2020.

[73] Bing Yang, Jiaqi Zhong, and Shan Liu. Pre-trained text representations for improving front-end text processing in mandarin text-to-speech synthesis. In *INTERSPEECH*, pages 4480–4484, 2019.

[74] Yang Zhang, Liqun Deng, and Yasheng Wang. Unified mandarin tts front-end based on distilled bert model. *arXiv preprint arXiv:2012.15404*, 2020.

[75] Junhui Zhang, Junjie Pan, Xiang Yin, Chen Li, Shichao Liu, Yang Zhang, Yuxuan Wang, and Zejun Ma. A hybrid text normalization system using multi-head self-attention for mandarin. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6694–6698. IEEE, 2020.

[76] Josef Robert Novak, Nobuaki Minematsu, and Keikichi Hirose. Phonetisaurus: Exploring grapheme-to-phoneme conversion with joint n-gram models in the wfst framework. *Natural Language Engineering*, 22(6):907–938, 2016.

[77] Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Zhifeng Chen, RJ Skerry-Ryan, Ye Jia, Andrew Rosenberg, and Bhuvana Ramabhadran. Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning. *arXiv preprint arXiv:1907.04448*, 2019.

[78] Vo Quang Dieu Ha, Nguyen Manh Tuan, Cao Xuan Nam, Pham Minh Nhut, and Vu Hai Quan. Vos: the corpus-based etnamese text-to-speech system, 2010.

APPENDIX

APPENDIX A

TEXT NORMALIZATION

A.1 Rules with Regular Expression

By using the Regular Expression to catch patterns that need normalization, which appears frequently in the language, containing numbers and characters, then replaced by syllables found in the dictionary. The output of this step is the paragraph without any digital characters.

Compared to the old VOS-frontend system[78], we do not lower the entire input before processing, thus we can handle cases with different pronunciation for uppercase and lowercase of the same words, and create a premise for backend processing when the capitalized words will be emphasized more through speech synthesis step.

We propose a new set of rules that are more systematic and general, scalable for future works. Based on the different contextual characteristics, rules are divided into four main categories to handle cases that need standardization, including Special case, Timedate, Address and Mathematical. The patterns in each rule set will be proceeded one by one, browse through the entire text to match text need normalization, then return the corresponding normalized string

Special cases rules capture cases that are out of context, with specific formats, including Phone number, Football, Website, Email, etc.

For phone numbers, the identifying feature is a sequence of numbers starting with 0 or a plus sign, the number of digits from 10 to 14 digits. Phone numbers in each country will have a different way of writing, and in fact, there will be different formats, so the three types of phone number rules are listed, the pattern for capturing hotline numbers is also included. Website patterns have identifiable characteristics with a prefix is http(s), www, ftp or suffix is popular domains. Common email pattern is used for matching then spell each letter and symbol by English letter. Sport patterns include specific formats such as lineups, scores, hyphen-minus and dot is not spelled.

Time-date are rules for capturing phrases that show date and time elements. With the time indicating hours, minutes, and seconds, we identify by signals such as ”h, g” or suffixed by AM / PM. These rules need to ensure the validity of time, if time is invalid, the system keeps them for later processing. If ”-” is followed by captured regex, it is read as ”đến”. Date patterns have a form such as DD / MM / YYYY or with the prefix “Ngày, sáng, trưa, chiều, tối, đêm, hôm, etc”. In addition, the timedate rules also capture and handle ”FROM-TO” pattern cases.

Mathematical patterns capture cases containing normal numbers, floating point Numbers, Roman numerals, mathematical expression or unit of measurement.

With normal numbers, we implement a function to convert numbers to letters. However, the normal number not only consists of successive digits but also has a way of writing that splitting them into groups of 3 numbers, separated by commas, dot or space (e.g 12,000,000). This style of writing leads to confusion with floating point numbers, so to avoid false standardization, these cases will be matched first, then floating number, and finally normal number. For strings longer than 15 characters, each digit is converted individually, if the normalized text of number is too large, the string is be separated by commas to read more fluently. If there is an ”- +” in front of the string, it is replaced with ”cộng, “trừ”, except in the case where the number stands at the beginning of a sentence, ”+ -” is treated as a bullet symbol. We categorized into two types of floating point, dot or comma. Floating point is replaced by ”phẩy”, the integer part is read as normal number and discard frontier zero, with a fractional part, frontier zero is replaced with ”không” and the following numbers read as normal numbers.

Unit of measurement are terms that refer to quantities, percentage or currency, followed by numbers, which can be an alphabetic or symbols. Because the system does not lower the input text, it can correctly standardize for upper and lowercase units (e.g Mbps and MBps). Units dictionary distinguishes the upper and lower case. Besides the usual units, the patterns also need to capture ”Unit/Unit” formats, matching strings are checked in the unit dictionary for readable words, if the matching is not in this list, we return the origin matching, keep for later processing. When it is sure that both sides of the slash are units, the ”/” is replaced by ”trên”, this will avoid confusion in the case of ”nam/nữ”. Unit of measurement is also captured in FROM-TO pattern, there are two common types e.g: ”10-20 km/h” and ”10 km/h - 20 km/h”, our system make sure not to be confused with cases where ”-” is read as ”minus”.

For Roman numerals, to ensure accuracy does not fail in capturing, comparing to the old VOS, we just consider uppercase [X, I, V] is able to be Roman numerals, [L, C, D, M] are also Roman characters, but they rarely appear in the text, sometimes even leading to confusion with acronyms. The Roman numerals also have a FROM-TO structure, such as ”XVI-XXI”, but it is easier to handle and more consistent than the unit of measurement. The matching is checked for correctness by converting it into a decimal number, then converting the result back into roman form to compare with the original matching, then the matching is converted to normalized text.

Address-Code are rules for capturing phrases about addresses, locations, codes and all phrases containing numbers. Code number pattern match all remain cases with numbers, the matching sequence will be trim punctuation at both sides and separated into each alphanumeric substrings (e.g: MH370 is split into MH and 370). For each substring, if it is a fully capitalized letter sequence, it will be transcribed, if it contains lowercase letters, the system will keep that letter clusters and add space separate for each cluster. If the substring is numeric and the length of the continuous number string is greater than 4, each number will be transcribed, otherwise, it will read the whole number as a normal number. Cases include special character or symbol will be mapped with corresponding phonetics, but with ”-” is not spelled.

A.2 Dictionary Checking

After running through the rules sets, the string just contains letters, space and special characters. This string will be split into many segments which separated by spaces, each token containing no space and no special characters before and after the string. The system runs over each token to validate if it is readable by checking it in Dictionary Vietnamese syllable which includes 7698 syllables. If the token does not exist, we look it up in the mapping dictionary instead, search and replace it with the corresponding word.

Abbreviations mapping dictionary includes 3 different mapping methods. With acronyms such as ”NSƯT, GDĐT”, we have to normalize to its original form. The second type is initialisms, including words like STEM and UNESCO, so we have to transcribe them. The last type is the acronyms that need to spell each letter such as PNJ, FPT, AFF, we do not handle it in this step and consider it as unknown to limit misunderstandings when not sure.

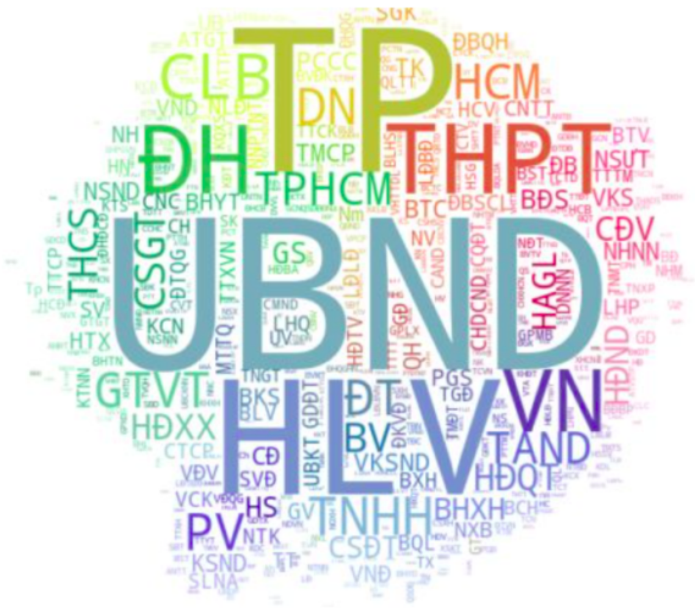


Figure A.1: Word Cloud for Vietnamese Abbreviation from Statistics on News.

With dictionary mapping abbreviations, uppercase and lowercase tokens are often referred to as two separate objects, with different contexts and probabilities, an acronym can have more than one meaning. Acronyms that appear less frequently are filtered out to limit misunderstandings

Teen Code - Slang - Lingo mapping dictionary is updated by adding more than 300 slangs like ”H’Hen Nie, Ea H’leo”, added some lingo that does not appear in Popular Dictionary, words that backend doesn’t support like “Đắk Lắk, Pleiku”. It also handles inconsistency in writing of the same word, such as ”thuỷ - thủy” or ”tuỳ - tùy”. Loanwords like ”oxy, axit” and common misspellings are also updated.

Special Symbols After browsing through the dictionaries, if the token does not belong to any Mapping Dictionaries, we continue to split the token into smaller substrings separated by symbols and special characters. The system shows the corresponding reading for each symbol, non-stop punctuation such as brackets, quotes are removed. The system continues checking each substring in the mapping dictionaries again, this process (tokenize strings with space first, if the token is unknown, then tokenize with the special character) avoids errors when the string is written adjacent to each other. This way will handle both cases “GD-ĐT” and “ê-kíp”, the hyphen in the first case will be omitted to ”GDĐT”, the second case will be replaced with space to ”ê kíp”, and many similar cases with symbolic problems are also solved. If the token is still not in any mapping dictionary, we treat it as an unknown word, if it is uppercase, spell each character as an English letter, if it is lowercase and containing vowels, we will leave the backend to handle by letter to sound, if it does not contain the vowel, spell each character as Vietnamese letter.

Punctuations: The final step is to standardize the output text, including removing duplicate white spaces, handling punctuations including removing no voice marks like “()[]”, and replace all punctuation marks with only comma and dot, which represent as the sound unit.

A.3 Vietnamese Syllables Statistics

In order to update the Vietnamese syllables used today, we proceeded to build a News Corpus with sources of 9 online newspapers, crawled from 7/2018. The Corpus is divided at the sentence level, consisting of a total of 6,308,173 sentences, the number of unnormalized sentences is 3,740,507 sentences, accounting for more than 59.29 %.

