CROSS-PLATFORM CHATBOT

by

Shun Guo

Signature Work Product, in partial fulfillment of the
Duke Kunshan University Undergraduate Degree Program

June 3, 2022

Signature Work Program
Duke Kunshan University

APPROVALS

Mentor: Mustafa Misir, Division of Natural and Applied Sciences

---

Marcia B. France, Dean of Undergraduate Studies

CONTENTS

ABSTRACT

Abstract (English): In this project, I am going to apply deep learning approaches in Text-to-Speech (TTS) tasks to build a chatbot that can be deployed over different social network platforms. In my experiments, I trained two different existing TTS algorithms Tacotron 2 and Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech (VITS) using open corpus in English, Chinese, and Japanese. Different from evaluation processes in previous research, in this project, I applied a new deep learning based objective evaluation process, MOSNet, to evaluate the Mean Opinion Score (MOS), which was subjectively evaluated in past research. The evaluation shows VITS (MosNet=2.63±0.08/2.77±0.06) has a better performance than Tacotron 2 (2.56±0.035/2.77± 0.06). I also used Mel-cepstral disctance (MCD), another widely used objective evaluation method to compare the similarities between the inference data from both algorithm and the sample data. The results again verifies that VITS inferences share greater similarities with the real-human voice samples. Based on these evaluations, VITS based algorithm was applied in chatbot construction. The chatbot was built on an open-source cross-platform chatbot framework, Nonebot 2. The experiment of this project is running on QQ, a Chinese social platform. The responses text was generated from records, preset values, and open Natural Language Processing (NLP) tools such as chatGPT. The voice response is generated based on the text. This application shows the effectness of deep learning approach in improving chatbot user experiences and explores the future of human-machine interactions.

摘要（中文）：在这个项目中，我将基于文本语音转换（TTS）深度学习方法，建立一个

可以在不同社交网络平台上部署的聊天机器人。本项目使用了英文、中文和日文的公开语料库，训练了 Tacotron 2 和” 带有对抗性学习的条件变异自动编码器”（VITS）算法。区别于以往研究中的评估过程，这个项目应用了一个新的基于深度学习的客观评价体系 MOSNet 来评价过去研究中使用主观评价体系得出的平均意见得分（MOS）。评价结果显示，VITS（MosNet=2.63±0.08/2.77±0.06）比 Tacotron 2（2.56±0.035/2.77±0.06）的表现更好。本项目还使用了另一种广泛使用的客观评价方法 Mel-cepstral disctance（MCD）来比较两种算法的推理数据与样本数据之间的相似性。结果再次验证了 VITS 的推断与真实人类语音样本有更大的相似性。基于这些评估，VITS 算法被应用于本项目聊天机器人的构建。本项目中聊天机器人基于开源的跨平台聊天机器人框架 Nonebot 2 建立。本项目的实验是在中国的社交平台 QQ 上进行的。回复文本是由过去聊天记录、预设值和开放的自然语言处理（NLP）工具（如 chatGPT）生成的。语音由这些文本生成。这个项目展示了深度学习在改善聊天机器人用户体验方面的效果，并探索了人机交互的未来。

ACKNOWLEDGEMENTS

I want to give tribute to everyone helped me in this project. My mentor gave me great help in research direction, which built the basis for everything. The Nonebot and VITS community are also very helpful in this project. They pointed out my mistakes, helped me find the right way to clean my data, set parameters, and evaluate results. They also helped me a lot in solving technical problems in coding. I also want to thank my classmates who helped me find useful literature such as MosNet in this project. I appreciate their help throughout the project.

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The chatbot is an important application of Artificial Intelligence (AI) systems, which "integrates our daily lives with the creation and analysis of intelligent software and hardware" [1]. It is a computer program that mimics real-human responses by simulating the natural language and generating real-human voices. Chatbots have been utilized as interactive agents, digital assistants, and artificial conversation entities in education, business, and e-commerce. Alan Turing's article "Can machine think" [24] triggered people's interest in the idea of chatbots. The first chatbot is developed in 1966 for psychological treatment use[29]. It used a pattern matching mechanism and template-based pattern. A huge breakthrough in chatbot was made in 1995, by the chatbot ALICE [7]. It was developed with a pattern-matching algorithm. The commercial applications of chatbots surged in recent years, with the creation of virtual personal assistants like Apple Siri, Microsoft Cortana, etc.