Table A.1: Number of sentences from popular Vietnamese newspapers

dantri	danviet	nld	thanhnien	tto
653545	386444	103218	634974	177287
tuoitre	vnexpress	vnn	zingnews	
167162	157202	481566	979109	

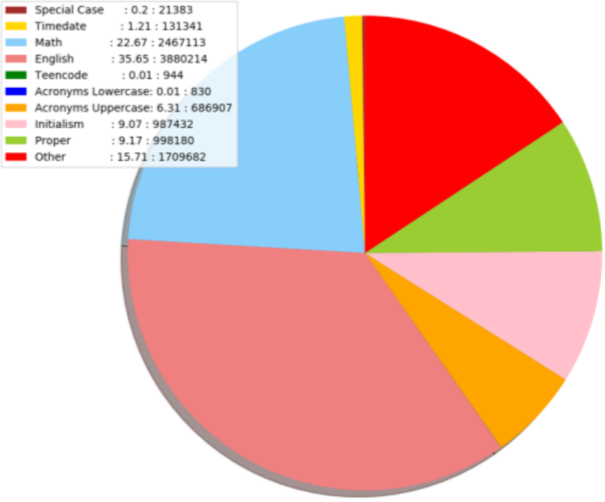


Figure A.2: Statistics on Tokens need to be standardized in Vietnamese.

We continue to word-tokenize every sentence, totaling 10.88 million tokens. The tokens will be classified into groups such as Special Case, Time-date, Math and Number, English, Slang, Teencode, Acronyms, Initialism, Proper Noun, etc.

Table A.2: Statistics on Tokens need to be standardized in Vietnamese

Category	Percent	Number of sentences
Special	0.2	21383
Timedata	1.21	131341
Math	22.67	2467113
English	35.65	3880214
Acronyms Lowercase	0.01	830
Acronyms Uppercase	6.31	686907
Teencode	0.01	944
Intialism	9.07	987432
Proper	9.17	998180
Other	15.71	1709682

Some notable points from the statistics on News Corpus are: special cases only account for 0.2%, most of the Timedate cases are the publication date of the news, English accounts for 40 percent, and most of the abbreviations are all in uppercase.

Type	Case	Test (thực tế,bao quát, không trùng lắp, tỉ lệ theo độ phổ biến)	VOS	VOS Update Frontend	Output
20	Mathematic: normal number	Tại cơ quan báo điện tử Dân trí, sau khi biết tin được bạn đọc giúp đỡ số tiền **285.550.000** đồng	🟩	🟩	tại cơ quan báo điện tử dân trí , sau khi biết tin được bạn đọc giúp đỡ số tiền hai trăm tám mươi lăm triệu năm trăm năm mươi nghìn đồng
	Mathematic: normal number	Tổng doanh số bán hàng của toàn thị trường đạt **17.067** xe, trong đó có **11.625** xe du lịch, **4.174** xe thương mại và 180 xe chuyên dụng	🟩	🟩	tổng doanh số bán hàng của toàn thị đạt mười bảy nghìn không trăm sáu mươi bảy xe , trong đó có mười một nghìn sáu trăm hai mươi lăm xe du lịch , bốn nghìn một trăm bảy mươi tư xe thương mại và một trăm tám mươi xe chuyên dụng
	Mathematic: normal number	quỹ đầu tư vàng lớn thế giới đã bán ra lượng vàng lớn với **21,75** tấn vàng, lượng vàng nắm giữ còn **802,12** tấn.	🟩	🟩	quỹ đầu tư vàng lớn thế giới đã bán ra lượng vàng lớn với hai mươi mốt phẩy bảy mươi lăm tấn vàng , lượng vàng nắm giữ còn tám trăm linh hai phẩy mười hai tấn .
	Mathematic: normal number	hậu quả sau: **1-** Buộc nộp lại số lợi bất hợp pháp có được do thực hiện hành vi vi phạm hành chính; **2-** Buộc thu hồi	🟩	🟩	hậu quả sau . một buộc nộp lại số lợi bất hợp pháp có được do thực hiện hành vi vi phạm hành chính , hai buộc thu hồi
	Mathematic: normal number + measure	giá vàng giao ngay tại châu Á qua niêm yết có biên độ giảm nhẹ xuống mức **1.313,6 USD/ounce.**	🟥	🟩	giá vàng giao ngay tại châu á qua niêm yết có biên độ giảm nhẹ xuống mức một nghìn ba trăm mười ba phẩy sáu diu ét đi một ao .
RULE	Mathematic: Measure	chính phủ Hàn Quốc ước tính sẽ mất khoảng **3,2 nghìn tỷ won** để cung cấp khoảng **2 triệu KW điện** cho Triều Tiên	🟥	🟩	chính phủ hàn quốc ước tính sẽ mất khoảng ba phẩy hai nghìn tỷ quan để cung cấp khoảng hai triệu kí lô quát điện cho triều tiên
	Mathematic: Measure	giá vàng đang được giao dịch ở mức **36,95 triệu đồng/lượng (mua vào) - 37,22 triệu đồng/lượng (bán ra)**, tăng tiếp mỗi chiều **70.000 đồng và 170.000 đồng/lượng** so với phiên hôm qua.	🟥	🟨	giá vàng đang được giao dịch ở mức ba mươi sáu phẩy chín mươi lăm triệu đồng , lượng mua vào ba mươi bảy phẩy hai mươi hai triệu đồng , lượng bán ra , tăng tiếp mỗi chiều bảy mươi nghìn đồng và một trăm bảy mươi nghìn đồng , lượng so với phiên hôm qua .
	Mathematic: Measure	Ngổn ngang nỗi lo đầu năm học mới: Phổ biến tình trạng **55 học sinh/lớp**	🟥	🟨	ngổn ngang nỗi lo đầu năm học mới . phổ biến tình trạng năm mươi lăm học sinh , lớp
	Mathematic: Measure	nâng giá đất Cần Giờ lên gấp **5-7** lần, cao ngất ngưởng trên dưới 30 triệu **đồng/m^2**	🟥	🟥	nâng giá đất cần giờ lên gấp năm bảy lần , cao ngất ngưởng trên dưới ba mươi triệu đồng , mờ
	Mathematic: Measure	sở hữu động cơ V8 **4.0 lít** sản sinh công suất 789 mã lực và **800Nm mô-men xoắn** cực đại.	🟥	🟩	sở hữu động cơ vê tám bốn . không lít sản sinh công suất bảy trăm tám mươi chín mã lực và tám trăm na nô mét mô men xoắn cực đại .
	Mathematic: Measure	thu mua **180.000 đồng/kg** thay vì gần **670.000 đồng/kg (30 USD/kg)**	🟥	🟨	thu mua một trăm tám mươi nghìn đồng một kí lô gam thay vì gần sáu trăm bảy mươi nghìn đồng một kí lô gam ba mươi diu ét đi , ki lô gam
	Mathematic: Measure	không phát hiện về vấn đề hô hấp đối với Việt, nhưng trái lại em ấy chỉ nặng **48kg** và cao **1,60m**	🟥	🟩	không phát hiện về vấn đề hô hấp đối với việt , nhưng trái lại em ấy chỉ nặng bốn mươi tám kí lô gam và cao một phẩy sáu mươi mét
	Mathematic: Measure	ông Tự được mua với giá rẻ bởi cá ngừ đại dương hiện nay dao động **100.000-120.000 đồng/kg.**	🟥	🟩	ông tự được mua với giá rẻ bởi cá ngừ đại dương hiện nay dao động một trăm nghìn một trăm hai mươi nghìn đồng trên kí lô gam .
	Mathematic: phase	tổng sản phẩm quốc gia của Triều Tiên chỉ bằng **1/45** so với Hàn Quốc	🟨	🟨	tổng sản phẩm quốc gia của triều tiên chỉ bằng một , bốn mươi lăm so với hàn quốc
	Mathematic: phase	Tìm giá trị của biến x để **A ≥ k** (hoặc **A ≤ k, A > k, A < k** …)	🟥	🟩	tìm giá trị của biến x để a lớn hơn hoặc bằng k hoặc a nhỏ hơn hoặc bằng k , a lớn hơn k , a nhỏ hơn k
	Mathematic: Roman Number	Nói chuẩn bị Đại hội **XIII** không phải chỉ cho đến năm 2026 mà phải có tầm nhìn chiến lược dài hơn	🟥	🟩	nói chuẩn bị đại hội mười ba không phải chỉ cho đến năm hai nghìn không trăm hai mươi sáu mà phải có tầm nhìn chiến lược dài hơn

	Mathematic: Roman Number	Địa ốc Alibaba rao bán 3 khu đất là dự án Alibaba Central Park, Alibaba Central Park **II**, Alibaba Central Park **III**.	🟥	🟩	địa ốc alibaba rao bán ba khu đất là dự án alibaba central park , alibaba central park hai , alibaba central park ba .
	Mathematic: Roman Number	Đại hội lần thứ **XIII** của Đảng có thể coi là Đại hội bản lề mang tầm chiến lược của nửa đầu thế kỷ **XXI**	🟩	🟩	đại hội lần thứ mười ba của đảng có thể coi là đại hội bản lề mang tầm chiến lược của nửa đầu thế kỷ hai mươi mốt
	Mathematic: Roman Number	trước kỳ họp thứ 6, Quốc hội khóa **XIV**, công nhân lao động gang thép Thái Nguyên đã gửi tâm thư	🟩	🟩	trước kỳ họp thứ sáu , quốc hội khóa mười bốn , công nhân lao động gang thép thái nguyên đã gửi tâm thư
	Mathematic: Roman Number	Chùa Cầu mang các đặc trưng kiến trúc cổ của Hội An thế kỷ **XVIII-XIX**.	🟥	🟩	chùa cầu mang các đặc trưng kiến trúc cổ của hội an thế kỷ mười tám mười chín .
15	Timedate: date	**Chiều 3/10**, một nghi phạm trong vụ nhà báo mất tích đã thiệt mạng trong một tai nạn ôtô	🟩	🟩	chiều ba tháng mười , một nghi phạm trong vụ nhà báo mất tích đã thiệt mạng trong một tai nạn ôtô
	Timedate: date	**Ngày 24/9**, Việt cùng với người nhà vào TP.HCM để khám sức khỏe.	🟩	🟩	ngày hai mươi tư tháng chín , việt cùng với người nhà vào thành phố hồ chí minh để khám sức khỏe .
	Timedate: date	**ngày 25.10.2017**, rất nhiều cư dân mạng đã vào tài khoản cá nhân của Trương Hạo Liêm để tố giác	🟩	🟩	ngày hai mươi lăm tháng mười năm hai nghìn không trăm mười bảy , rất nhiều cư dân mạng đã vào tài khoản cá nhân của trương hạo liêm để tố giác
	Timedate: date	Venezuela đã thông qua bầu cử **(5/2018)** để lựa chọn chính phủ này, nó là hợp hiến	🟥	🟨	venezuela đã thông qua bầu cử năm , hai nghìn không trăm mười tám để lựa chọn chính phủ này , nó là hợp hiến
	Timedate: date	sau khi báo cáo doanh thu dưới ước tính của giới phân tích trong quý **1/2019**.	🟥	🟩	sau khi báo cáo doanh thu dưới ước tính của giới phân tích trong quý một , hai nghìn không trăm mười chín .
	Timedate: date	**giai đoạn 2021 - 2026**, thị trường hình thành một mức giá ảo làm cho tính thanh khoản bị tê liệt	🟩	🟩	giai đoạn hai nghìn không trăm hai mươi mốt hai nghìn không trăm hai mươi sáu , thị trường hình thành một mức giá ảo làm cho tính thanh khoản bị tê liệt
	Timedate: date	trước hết là những lỗ hổng trong quy chế thi, ngăn chặn kịp thời gian lận trong thi cử của **năm học 2019 - 2020**.	🟩	🟩	trước hết là những lỗ hổng trong quy chế thi , ngăn chặn kịp thời gian lận trong thi cử của năm học hai nghìn không trăm mười chín hai nghìn không trăm hai mươi .
	Timedate: date	Sau **đêm 26-2**, quả tim mới đã đập rộn ràng trong lồng ngực người công nhân nghèo.	🟩	🟩	sau đêm hai mươi sáu tháng hai , quả tim mới đã đập rộn ràng trong lồng ngực người công nhân nghèo .
	Timedate: date	Dự báo đợt nắng nóng và thời tiết oi bức ở các tỉnh miền Bắc sẽ giảm dần từ **15-8** khi nhiệt độ giảm xuống còn 34 độ.	🟩	🟨	dự báo đợt nắng nóng và thời tiết oi bức ở các tỉnh miền bắc sẽ giảm dần từ mười lăm tám khi nhiệt độ giảm xuống còn ba mươi tư độ .
	Timedate: date	dành cho những học sinh đăng ký tham dự **ngày 11&12/03**.	🟥	🟨	dành cho những học sinh đăng ký tham dự ngày mười một và mười hai xuyệt không ba .
	Timedate: time	Hơn **10h** trưa, Lan với bạn trai mới về đến nhà.	🟩	🟩	hơn mười giờ trưa , lan với bạn trai mới về đến nhà .
	Timedate: time	Khu trang trại rộng 5.000 mét có hệ thống camera hoạt động **24/24h**.	🟩	🟨	khu trang trại rộng năm nghìn mét có hệ thống camera hoạt động hai mươi tư , hai mươi tư giờ .
	Timedate: time	đám cháy được phát hiện vào khoảng **7h36** tại kho hàng hoá cho thuê thuộc Công ty Cổ phần	🟩	🟩	đám cháy được phát hiện vào khoảng bảy giờ ba mươi sáu tại kho hàng cho thuê thuộc công ty cổ phần
	Timedate: time	Từ **9h - 11h45 (thứ 2 - thứ 7)**, bạn sẽ nhận hosting sau khoảng **20-40 phút** từ khi gửi yêu cầu nhận host	🟥	🟩	từ chín giờ đến mười một giờ bốn mươi lăm thứ hai thứ bảy , bạn sẽ nhận hosting sau khoảng hai mươi bốn mươi phút từ khi gửi yêu cầu nhận host
	Timedate: time	Thời gian mở cửa: **9:30am - 8:00pm** tất cả các ngày trong tuần	🟥	🟩	thời gian mở cửa . chín giờ ba mươi ây em đến tám giờ bi em tất cả các ngày trong tuần
15	Address - Code number: Political Division	Tại điểm tập kết rác trên đường Tam Bình **(KP.2, P.Tam Phú, Q.Thủ Đức)**, cũng không còn cảnh ùn ứ, rác tràn bên lề đường trước đó đã được hốt gọn, vệ sinh khu vực được làm sạch sẽ.	🟥	🟩	tại điểm tập kết rác trên đường tam bình khu phố hai , phường tam phú , quận thủ đức , cũng không còn cảnh ùn ứ , rác tràn bên lề đường trước đó đã được hốt gọn , vệ sinh khu vực được làm sạch sẽ .

	Address - Code number: Political Division	Em Triệu là học sinh lớp Trường tiểu học Hưng Định, P. Hưng Định, **TX.Thuận An**	🟩	🟩	em triệu là học sinh lớp trường tiểu học hưng định , phường hưng định , thị xã thuận an
	Address - Code number: Political Division	Công an **H.**Bến Cầu, Tây Ninh đã bàn giao nghi can Lê Ngọc Hải 22 tuổi, ngụ **P.2, Q.8, TP.HCM** cho Công an H. Bình Chánh TP.HCM	🟨	🟩	công an huyện bến cầu , tây ninh đã bàn giao nghi can lê ngọc hải hai mươi hai tuổi , ngụ phường hai , quận tám , thành phố hồ chí minh cho công an huyện bình chánh thành phố hồ chí minh
	Address - Code number: Street	**Lô B1, đường C2**, phường Thạnh Mỹ Lợi, Khu công nghiệp Cát Lái, quận 2, thành phố Hồ Chí Minh.	🟥	🟩	lô bê một , đường xê hai , phường thạnh mỹ lợi , khu công nghiệp cát lái , quận hai , thành phố hồ chí minh .
	Address - Code number: Street	anh cùng con gái đến quán cà phê của gia đình trên **đường D13**, quận Tân Phú để chơi.	🟥	🟩	anh cùng con gái đến quán cà phê của gia đình trên đường dê mười ba , quận tân phú để chơi .
	Address - Code number: Street	nằm Lô C2, **đường N17-18**, KCN Sóng Thần 2, thị xã Dĩ An, Bình Dương	🟥	🟩	nằm lô xê hai , đường nờ mười bảy , mười tám , khu công nghiệp sóng thần hai , thị xã dĩ an , bình dương
	Address - Code number: Office	Học sinh khối 12 học trên tầng 5 nhưng vì **lớp 12A7** có Hoài Thương nên chúng tôi bố trí lớp ở tầng 1 để tiện hơn cho Thương.	🟩	🟩	học sinh khối mười hai học trên tầng năm nhưng vì lớp mười hai a bảy có hoài thương nên chúng tôi bố trí lớp ở tầng một để tiện hơn cho thương .
	Address - Code number: Office	khách hàng mua căn hộ **B20-05** tại dự án, cho biết: Chậm tiến độ là một trong những điều lo ngại nhất vì chúng tôi mua nhà để ở	🟥	🟩	khách hàng mua căn hộ bê hai mươi , không năm tại dự án , cho biết . chậm tiến độ là một trong những điều lo ngại nhất vì chúng tôi mua nhà để ở
	Address - Code number: Office	hai tòa căn hộ **C1 và C2** có chiều cao 25 tầng, được xây dựng	🟥	🟩	hai tòa căn hộ xê một và xê hai có chiều cao hai mươi lăm tầng , được xây dựng
	Address - Code number: Office	hoang mang lo lắng vô cùng cho cuộc sống về sau tại chung cư này, bà Nguyễn Thị Ngọc Mai, **căn hộ A04.10** cho biết.	🟥	🟩	hoang mang lo lắng vô cùng cho cuộc sống về sau tại chung cư này , bà nguyễn thị ngọc mai , căn hộ a không bốn chấm mười cho biết .
	Address - Code number: Office	Trong đó, toà **căn hộ M1** nổi bật với vị trí tách biệt với các tòa còn lại, sở hữu bốn mặt thoáng rộng.	🟥	🟩	trong đó , căn hộ mờ một nổi bật với vị trí tách biệt với các tòa còn lại , sở hữu bốn mặt thoáng rộng .
	Address - Code number	xem xét lại hợp đồng bán máy bay chiến đấu **F-35** và hệ thống **S-400** cho Thổ Nhĩ Kỳ	🟥	🟩	xem xét lại hợp đồng bán máy bay chiến đấu ép , ba mươi lăm và hệ thống ét , bốn trăm cho thổ nhĩ kỳ
	Address - Code number	nghiên cứu điều chỉnh quy hoạch dự án theo Thông báo **số 331-TB/BTV** của Ban Thường vụ Thành ủy Đà Nẵng.	🟥	🟥	nghiên cứu điều chỉnh quy hoạch dự án theo thông báo số ba trăm ba mươi mốt thông báo , biên tập viên của ban thường vụ thành ủy đà nẵng .
	Address - Code number	tàu cá **KH96662** do ông Trần Văn Tự làm thuyền trưởng, cập cảng cá Hòn Rớ, Nha Trang.	🟥	🟩	tàu cá ca hát chín sáu sáu sáu hai do ông trần văn tự làm thuyền trưởng , cập cảng cá hòn rớ , nha trang .
	Address - Code number	xin trân trọng gửi đến quý vị và các bạn lộ trình xe bus tuyến **47A**: BX Long Biên – Bát Tràng	🟩	🟩	xin trân trọng gửi đến quý vị và các bạn lộ trình xe bus tuyến bốn mươi bảy a . bến xe long biên bát tràng
15	Special Case: sđt	quan tâm tới chương trình này đều có thể gọi điện thoại tới số hotline **090 6699 036** để đăng ký và đặt hẹn.	🟥	🟩	quan tâm tới chương trình này đều có thể gọi điện thoại tới số hotline không chín không sáu sáu chín chín không ba sáu để đăng ký và đặt hẹn .
	Special Case: sđt	SĐT**: 0165.439.1742**	🟥	🟩	số điện thoại . không một sáu năm bốn ba chín một bảy bốn hai
	Special Case: sđt	cần báo ngay cho Tổng đài chăm sóc khách hàng của Tổng Công ty Điện lực Hà Nội theo số điện thoại **19001288** - 024 22222000 (phục vụ 24/7), chính quyền hoặc Công an địa phương, đơn vị quản lý điện gần nhất để kịp thời xử lý.	🟥	🟩	cần báo ngay cho tổng đài chăm sóc khách hàng của tổng công ty điện lực hà nội theo số điện thoại một chín không không một hai tám tám không hai bốn hai hai hai hai hai không không không phục vụ hai mươi tư , bảy , chính quyền hoặc công an địa phương , đơn vị quản lý điện gần nhất để kịp thời xử lý .
	Special Case: website	trong đó có nghiên cứu xuất bản của Hiệp hội Thể thao Nữ của Mỹ, (**https://bit.ly/2KLIEZo**), chỉ ra rằng chơi thể thao giúp học sinh đạt kết quả học tập tốt hơn	🟥	🟨	trong đó có nghiên cứu xuất bản của hiệp hội thể thao nữ của mỹ , hát tê tê pê ét xuyệt xuyệt bê i tê chấm lờ i xuyệt hai ca lờ lờ e giét o , chỉ ra rằng chơi thể thao giúp học sinh đạt kết quả học tập tốt hơn