Chatbots have promising potential in providing quick and convenient support for questions concerning productivity or entertainment. Machine learning algorithms, a new form of artificial intelligence that enables computer programs to improve from "reading" through data, empower the chatbot to give natural, reliable responses and further build connections with its users. The human likeness of chatbots is further improved by integrating visual and identity cues into them. Instead of simple text response, these integrated module empower the chatbot to give responses in various forms and therefore greatly improve the user friendliness, making them useful tools and inter-

esting companions in the virtual world. Following this clue, I initiated this project to integrate the latest Text-to-Speech (TTS) algorithms in a chatbot to further boot its human-likeness, which I believe will lead to better interaction experience.

In the realization of this goal, messaging frame work, which enables the chatbot to have basic information interactions with the users, plays a vital role. Messaging Framework is a system of programs that enables chatbots to retrieve, store, and send messages in various forms such as text, image, audio, or file. There have been many open or commercial messaging frameworks [19]. Rakesh and Manoj (2020) [19] present an analysis of an open-source chatbot framework RASA. Its basic architecture is demonstrated below in Figure 1.1. The message from the end user is fed to an interpreter. The tracker is used to maintain the conversation state and receive results from the message interpreter. The output from the tracker is given to the Policy module, which acts on the current state of the tracker. The Policy decides the action and the action is transformed into readable messages. The policy is the program defined by the chatbot developers and is the key part of the chatbot. The tracker will maintain the log of selected actions and update the conversation status. This framework gives a basic idea of the working process of chatbot messaging frameworks. There are also simpler frameworks such as Dialogflow provided by Google. Its vital part is only the user request and agent response and all other parts could be integration from other websites or applications. It is applied to Google Assistant and Amazon Alexa.

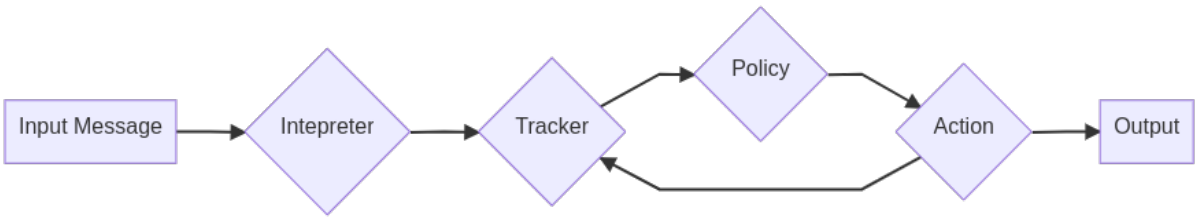


Figure 1.1: RASA messaging framework architecture.

Text-to-Speech (TTS) is a task in speech synthesis that converts written text into spoken language. TTS algorithms has experienced great development in past few decades. TTS technology has a various application in industries such as telephony and entertainment. The goal of TTS task is to create artificial voice that mimics human voice. This requires the algorithm to interpret the text as well as generate natural-sounding

speech. In recent years, deep learning, which applies multi-layers in neural networks in machine learning, have greatly improved the performance of TTS algorithms. The application of deep learning techniques such as recurrent neural networks (RNNs) and attention mechanism makes it possible for TTS algorithms to be applied in audio books and navigation systems.

In this project, I am going to explore different TTS systems in past research. Based on the experiments of past research, I will find different data set to conduct my own experiments. With the results, I will compare the performance of different systems based on objective evaluation metrics. Additionally, I will find ways to implement those systems in various language background and integrate different language models into a single chatbot. I will test the performance of the chatbot from different perspectives such as respond speed and respond naturalness. With these results, I will also explore the directions of future work.

Chapter 2

RELATED WORKS

In this project, the messaging framework used is another open-source messaging framework, Nonebot[30]. It was originally a framework developed especially for QQ, a Chinese social media platform developed by Tencent. Its compatibility is expanded with contributions from the community. Now it is compatible with multiple social media platforms such as WeChat and Telegram. This framework enables the development highly expandable chatbot. The Nonebot community also offers extensions that help build a powerful chatbot quickly. Its adapter structure enables it to work across platforms. As shown in Figure 2.1. below, the driver will communicate with different social platforms through their application programming interface (API) for chatbot developers. It will convert messages from different platforms into events, which will be further reported to the policy module. The policy module will determine its response to specific events. Specifically, it will first determine whether execute or not by permission rules and the matcher will determine the specific command executed. This framework is selected for this project as it is the only one that supports Chinese social platforms. Also, it is highly flexible as the policy module is completely defined by chatbot developers. This makes integrating Text-To-Speech (TTS) module in this chatbot possible.