	Special Case: website	Công ty Việt Nam quản lý để kiểm tra thông tin gói sản phẩm mình vừa mua tại đại lý: http://vews.**3m**.com	🟥	🟨	công ty việt nam quản lý để kiểm tra thông tin gói sản phẩm mình vừa mua tại đại lý . hát tê tê pê xuyệt xuyệt vê e vê kép ét chấm ba mờ chấm com
	Special Case: website	Hòa Phát trong Top 10 DN lớn nhất 2018 (Nguồn: **www.vnr500.com.vn**)	🟥	🟨	hòa phát trong top mười doanh nghiệp lớn nhất hai nghìn không trăm mười tám nguồn . vê kép vê kép vê kép chấm vê nờ rờ năm không không chấm com chấm vê nờ
	Special Case: website	Chương trình do nhóm yêu thích du lịch và khám phá trên diễn đàn **phuot.vn** khởi xướng.	🟥	🟨	chương trình do nhóm yêu thích du lịch và khám phá trên diễn đàn phuot . việt nam khởi xướng .
	Special Case: email	Chúng tôi chân thành đón nhận các phản hồi từ mọi người thông qua bình luận hoặc email về cho chúng tôi tại **tripx.vn@gmail.com.**	🟥	🟨	chúng tôi chân thành đón nhận các phản hồi từ mọi người thông qua bình luận hoặc email về cho chúng tôi tại ti a ai pi ít chấm vi en a còng giy meo chấm com .
	Special Case: coordinates	Về tàu lao động neo đậu tại **7o30’58”** Vĩ độ Bắc, **109o49’55”** Kinh Đông, gần nhà dàn Tư Chính, Uỷ ban Quốc gia Ứng phó sự cố, thiên tai	🟥	🟨	về tàu lao động neo đậu tại bảy o ba mươi năm mươi tám vĩ độ bắc , một trăm linh chín o bốn mươi chín năm mươi lăm kinh đông , gần nhà dàn tư chính , uỷ ban quốc gia ứng phó sự cố , thiên tai
	Special Case: footbal	đội tuyển Quốc gia Việt Nam đã hạ những chú voi chiến Thái Lan với tỉ số 1-0 để giành quyền vào chung kết	🟥	🟩	đội tuyển quốc gia việt nam đã hạ những chú voi chiến thái lan với tỉ số một không để giành quyền vào chung kết
	Special Case: footbal	có chiến thắng kinh hoàng **10-0** và đây chính là trận thắng có cách biệt lớn nhất	🟥	🟩	có chiến thắng kinh hoàng mười không và đây chính là trận thắng có cách biệt lớn nhất
	Special Case: footbal (thể thao nói chung)	Tiến Minh đã thắng tay vợt trẻ Trung Quốc **2-1 (21-23, 21-11, 21-9).**	🟥	🟩	tiến minh đã thắng tay vợt trẻ trung quốc hai một hai mươi mốt hai mươi ba , hai mươi mốt mười một , hai mươi mốt chín .
	Special Case: footbal	Cả hai đội đều tung ra sân đội hình mạnh nhất, Real Madrid đá **4-3-3** với Casemiro, Modric, Kroos hỗ trợ	🟥	🟩	cả hai đội đều tung ra sân đội hình mạnh nhất , real madrid đá bốn ba , ba với casemiro , modric , kroos hỗ trợ
	Special Case: footbal	Luis Suarez căng ngang khiến Varane lóng ngóng đá phản lưới nhà, nâng tỷ số lên **2-0** cho Barcelona	🟥	🟩	luis suarez căng ngang khiến varane lóng ngóng đá phản lưới nhà , nâng tỷ số lên hai không cho barcelona
	Special Case: footbal	Có khoảng 220 cầu thủ của 8 đội ở vòng chung kết **U19** quốc gia là dịp ông tận mắt theo dõi đầy đủ những tinh hoa tương lai của bóng đá Việt Nam.	🟩	🟩	có khoảng hai trăm hai mươi cầu thủ của tám đội ở vòng chung kết u mười chín quốc gia là dịp ông tận mắt theo dõi đầy đủ những tinh hoa tương lai của bóng đá việt nam .
19	Acronyms - Initialism	Sở **GD-ĐT** TPHCM vừa công bố kế hoạch thực hiện công tác cải cách hành chính	🟩	🟩	sở giáo dục đào tạo thành phố hồ chí minh vừa công bố kế hoạch thực hiện công tác cải cách hành chính
	Acronyms - Initialism	Hội Sinh viên Việt Nam, Hội **LHTN** Việt Nam, Đội **TNTP** Hồ Chí Minh… và hệ thống các kênh fanpage “vệ tinh”, hệ thống các hội, nhóm	🟩	🟩	hội sinh viên việt nam , hội liên hiệp thanh niên việt nam , đội thiếu niên tiền phong hồ chí minh và hệ thống các kênh fanpage vệ tinh , hệ thống các hội , nhóm
	Acronyms - Initialism	**PV** Dân trí có liên lạc qua điện thoại với Đại tá, **PGS, TS** Trần Sơn Hà - Hiệu trưởng Trường sĩ quan Thông tin và được thầy Hà yêu cầu cần đến tận nơi	🟥	🟩	phóng viên dân trí có liên lạc qua điện thoại với đại tá , phó giáo sư , tiến sĩ trần sơn hà hiệu trưởng trường sĩ quan thông tin và được thầy hà yêu cầu cần đến tận nơi
	Acronyms - Initialism	dự án máy bay chiến đấu thế hệ mới của Mỹ mang tên **PCA**, tạm dịch: Xuyên thủng lá chắn phòng không đối phương, đang được Mỹ nghiên cứu và phát triển.	🟥	🟩	dự án máy bay chiến đấu thế hệ mới của mỹ mang tên pi si ây , tạm dịch . xuyên thủng lá chắn phòng không đối phương , đang được mỹ nghiên cứu và phát triển .
	Acronyms - Initialism	Nghị quyết số 20 – **NQ/TW** của Hội nghị Ban CHTW vẫn còn hạn chế, Chủ tịch nước Nguyễn Phú Trọng phát biểu khai mạc	🟥	🟩	nghị quyết số hai mươi nghị quyết trung ương của hội nghị ban chấp hành trung ương vẫn còn hạn chế , chủ tịch nước nguyễn phú trọng phát biểu khai mạc
	Acronyms - Initialism	Nghị định **91/2015/NĐ-CP** (không vượt quá 3 lần), kiểm soát công nợ của **TKV** ngày càng có hiệu, đây là những chỉ số **SXKD** nổi bật năm 2018 của **TKV**.	🟥	🟨	nghị định chín mươi mốt , hai nghìn không trăm mười lăm xuyệt nờ đê , xê pê không vượt quá ba lần , kiểm soát công nợ của ti cây vi ngày càng có hiệu , đây là những chỉ số sản xuất kinh doanh nổi bật năm hai nghìn không trăm mười tám của ti cây vi .
	Acronyms - Initialism	Theo **ĐBQH** Hoàng Văn Hùng đề nghị với Chính phủ, Bộ Công Thương chỉ đạo quyết liệt	🟩	🟩	theo đại biểu quốc hội hoàng văn hùng đề nghị với chính phủ , bộ công thương chỉ đạo quyết liệt

	Acronyms - Initialism	Tặng gói **DV** 100 triệu - XIN HỌC BỔNG dành cho 5 bạn đầu tiên nộp hồ sơ tại triển lãm	🟩	🟩	tặng gói dịch vụ một trăm triệu xin học bổng dành cho năm bạn đầu tiên nộp hồ sơ tại triển lãm
	Acronyms - Initialism	Sở **NN-PTNT** Quảng Ngãi cho biết trên địa bàn tỉnh vừa xuất hiện ổ dịch tả lợn	🟥	🟩	sở nông nghiệp phát triển nông thôn quảng ngãi cho biết trên địa bàn tỉnh vừa xuất hiện ổ dịch tả lợn
	Acronyms - Initialism	**HĐXX** xác định Nguyễn Minh Hùng, cựu Chủ tịch Pharma, phạm tội Buôn bán hàng giả là thuốc chữa bệnh, tuyên phạt 17 năm tù	🟩	🟩	hội đồng xét xử xác định nguyễn minh hùng , cựu chủ tịch pharma , phạm tội buôn bán hàng giả là thuốc chữa bệnh , tuyên phạt mười bảy năm tù
	Acronyms - Initialism	chủ tịch **CĐ** công ty, đa số **CN** đều có nhu cầu sử dụng điện thoại thông minh để truy cập internet, liên lạc với gia đình nên **CĐ** đã kết hợp với các **DN** bán hàng để **CN** mua trả góp, không lãi suất trên thiết bị này.	🟥	🟨	chủ tịch cao đẳng công ty , đa số si en đều có nhu cầu sử dụng điện thoại thông minh để truy cập internet , liên lạc với gia đình nên cao đẳng đã kết hợp với các doanh nghiệp bán hàng để si en mua trả góp , không lãi suất trên thiết bị này .
	Acronyms - Initialism	Việt Nam đã gặt hái thành công rực rỡ ở các giải đấu tầm châu lục cũng như khu vực **AFF** 2018	🟥	🟩	việt nam đã gặt hái thành công rực rỡ ở các giải đấu tầm châu lục cũng như khu vực ây ép ép hai nghìn không trăm mười tám
	Acronyms - Initialism	chương trình **VEF** 2.0 xin mời các bạn ứng viên tiềm năng, có quan tâm tới chương trình	🟥	🟩	chương trình vi i ép hai . không xin mời các bạn ứng viên tiềm năng , có quan tâm tới chương trình
	Acronyms - Initialism	Sở TT&TT tỉnh Bến Tre phát hiện Nguyễn Ngọc Ánh sử dụng nhiều mạng xã hội phát trực tiếp, trao đổi thông tin, cung cấp hình ảnh, tư liệu	🟥	🟩	sở thông tin và truyền thông tỉnh bến tre phát hiện nguyễn ngọc ánh sử dụng nhiều mạng xã hội phát trực tiếp , trao đổi thông tin , cung cấp hình ảnh , tư liệu
	Acronyms - Initialism	tôi đành tuyên bố với mọi người việc không phát hành bất kỳ **DVD** nào nữa	🟥	🟩	tôi đành tuyên bố với mọi người việc không phát hành bất kỳ đi vi đi nào nữa
	Acronyms - Initialism	**MIKGroup** không liên quan đến các dự án có tên	🟥	🟩	mikgroup không liên quan đến các dự án có tên
	Acronyms - Initialism	Chúng tôi sẽ kiến nghị lãnh đạo **BYT** cử cán bộ của các **BV** trong giai đoạn bệnh nhân chờ tái khám	🟩	🟩	chúng tôi sẽ kiến nghị lãnh đạo bộ y tế cử cán bộ của các bệnh viện trong giai đoạn bệnh nhân chờ tái khám
DICTIONARY	Acronyms - Initialism	Chủ tịch **HĐTV** Công ty Luật **BASICO**, Trọng tài viên **VIAC** cho rằng, gần như không có sự tranh cãi về trách nhiệm trong việc xảy ra mất tiền gửi ngân hàng đối với những trường hợp chỉ	🟥	🟨	chủ tịch hội đồng tư vấn công ty luật bi ây ét ai si âu , trọng tài viên vi ai ây si cho rằng , gần như không có sự tranh cãi về trách nhiệm trong việc xảy ra mất tiền gửi ngân hàng đối với những trường hợp chỉ
	Acronyms - Initialism	đây là tố chất không phải đạo diễn trẻ nào cũng có **NSƯT** CÔNG NINH	🟩	🟩	đây là tố chất không phải đạo diễn trẻ nào cũng có nghệ sĩ ưu tú công ninh
	Acronyms - Initialism	Bộ **GTVT** thanh tra 2 dự án **BOT** qua Nghệ An, Hà Tĩnh Bộ **GTVT** vừa công bố quyết định thanh tra 2 dự án **BOT** là dự án đầu tư xây dựng công trình nâng cấp, mở rộng **QL1A**	🟥	🟥	bộ giao thông vận tải thanh tra hai dự án bi âu ti qua nghệ an , hà tĩnh bộ giao thông vận tải vừa công bố quyết định thanh tra hai dự án bi âu ti là dự án đầu tư xây dựng công trình nâng cấp , mở rộng quy lờ một a
10	Teencode - Slang - Lingo	Ảnh cắt từ clip của **FB** Trần Thị Thanh Thanh	🟥	🟩	ảnh cắt từ clip của facebook trần thị thanh thanh
	Teencode - Slang - Lingo	Nổi bật trong đó là **nhữg** bài học giáo dục giới tính được truyền đạt	🟩	🟩	nổi bật trong đó là những bài học giáo dục giới tính được truyền đạt
	Teencode - Slang - Lingo	Chị **Kăn** Mèo, thôn Ti Nê, xã A Bung nói rằng, dệt thổ cẩm là nghề truyền thống ngày xưa	🟥	🟩	chị khăn mèo , thôn ti nê , xã a bung nói rằng , dệt thổ cẩm là nghề truyền thống ngày xưa
	Teencode - Slang - Lingo	Nằm ở vị trí thuận lợi cho việc du lịch nghỉ dưỡng ở **Đăk Lăk**, khách du lịch nếu ở đây có thể đi bộ để tham quan các địa điểm như Đài tưởng niệm Bác Hồ	🟥	🟩	nằm ở vị trí thuận lợi cho việc du lịch nghỉ dưỡng ở đắc lắc , khách du lịch nếu ở đây có thể đi bộ để tham quan các địa điểm như đài tưởng niệm bác hồ
	Teencode - Slang - Lingo	sản phẩm thu được là 20 lít rượu thóc ngon, ông **Nếnh** chia sẻ.	🟩	🟩	sản phẩm thu được là hai mươi lít rượu thóc ngon , ông nến chia sẻ .
	Teencode - Slang - Lingo	Những dòng tin nhắn gây **bưc** xúc cộng đồng mạng của chàng trai	🟩	🟩	những dòng tin nhắn gây bức xúc cộng đồng mạng của chàng trai

	Teencode - Slang - Lingo	Đá không bị **oxy** hóa, không chịu sự ăn mòn của muối và các loại **axit**	🟩	🟩	đá không bị ô xi hóa , không chịu sự ăn mòn của muối và các loại a xít
	Teencode - Slang - Lingo	Liên đoàn lao động thành phố phối hợp với Trung tâm văn hóa thành phố **Pleiku** đã tổ chức khai mạc giải bóng chuyền hơi nữ công nhân viên chức	🟨	🟩	liên đoàn lao động thành phố phối hợp với trung tâm văn hóa thành phố bờ lây cu đã tổ chức khai mạc giải bóng chuyền hơi nữ công nhân viên chức
	Teencode - Slang - Lingo	ô tô do anh Phan Thanh Nhật, 31 tuổi, trú **xã Ea H’leo**, điều khiển lưu thông theo hướng Gia Lai.	🟨	🟩	ô tô do anh phan thanh nhật , ba mươi mốt tuổi , trú xã e a hờ leo , điều khiển lưu thông theo hướng gia lai .
	Teencode - Slang - Lingo	Hoa hậu Hoàn vũ **H'Hen Niê** đi chân trần, phụ giúp bố mẹ công việc đồng áng ở quê nhà.	🟥	🟩	hoa hậu hoàn vũ hờ hen ni ê đi chân trần , phụ giúp bố mẹ công việc đồng áng ở quê nhà .
5	Symbol - Special character	em Việt xin trở lại trường và có **“**khiếu nại” tại Bệnh viện Quân y Trung ương **(**HCM), rất phức tạp...	🟩	🟩	em việt xin trở lại trường và có khiếu nại tại bệnh viện quân y trung ương hồ chí minh , rất phức tạp
	Symbol - Special character	một phần trong cuộc họp Nhóm làm việc chiến lược chung Nga**-**Thổ	🟩	🟩	một phần trong cuộc họp nhóm làm việc chiến lược chung nga thổ
	Symbol - Special character	Vì thế, tôi làm hài tết với mục đích chủ yếu để tìm niềm vui, không bị lão hoá**…** chứ không “nghỉ đông”	🟩	🟩	vì thế , tôi làm hài tết với mục đích chủ yếu để tìm niềm vui , không bị lão hoá chứ không nghỉ đông
	Symbol - Special character	Nhưng tôi nghĩ khó ai có thể làm được hay hơn **ê-kíp** hiện tại đâu.	🟩	🟩	nhưng tôi nghĩ khó ai có thể làm được hay hơn ê kíp hiện tại đâu .
	Symbol - Special character	Trong đó, Khu du lịch sinh thái chiếm đến hơn 90% diện tích; Khu nghỉ dưỡng chiếm **40 ha (3,03%);...**	🟩	🟩	trong đó , khu du lịch sinh thái chiếm đến hơn chín mươi phần trăm diện tích , khu nghỉ dưỡng chiếm bốn mươi héc ta ba phẩy không ba phần trăm ,
	English - Proper	chưa xử lí			
	English - Proper	chưa xử lí			
SUMMARY					
			60	97	

LIST OF PUBLICATIONS

[1] Do Tri Nhan, Nguyen Minh Tri, and Cao Xuan Nam. Vietnamese speech synthesis with end-to-end model and text normalization. In *2020 7th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 179–184, 2020. doi: 10.1109/NICS51282.2020.9335905.

[2] Tri-Nhan Do, Minh-Tri Nguyen, Hai-Dang Nguyen, Minh-Triet Tran, and Xuan-Nam Cao. HCMUS at mediaeval 2020: Emotion classification using wavenet feature with specaugment and efficientnet. In Steven Hicks, Debesh Jha, Konstantin Pogorelov, Alba García Seco de Herrera, Dmitry Bogdanov, Pierre-Etienne Martin, Stelios Andreadis, Minh-Son Dao, Zhuoran Liu, José Vargas Quiros, Benjamin Kille, and Martha A. Larson, editors, *Working Notes Proceedings of the MediaEval 2020 Workshop, Online, 14-15 December 2020*, volume 2882 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2020. URL `http://ceur-ws.org/Vol-2882/paper49.pdf`.

Vietnamese Speech Synthesis with End-to-End Model and Text Normalization

Do Tri Nhan*‡, Nguyen Minh Tri*‡, Cao Xuan Nam†‡
*Advance Program in Computer Science
†Faculty of Information Technology
University of Science, Ho Chi Minh City, Vietnam
‡Vietnam National University, Ho Chi Minh City, Vietnam

Abstract—**Speech synthesis systems are now getting smarter and more natural thanks to the power of deep neural networks. However, each language has a different phonological and contextual characteristics, we have conducted experiments, statistics, and applied Vietnamese phonetics to improve speech synthesis systems based on Tacotron2 neural networks. Our methods achieve the accuracy of 97% in text normalization task, and the synthesized speeches with a MOS score of 3.97, asymptotic to 4.43 of the voices that are directly recorded. We also provide a library for standardizing Vietnamese text called Vinorm and a package that converts text into a phonetic format called Viphoneme, which is used as an input for end-to-end neural networks, make the synthesis process faster, more intelligent and natural than using character inputs.**

I. Introduction

Text-to-speech system (TTS) has many applications like generating audio from the text for news reading, producing music, or replacing people's voice in case that person loses their ability to speak. Google's translator has its TTS system that supports many languages and many news websites in Vietnam have supported the automatic TTS system so that readers can hear the news even though they are busy doing other things.

A. Text-to-Speech overview

Many methods have been used to generate natural speeches from texts like the Unit Selection method, Statistic based method, or the most modern technique - Deep Neural Network.

1) Unit Selection: Unit Selection is a concatenate synthesis method that focuses on synthesizing audio based on unit-level like character or phoneme [1]. A lookup table is generated based on a large dataset that has information on each unit like its frequency, duration, position and nearby units. This method can generate voices that are almost the same with human voices, however, more data are needed to generate more natural voices.

2) Statistic based model: Speech synthesis model based on statistics is also one of the most commonly used speech synthesis models. The most famous statistical model is Hidden Markov model [2], which is a statistical time series model that utilizes the speech results. The acoustic parameters created from HMM which are selected according to the language parameters are used to control vocoder. However nowadays, with the development of GPU performance, those statistical models are replaced by Deep Learning model.

3) Deep Neural Network: Deep Learning is a method that especially suitable for unstructured data like image, text and sound. The appearance of the CNN model boosts the performance of those Deep Neural Network based TTS models since we can extract more information and features from audio spectrogram. One of the advantage of this learning method is that it does not need expert knowledge, in contrast with the requirement of having a large amount of data [3].

4) End-to-End models: Due to the development of Deep Neural Network learning method, many TTS systems moved to use end-to-end models and gain significant improving results, such as Tacotron2 [4] and FastSpeech [5]. These systems do not use complex linguistic and acoustic features, they learn to produce audio directly from text, generate human-like speech using neural networks. The system first generates mel-spectrograms from texts and then using vocoder like WaveNet to generate audios. They are able to generate emotional, smooth and clean speech, works well on out-of-domain and complex words, learns pronunciations based on phrase semantics and robust to spelling errors [6].

B. Related Works: WaveGlow and Tacotron2:

A modern speech synthesis system consists of two main parts: mel-spectrogram generator and vocoder. In 2016, Wavenet was introduced, is a combination of wavelet and neural networks, this technique estimates waveform samples from given input feature vectors - mel-spectrogram in speech synthesis [7]. Wavenet is a vocoder, which improves the synthesis process better than previous techniques, but the weakness of wavenet is that sequential generation is too slow for production environments, leading to the introduction of a Flow-based and GAN-TTS approaches. Flow-based approaches can be mentioned as Parallel WaveNet [8], ClariNet [9], FloWaveNet [10], the most typical of which is WaveGlow [11].

With the mel-spectrogram generator, there are methods such as Feed-Forward Transformer [12] or Attention based [13], in which the most typical is Tacotron2 [4] and the latest is FastSpeech2 [14]. Tacotron2 includes a recurrent sequence-to-sequence feature prediction network that maps input text to

[†] Corresponding author. Email: cxnam@fit.hcmus.edu.vn

mel-scale spectrograms, with a highlight that is the attention mechanism.

In order to apply these advanced models to Vietnamese, we need to standardize the data as well as propose using phonetic-based instead of character-based approach as an input of the neural network for taking the advantages of the Vietnamese language.

In this paper, we use Tacotron2 [4] and WaveGlow [11] for end-to-end Vietnamese speech synthesis system. We have rewritten the frontend of the speech synthesis system, we provide two python libraries:

Vinorm[1] to standardize text such as numeric characters, or abbreviations, local slang, and Viphoneme[2] to convert Vietnamese to grapheme format and from grapheme to International Phonetics Alphabet (IPA).

The rest of the paper is organized as follows: Section 2 reveals some of the major proposed methods on text normalization, text phonetization and speech synthesis with Tacotron2 model. Section 3 introduces the authors experiments on training and evaluation of our proposal and statistics about Vietnamese syllables usage in current online newspapers. Section 4 presents the conclusion and discusses some future works.