Though there have been many chatbots across different social platforms, there are few discussions about applying the latest Text-to-speech (TTS) algorithms in chatbots on social platforms. Therefore, the main contribution of this project will be exploring the

possibility of using the latest TTS algorithms on chatbots over social platforms. This is a promising attempt as there have been many practices of applying TTS algorithms in voice assistants as mentioned above. I believe this field is worth exploring as a speaking chatbot can build closer connections with users and therefore create greater value.

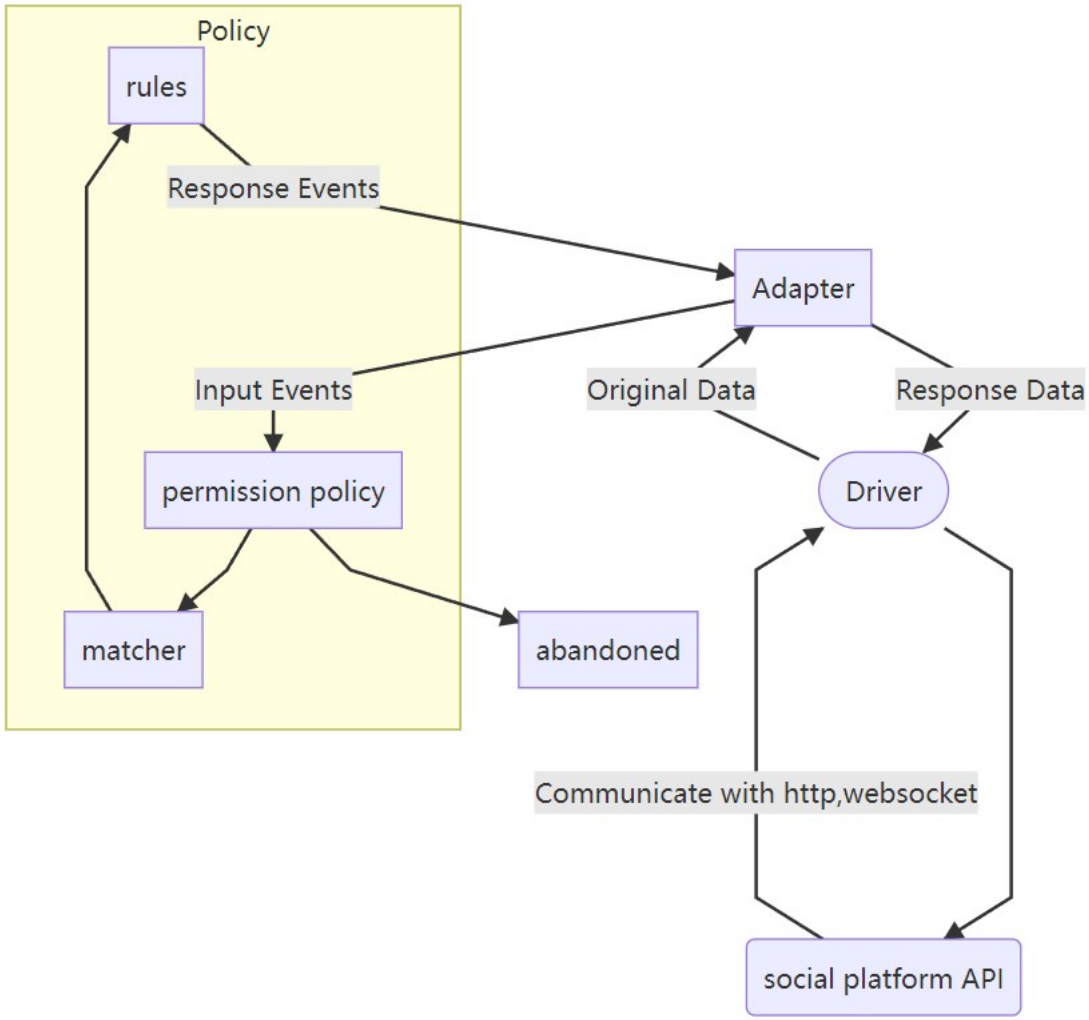


Figure 2.1: Nonebot chatbot framework architecture.