II. Proposed Methods

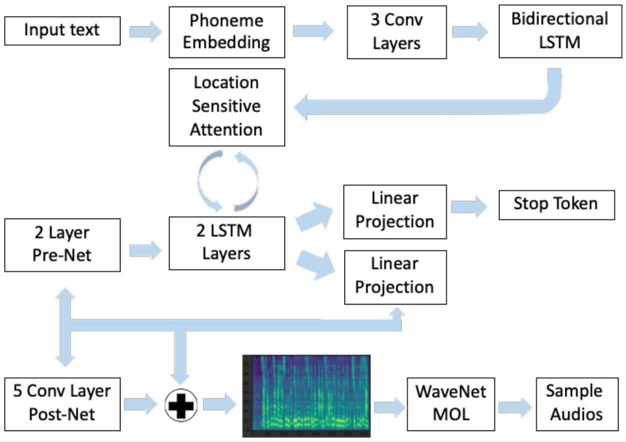


Figure 1: Tacotron2 architecture.

A. Vinorm: Text Normalization

Text Normalization is an important step in Text-to-Speech systems, helping to filter noise and making the input to be consistent with only Vietnamese syllables. The main task of the front-end of the TTS system is to standardize the text for the back-end system, the input is the raw text, we need to decide how to verbalize non-standard words, convert numbers, abbreviations, and words that cannot be pronounced into syllables, including dots, commas [15]. Every language needs different normalization processing methods because this problem is language-dependent [16]. It is impossible to build a complete text normalization because the language is ambiguous and evolves over time [17].

Text Normalization of Vietnamese Speech Synthesis today is still building grammars by hand instead of using automatic inference from large corpora because it has been the lack of annotated data [18]. To standardize text into readable words, the TTS system process through two steps, Rule-based and Dictionary-Checking.

1) Rules with Regular Expression: By using the Regular Expression to catch patterns that need normalization, which appears frequently in the language, containing numbers and characters, then replaced by syllables found in the dictionary. The output of this step is the paragraph without any digital characters.

Compared to the old VOS-frontend system [19], we do not lower the entire input before processing, thus we can handle cases with different pronunciation for uppercase and lowercase of the same words, and create a premise for backend processing when the capitalized words will be emphasized more through speech synthesis step.

We propose a new set of rules that are more systematic and general, scalable for future works. Based on the different contextual characteristics, rules are divided into four main categories to handle cases need standardization, including Special case, Timedate, Address and Mathematical. The patterns in each rule set will be proceeded one by one, browse through the entire text to match text need normalization, then return the corresponding normalized string

Special cases rules capture cases that are out of context, with specific formats, including Phone number, Football, Website, Email, etc.

For phone numbers, the identifying feature is a sequence of numbers starting with 0 or a plus sign, the number of digits from 10 to 14 digits. Phone numbers in each country will have a different way of writing, and in fact, there will be different formats, so the three types of phone number rules are listed, the pattern for capturing hotline numbers is also included. Website patterns have identifiable characteristics with a prefix is http(s), www, ftp or suffix is popular domains. Common email pattern is used for matching then spell each letter and symbol by English letter. Sport patterns include specific formats such as lineups, scores and hyphen-minus and dot is not spelled.

Time-date are rules for capturing phrases that show date and time elements. With the time indicating hours, minutes, and seconds, we identify by signals such as "h, g" or suffixed by AM / PM. These rules need to ensure the validity of time, if time is invalid, the system keeps them for later processing. If "-" followed by captured regex, it is read as "đến". Date patterns have a form such as DD / MM / YYYY or with the prefix “Ngày, sáng, trưa, chiều, tối, đêm, hôm, etc”. In addition, the timedate rules also capture and handle "FROM-TO" pattern cases.

Mathematical patterns capture cases containing normal number, floating Point Number, roman numerals, mathemati-

[1] https://pypi.org/project/vinorm
[2] https://pypi.org/project/viphoneme

cal expression or unit of measurement.

With normal number, we implement a function to convert numbers to letters. However, the normal number not only consists of successive digits, but also has a way of writing that splitting them into groups of 3 numbers, separated by commas, dot or space (e.g 12,000,000). This style of writing leads to confusion with floating point numbers, so to avoid false standardization, these cases will be matched first, then floating number, and finally normal number. For strings longer than 15 characters, each digit is converted individually, if the normalized text of number is too large, the string is be separated by commas to read more fluently. If there is an "- +" in front of the string, it is replaced with "cộng, “trừ”, except in the case where the number stands at the beginning of a sentence, "+ -" is treated as a bullet symbol. We categorized into two types of floating point, dot or comma. Floating point is replaced by "phẩy", the integer part is read as normal number and discard frontier zero, with fractional part, frontier zero is replaced with "không" and the following numbers read as normal number.

Unit of measurement are terms that refer to quantities, percentage or currency, followed by numbers, which can be an alphabetic or symbols. Because the system does not lower the input text, it can correctly standardize for upper and lowercase units (e.g Mbps and MBps). Units dictionary distinguish the upper and lower case. Besides the usual units, the patterns also need to capture "Unit/Unit" formats, matching strings are checked in the unit dictionary for readable words, if the matching is not in this list, we return the origin matching, keep for later processing. When it is sure that both sides of the slash are units, the "/" is replaced by "trên", this will avoid confusion in the case as "nam/nữ". Unit of measurement is also captured in FROM-TO pattern, there are two common types e.g: "10-20 km/h" and "10 km/h - 20 km/h", our system make sure not to be confused with cases where "-" is read as "minus".

For Roman numerals, to ensure accuracy does not fail in capturing, comparing to the old VOS, we just consider uppercase [X, I, V] is able to be Roman numerals, [L, C, D, M] are also Roman characters, but they rarely appear in the text, sometimes even leading to confusion with acronyms. The Roman numerals also have a FROM-TO structure, such as "XVI-XXI", but it is easier to handle and more consistent than the unit of measurement. The matching is checked for correctness by converting it into a decimal number, then converting the result back into roman form to compare with the original matching, then the matching is converted to normalized text.

Address-Code are rules for capturing phrases about addresses, locations, codes and all phrases containing numbers. Codenumber pattern match all remain cases with numbers, the matching sequence will be trim punctuation at both sides and separated into each alphanumeric substrings (e.g: MH370 is split into MH and 370). For each substring, if it is a fully capitalized letter sequence, it will be transcribed, if it contains lowercase letter, the system will keep that letter clusters and add space separate for each cluster. If the substring is numeric and the length of continuous number string is greater than 4, each number will be transcribed, otherwise, it will read the whole number as a normal number. Cases include special character or symbol will be mapped with corresponding phonetics, but with "-" is not spelled.

2) Dictionary Checking: After running through the rules sets, the string just contains letters, space and special character. This string will be split into many segments which separated by spaces, each token containing no space and no special characters before and after the string. The system runs over each token to validate if it is readable by checking it in Dictionary Vietnamese syllable which includes 7698 syllables. If the token does not exist, we look it up in the mapping dictionary instead, search and replace it with the corresponding word.

Abbreviations mapping dictionary includes 3 different mapping methods. With acronyms such as "NSƯT, GDĐT", we have to normalize to its original form. The second type is initialisms, including words like STEM and UNESCO, so we have to transcribe it. The last type is the acronyms that need to spell each letter such as PNJ, FPT, AFF, we do not handle it in this step and consider it as unknown to limit misunderstandings when not sure.

With dictionary mapping abbreviations, uppercase and lowercase tokens are often referred as two separate objects, with different contexts and probabilities, an acronym can have more than one meaning. Acronyms that appear less frequently are filtered out to limit misunderstandings

Teen Code - Slang - Lingo mapping dictionary is updated by adding more than 300 slangs like "H'Hen Nie, Ea H'leo", added some lingo that not appear in Popular Dictionary, words that backend doesn't support like “Đắk Lắk, Pleiku”. It also handles inconsistency in writing of the same word, such as "thuỷ - thủy" or "tuỳ - tùy". Loanwords like "oxy, axit" and common misspellings are also updated.

Special Symbols After browsing through the dictionaries, if the token does not belong to any Mapping Dictionaries, we continue to split the token into smaller substring separated by symbols and special characters. The system shows the corresponding reading for each symbol, non-stop punctuation such as brackets, quotes are removed. The system continues checking each substring in the mapping dictionaries again, this process (tokenize strings with space first, if the token is unknown, then tokenize with special character) avoids errors when the string is written adjacent to each other. This way will handle both cases “GD-ĐT” and “ê-kíp”, hyphen in the first case will be omitted to "GDĐT", the second case will be replaced with space to "ê kíp", and many similar cases with symbolic problems are also solved. If the token is still not in any mapping dictionary, we treat it as unknown word, if it is uppercase, spell each character as English letter, if it is lowercase and containing vowels, we will leave the backend to handle by letter to sound, if it does not contain the vowel, spell each character as Vietnamese letter.

Punctuations: The final step is to standardize the output text, including removing duplicate white spaces, handling

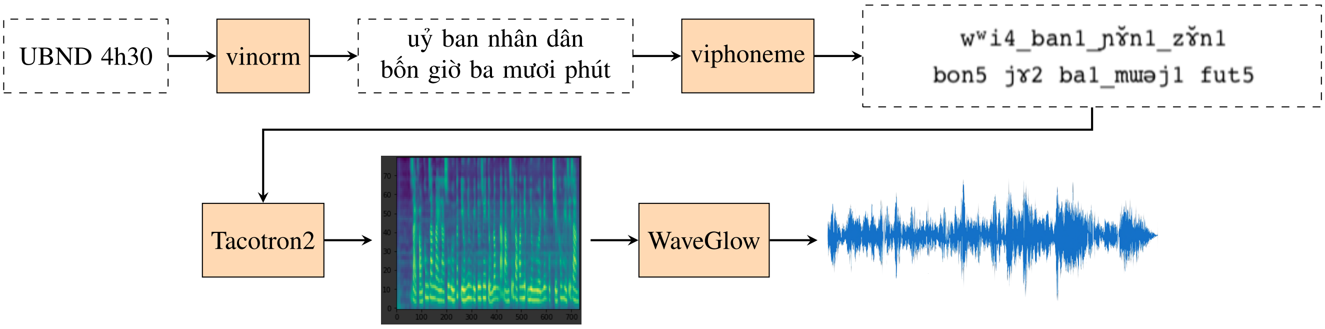


Figure 2: Proposal method pipeline.

punctuations including removing no voice marks like “()[]", and replace all punctuation marks with only comma and dot, which represent as the sound unit.

B. Viphoneme: Text Phonetizer

In order to synthesize words that have never appeared in the train set or out of vocabulary words (OOV), we now use the grapheme instead of the character as the input for the end-to-end model. This makes the model converges faster.

In order to represent mix-codes, foreign languages, languages need to have a uniform form of representation such as IPA or ARPABET. We have replaced the grapheme representation symbols with the IPA representation, which is both more concise than the ARPABET format, which many languages can be converted into.

Because IPA is for describing sound, we not only create general lexicon for Vietnamese but it also depends on the speaker's own region in the train (dialect). The presentation of IPA for Vietnamese has many ways, and is still not unified. We refer to the method of Pham 2006 [20], customize the way some phonemes represent and some labiovelar on-glide. Because IPA does not display tones, we have signed the blanks, grave, acute, hook, tilde, dot accents with the numbers from 1 to 6. The output will be in the following format:

$$(C1)(w)V(G|C2)+T$$

With C1 is initial consonant onset, w is labiovelar on-glide, V is vowel nucleus, G is off-glide coda, C2 is final consonant coda and T is tone. For example, the word "xuống" is parsed into structure as above "x-u-ô-ng-2", then these grapheme are replaced with IPA symbols.

Some special cases when converting from raw text to phoneme format such as the unification of the position of tones in words or the elimination of words, e.g: quyết -> qu-uyê-t-2. We then ran through the list of syllables in Vietnamese to make sure that all the words were unique.

C. Audio Processing and Model Configuration:

To be able to generate speech as closely as possible to human voices and match Wavenet's input, we reduce the sampling rate of each input audios data from 44100 Hz to 22050 Hz by using FFmpeg library to ensure the audio sample rate changes but keeps the speech rate unchanged. We remove silence at the start and the end, then adding one second silence to the end of each audio in order to help the model to recognize the end of the sentence better.

Finally, we use the vinorm and viphoneme package as mentioned above to change our text from normal Vietnamese characters to Vietnamese phonemes in IPA form. There are a total of 144 IPA characters including tones, Vietnamese phoneme, English phoneme, dot, comma, and other special characters, each IPA character will be mapped corresponding to a number. Therefore the input text will be converted to a sequence of numbers and this sequence will be the input for the embedding layer.

Our model almost the same with the vanilla Tacotron2 model, with changing from Character Embedding to Phoneme embedding, based on our suggested text normalization method.

III. Experiment

A. Vietnamese Syllables Statistics

In order to update the Vietnamese syllables used today, we proceeded to build a News Corpus with sources of 9 online newspapers, crawled from 7/2018. The Corpus is divided at the sentence level, consisting of a total of 6,308,173 sentences, the number of unnormalized sentences is 3,740,507 sentences, accounting for more than 59.29 %.

dantri	danviet	nld	thanhnien	tto
653545	386444	103218	634974	177287
tuoitre	vnexpress	vnn	zingnews	
167162	157202	481566	979109	

Table I: Number of sentences from popular Vietnamese newspapers

We continue to word-tokenize every sentence, totaling 10.88 million tokens. The tokens will be classified into groups such as Special Case, Time-date, Math and Number, English, Slang, Teencode, Acronyms, Initialism, Proper Noun, etc.

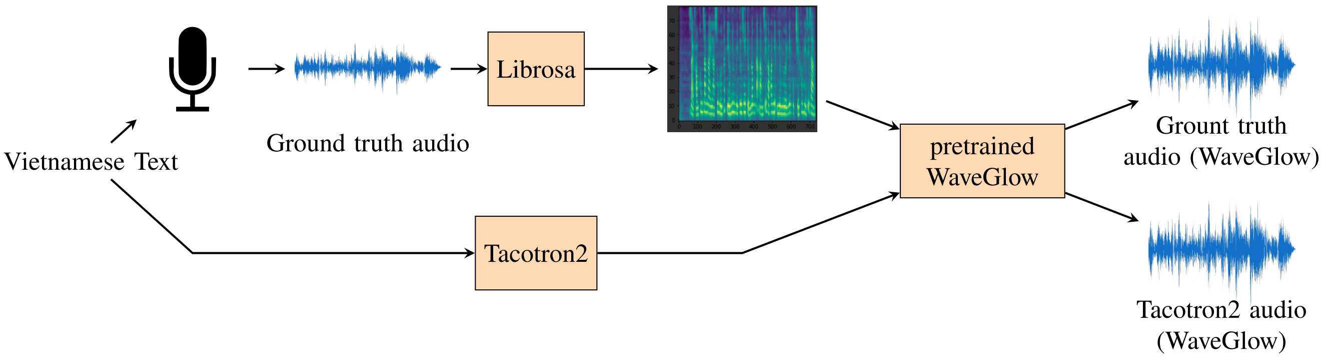


Figure 3: Ground truth (WaveGlow) generating process.

Category	Percent	Number of sentences
Special	0.2	21383
Timedata	1.21	131341
Math	22.67	2467113
English	35.65	3880214
Acronyms Lowercase	0.01	830
Acronyms Uppercase	6.31	686907
Teencode	0.01	944
Intialism	9.07	987432
Proper	9.17	998180
Other	15.71	1709682

Table II: Statistics on Tokens need to be standardized in Vietnamese

Some notable points from the statistics on News Corpus are: special cases only account for 0.2%, most of the Timedate cases are the publication date of the news, English accounts for 40 percent, and most of the abbreviations are all in uppercase.

B. Vietnamese Normalization and Phonetization

To select the language for the text normalization problem, we decided to choose C ++ because it is suitable for Cross-Platform Deployment, has a fast running time, uses less memory than Perl and is compatible with the current VOS backend. Some famous frameworks for text to speech systems also use C and C ++ such as Festival Speech Synthesis System of University of Edinburgh [21], Flite of CMU [22], Hts-engine use for Jtalk, Sinsy, and eSpeak is also written in C and C ++ [23].

One of the problems when using C ++ is to handle Vietnamese Unicode, we use ICU4C library version 64.2, an International Components for Unicode. This library is opensource, well-documented, robust and reliable. The ICU provides basic regular expression operators and especially Case Insensitive Matching, which helps the regular expression to capture both uppercase and lowercase letters, preserving the properties of the input text, which will be beneficial for handling in back-end steps, helping voice more natural, change the overall intonation like stress on capitalized words.

We provide a python package on Ubuntu 18.04 that can be installed at the Python Package Index called ViNorm.

From the data collected as mentioned above, we extracted 100 tricky need-normalized cases[3] to use as the baseline for improvement, 500 random cases in practical contexts for testing our proposal. These test cases do not include normal sentences, foreign words and proper nouns. With the 500 test cases, we improved the frontend of VOS from 60% to 97%. Some cases are wrong when mapping acronyms due to its plurality, such as BTC, we can read as "Ban tổ chức", "Bộ tài chính", or it can also be a stock symbol.

Method	Accuracy
Front-end VOS 2.0	60%
Updated Front-end VOS	97%

C. Speech synthesis

1) Dataset: The dataset used in this experiment is provided by InfoRe Jsc, which is also the Big Corpus set in International Workshop on Vietnamese Language and Speech Processing (VLSP) 2019 [24]. The dataset included about 22 hours with 13,462 utterances of north-accent female Vietnamese. Because the data set contains lots of noisy audio, we filtered out and removed more than 2000 samples, many of samples that the reader stopped in the wrong place also affect the training process.

2) Training: We train the model with Nvidia Quadro k6000 and use a batch size of 32, with learning rate is 10^{-5}. We run the model with 200 epochs and it converges after the 134000^{th} iteration. The total amount of training is about 240 hours, approximately 10 days.

Some audio parameters for training models such as filter length are 1024, hop length is 256, window length is 1024, number of mel channels is 80.

3) Results and Evaluation: Since we don't train WaveGlow model, we use the pretrained of WaveGlow provided by NVIDIA, which is able to generate good results on both English and Vietnamese.

[3] https://github.com/NoahDrisort/NICS_Appendix

To prevent the result from being the maximum decoder step, we decrease the gate_threshold parameter to 0.05 so that the result will have appropriate time length. The resulted audios still contain a lot of noise so we reduce them by using the noise reduction API and then, increase the sound level.

To evaluate the Tacotron2 model when applied to Vietnamese, without being dependent on vocoder waveglow, ground truth audios for testing are converted into mel-spectrograms and then converted these mel-spectrograms back to audios by using pre-trained WaveGlow as shown in fig. 3. This processed ground truth is called Groundtruth (Mel + WaveGlow), they will be compared with voices synthesized and standardized by our model.

To evaluate the result, we choose the MOS (mean opinion score) scoring system on the test set to check the quality of audios [25]. Each person who joins the survey listens to 40 audios, which include 20 audios generated from Tacotron2 and 20 corresponding ground truth audio generate from the process. They were asked to grade from 5 to 1, based on how natural and smooth those speeches compare to real human speeches. The final score of each audio type will be equal to the total score divides by the number of survey participants. Below is the MOS result gain from the survey of at least 20 people.

Model	MOS
Tacotron2 (WaveGlow)	3.97
Groundtruth (Mel + WaveGlow)	4.43

IV. Conclusion

Through statistics, we have given an overview of the standardized text of a Vietnamese speech synthesis system, the text normalization proposals for TTS systems were introduced and packaged in Pypi is called Vinorm. In this package, we improve the numeric processing modules, the processing special character module to produce the result to match the context of document. Larger dictionaries are used to cover more words and implementing module to recognize words which have different pronunciation in uppercase and lowercase. Updating for the regular expression step and the expansion of mapping dictionaries have helped VOS solve many special cases, especially for tokens that contain both numbers and letters. The results when compared to the VOS 2.0 text standardizer were superior when handling specific cases. Continuing with this result, the Viphoneme package was introduced as a bridge that converts Vietnamese text input into the input sequence of the Tacotron2, an end-to-end neural network model. The output has natural and fluent voice, quite similar to the real voice used for training with a MOS score of 3.97. This result opens up new directions in our research with a modern end-to-end model in Vietnamese, and it is possible to improve the result with a clean dataset with fewer noise samples.

Acknowledgement

This work is supported by AILAB and research funding from Advanced Program in Computer Science, University of Science, Vietnam National University - Ho Chi Minh City

References

[1] S. Kayte, M. Mundada, and C. Kayte, "A review of unit selection speech synthesis," vol. 5, p. 5, 11 2015.

[2] S. Kayte, M. Mundada, and J. Gujrathi, "Hidden markov model based speech synthesis: A review," *International Journal of Computer Applications*, vol. 130, pp. 975–8887, 12 2015.

[3] Y. Ning, S. He, Z. Wu, C. Xing, and L.-J. Zhang, "A review of deep learning based speech synthesis," *Applied Sciences*, vol. 9, p. 4050, 09 2019.

[4] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," 2017.

[5] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," 2019.

[6] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," 2017.

[7] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.

[8] A. van den Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. van den Driessche, E. Lockhart, L. C. Cobo, F. Stimberg, N. Casagrande, D. Grewe, S. Noury, S. Dieleman, E. Elsen, N. Kalchbrenner, H. Zen, A. Graves, H. King, T. Walters, D. Belov, and D. Hassabis, "Parallel wavenet: Fast high-fidelity speech synthesis," 2017.

[9] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," 2018.

[10] S. Kim, S. gil Lee, J. Song, J. Kim, and S. Yoon, "Flowavenet : A generative flow for raw audio," 2018.

[11] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," 2018.

[12] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6706–6713, 07 2019.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[14] 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," 2020.

[15] R. Sproat, A. W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards. (2001) Normalization of non-standard words.

[16] M. Chu, H. Peng, and Y. Zhao. (2009, Feb. 24) Front-end architecture for a multi-lingual text-to-speech system. US Patent 7,496,498.

[17] D. Yarowsky. (1993) Text normalization and ambiguity resolution in speech synthesis.

[18] R. Sproat. (2010) Lightly supervised learning of text normalization: Russian number names.

[19] V. Q. D. Ha, N. M. Tuan, C. X. Nam, P. M. Nhut, and V. H. Quan. (2010) Vos: the corpus-based etnamese text-to-speech system.

[20] A. H. Pham, "Vietnamese rhyme," *Southwest Journal of Linguistics*, vol. 25, pp. 107–142, 01 2006.

[21] A. W. Black, P. Taylor, and R. Caley, "The festival speech synthesis system: system documentation," 1997.

[22] M. H. Lee, "Migrating dari clustergen flite text-to-speech voice from desktop to android," 2014.

[23] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The hmm-based speech synthesis system (hts) version 2.0," *Proc. of ISCA SSW6*, 09 0002.

[24] N. T. T. Trang and N. X. Tung, "Text-to-speech shared task in vlsp campaign 2019: Evaluating vietnamese speech synthesis on common datasets."

[25] R. C. Streijl, S. Winkler, and D. S. Hands. (2016) Mean opinion score (mos) revisited: methods and applications, limitations and alternatives.

HCMUS at MediaEval 2020: Emotion Classification Using Wavenet Features with SpecAugment and EfficientNet

Tri-Nhan Do[1,3], Minh-Tri Nguyen[1,3],
Hai-Dang Nguyen[1,3], Minh-Triet Tran[1,2,3], Xuan-Nam Cao[1,3]
[1]University of Science, VNU-HCM
[2]John von Neumann Institute, VNU-HCM
[3]Vietnam National University, Ho Chi Minh city, Vietnam
{dtnhan,nmtri17}@apcs.vn,nhdang@selab.hcmus.edu.vn,{tmtriet,cxnam}@fit.hcmus.edu.vn

ABSTRACT

MediaEval 2020 provided a subset of the MTG-Jamendo dataset, aimed to recognize mood and theme in music. Team HCMUS proposes several solutions to build efficient classifiers to solve this problem. In addition to the mel-spectrogram features, new features extracted from the wavenet model is extracted and utilized to train the EfficientNet model. As evaluated by the jury, our best result achieved of 0.142 in PR-AUC and 0.76 in the ROC-AUC measurement. With fast training and lightweight features, our proposed methods are potential to work well with deeper neural networks.

1 INTRODUCTION

Emotions and Themes in Music task in MediaEval [1] is difficult and challenging due to the ambiguity of tags in the real world. Mood is often influenced by human perception, different people will have different feelings, moreover, this is a multi-class classification problem with more than 56 tags. The dataset is pretty unbalanced in the distribution of mood labels, each audio music is composed of multi-labels that there can be many emotions in the same song.

To be able to solve this task, the authors have tried many methods, using many kinds of models, input features or loss functions. Our best result is an ensemble of two kinds of different methods, one using provided mel-spectrogram features with EfficientNet model and the other using waveNet features with MobileNetV2 model [7, 9].

2 RELATED WORK

Data augmentation is important when training neural network model. Traditional audio augmentation methods try to modify the speed of the waveforms or alter the original signal samples with noises, this method need much computational cost. With SpecAugment approach[6], they adjust the spectrogram by warping it in the time direction, masking blocks of consecutive frequency channels, and masking blocks of utterances in time. This approach is more simple, cost less time and resources.

WaveNet model is applicable in many problem of signal processing, time series forecasting and music generation[4]. Therefore, the authors also try following this approach by using a pre-trained WaveNet model to extract feature vectors from raw audio and then, using those features as inputs for convolutional neural networks.

3 APPROACH

We follow many approaches which include two main inputs: mel-spectrogram features and wavenet features.

3.1 Data analysis

As in the figure, the green part shows the audio with only one label mood/theme, the yellow part shows the audio with 2 to 3 moods, the red part shows the audio with more than 3 moods. Number of sample audio for training is 9949 with a total of 17885 moods. On average, each class will have 319 audio with a standard deviation of 202.75. The maximum number of moods of an audio is 8. Mood / theme that appears most is happy with 927 audios.

We can see that the data is extremely unbalanced, and some classes have no audio representing it entirely. Therefore, it is necessary to have a way to reduce the complexity of the data.

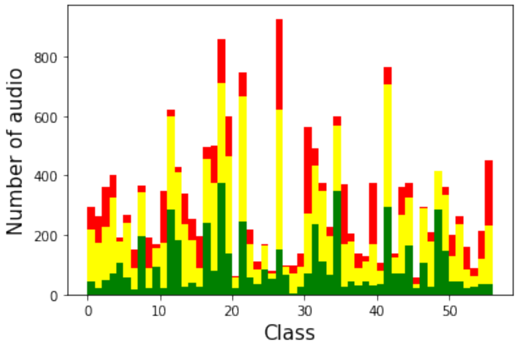


Figure 1: Histogram of mood and theme of training set

3.2 Data preprocessing

3.2.1 Data balance: To reduce the ambiguity of the data, the authors try to change each audio's label from multi-label to single label, keeping the most significant tag of each audio, reduce standard deviation, give preference to moods with little data.

3.2.2 Features preprocessing: **Wavenet feature**: Based on the idea of using wavenet as classifier for raw waveform music audio [5, 10], the authors use WaveNet-style autoencoder model that conditions an autoregressive decoder on temporal codes learned from the raw audio waveform, this model was pretrained from high-quality dataset of musical notes Nsynth [2].

Based on the dataset's statistic, the minimum length of audio is 30 seconds and based on the limitation of the authors' training machine, sound samples greater than 400 seconds in length will be

MediaEval'20, December 14-15 2020, Online

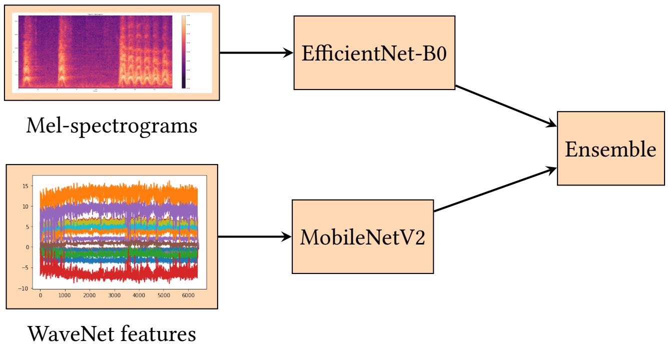


Figure 2: Overview of submission 1.

trimmed to take the middle part. Each sample is again randomly cut for 30 seconds and then extract features from them. This approach is quite subjective and causes loss of input data, we planned to experiment with random cutting from 400 seconds of audios after each epoch. The output of a 30 seconds audio is 16 frames multiply with 937-time steps.

Mel-spectrogram: Each sample feature has 96 channels and time frames are randomly cropped to 6950 after each epoch.

3.3 Data augmentation

SpecAugment: To train models more efficiently, the authors include segmentation method SpecAugment which was introduced by Google. This method masks blocks of consecutive time steps and channels in each mel-spectrogram. The result when using this method is increased significantly, PR-AUC-macro is improved from 0.134 to 0.139.

Each input have 70% chance to be augmented by using SpecAugment, each mel-spectrogram will have two blocks of time masking and two blocks of channel masking.

3.4 Deep Neural Network model

Since both mel-spectrogram features and wavenet feature can be expressed as images, the authors use convolutional models such as MobileNet and EfficientNet. The mel-spectrogram features are passed to EfficientNet-B0, on the other hand, the waveNet features are passed to MobileNetV2 and EfficientNet-B7. Because waveNet features are not large enough to fit EfficientNet-B7, the authors duplicate the number of channels so that these kinds of features can be used.

In addition, we also tested the SVM model, InceptionNet, Resnet, and to capture the long-term temporal characteristics, self-attention was added as in the method of AMLAG 2019[8], but this method produce a slight improvement in the result.

3.5 Loss function

For the loss function, binary cross entropy loss (BCE) is applied for both MobileNet V2 and EfficientNet. The authors also try to apply Focal Loss[3] since the dataset is pretty unbalanced, however it does not gain better results on our dataset after the balance step.

4 EXPERIMENTS AND RESULTS

Our experiments are done on a computer server with Nvidia Quadro k6000 graphic card. Method A,B and D are not submitted to the challenge. We realize that data balancing method leads to a better result comparing to the original dataset with default labels. Based on the experiments on the validation set, our ensemble models are calculated with factors of 0.7 and 0.3 for mel-spectrogram features and wavenet features to gain the best results.

Method	Features and Model	PR-AUC-macro
A	Mel-spectrogram EfficientNet-B0	0.127
B	Mel-spectrogram EfficientNet-B0 with data processing	0.134
C (run2)	Mel-spectrogram EfficientNet-B0 using augmentation	0.139
D	WaveNet MobileNetV2	0.102
E (run3)	WaveNet EfficientNet-B7	0.105
F (run1)	Ensemble C and D	0.1413
G (run4)	Ensemble C and E	0.1414

Table 1: Experiment results

5 CONCLUSION AND FUTURE WORKS

The EfficientNet model was shown to be more efficient than previous models in the mood and theme classification problem. The results can be improved by training mel-spectrogram features on other more complex EfficientNet models.

Although the result when training on wavenet features is not higher than mel-spectrogram features, but when assembling two models using these features, the results are improved, it shows that wavenet can extract other aspects of the dataset. Because the wavenet features were extracted by using a pretrained model, the augmentation methods have not been fully applied, for the future work, there are still more improvements to come when training WaveNet-style autoencoder models on the Jamendo dataset.

ACKNOWLEDGMENTS

Research is supported with computing infrastructure by SELAB and AILAB, University of Science, Vietnam National University - Ho Chi Minh City.

REFERENCES

[1] Philip Tovstogan Minz Won Dmitry Bogdanov, Alastair Porter. 2020. MediaEval 2020: Emotion and theme recognition in music using Jamendo. In *MediaEval 2020 Workshop*.

[2] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. 2017. Neural audio synthesis of musical notes with wavenet autoencoders. In *International Conference on Machine Learning*. PMLR, 1068–1077.

[3] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*. 2980–2988.

[4] 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* (2016).

[5] Sandeep Kumar Pandey, HS Shekhawat, and SRM Prasanna. 2019. Emotion recognition from raw speech using wavenet. In *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*. IEEE, 1292–1297.

[6] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779* (2019).

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[8] Manoj Sukhavasi and Sainath Adapa. 2019. Music theme recognition using CNN and self-attention. *arXiv preprint arXiv:1911.07041* (2019).

[9] Mingxing Tan and Quoc V Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946* (2019).

[10] Xulong Zhang, Yongwei Gao, Yi Yu, and Wei Li. 2020. Music Artist Classification with WaveNet Classifier for Raw Waveform Audio Data. *arXiv preprint arXiv:2004.04371* (2020).