TTS task focuses on generating natural, real-human speech from texts. Dominating methods utilized in this field are introduced in the following part and a comparison of them is presented in Table 2.1. A common method used in evaluating TTS algorithms is the Mean Opinion Score (MOS)[27]. The opinion score is first proposed by International Telecommunication Union (ITU) as the ”value on a predefined scale that a subject assigns to his opinion of the performance of a system”[23]. MOS is the aver-

age of these scores across subjects. It is the most popular indicator of perceived media quality and is applied to measure the synthesized audio quality of the algorithm above. Both subjective and objective measurement tests are used to obtain MOS (referred to as subjective MOS and objective/predicted MOS respectively). The specific methods applied in designing subjective tests will be introduced in the next section. MOS has a range of from 0 to 5, where the real human speech without loss is between 4.5 to 4.8. Higher score in MOS implies better performance of TTS systems.

The first dominating method in this field is the unit selection method [8]. It becomes the main method for many years since published. This method selects units of speech from a large audio database, where unit selections are determined by context information. The units in the database are considered a state transition network, a type of graphical model used to represent the sequence of states and transitions in a system or process. Synthesized speech is produced by concatenating the waveform of the selected units. The speech database mentioned here should be a large database of units with different prosodic and spectral characteristics such that real human-like speech can be produced. The vital part of this method is the unit selection method (the "CHATR speech synthesis system"[8]). This system is based on two cost functions. Let t_i be the target representation, and u_i be the selected unit. $C^C(u_i, u_{i-1})$ is then defined as the concatenation cost and $C^t(u_i, t_i)$ is defined as the target cost. They are used to measure the quality of the join and the difference between the database unit and the target. The selected vectors of the unit will therefore generate a vector of costs. Their dot product with the weight vector gives the target and concatenation cost of the system. The total cost combines these costs and the cost of silence. The target of this task will be finding the vector of units that minimizes the total cost. The training process is based on regression training on the weight vector based on the cost functions.

The statistical parametric speech synthesis[31] is a following method that seeks to solve the problem of boundary unnaturalness in the synthesis of the preceding method. Synthesis from the parametric method usually sound less natural than the previous method. This method can be most simply described as "generating the average of some sets of similarly sounding speech segments"[31]. This method first extract parametric representations of speech from a database and model them using generative

models like Hidden Markov Model (HMM). The model parameters are estimated using the maximum likelihood criterion, or to find $\hat{\lambda} = \arg\max\{p(O|W,\lambda)\}$, where O is the training data, λ is the parameters and W is the corresponds word sequences. Then for a given word sequence, the method synthesize it by generating speech parameters $\hat{o}$ that have $\hat{o} = \arg\max_o\{p(o|w,\hat{\lambda})\}$. The speech is reconstructed from the parametric representations of speech.

Text to speech method started to produce audio with quality that rivals real human speech from the WaveNet[15] method. Different from previous methods, WaveNet generate raw waveform of audio signal directly. This enables it to model any kind of audio including music. This method is based on Pixel Recurrent Neural Networks (PixelRNN)[26], a neural network algorithm applied in Computer Vision tasks such as image completion. In essence, WaveNet is an autogressive model where each sample depends on the previous ones. Joining probability of the waveform is a product of conditional probabilities of previous steps, or mathematically $p_\theta(\mathbf{x}) = \prod_{t=1}^{T}(x_t|x_1,\cdots,x_{t-1})$. Each time the algorithm samples a value from the calculated distribution and feeds the value back as the generated prediction. The combined prediction is again used as the input. This process is repeated until the entire speech waveform is generated. This leads to the main shortcoming of this algorithm. Because the process is recursive, the generation process can be very slow an computationally expensive. The first version of WaveNet has a MOS of 4.21 in English language. Fast WaveNet[16] improves the cost of the WaveNet method. It reduces the time complexity from $O(2^L)$ to $O(L)$ with a caching system that stored previous calculations. The structure of WaveNet is presented below in Figure 2.2.

Deep voice[2] proposed by Baidu, a Chinese tech company, is the foundation of end-to-end speech synthesis. This algorithm applied 4 different neural networks that form a ”end-to-end pipeline”, a single network that gives output directly from input. A multilayer encoder-decoder model converts graphemes to phonemes. A hybrid of Convolutional Neural Network (CNN)[14] and Recurrent Neural Networks network (RNN)[11] is trained to predict the segmentation between vocal sounds, where CNN and RNN are neural networks that are widely applied in deep learning tasks for feature abstraction and pattern catching respectively.The phonemes duration and frequencies is pre-

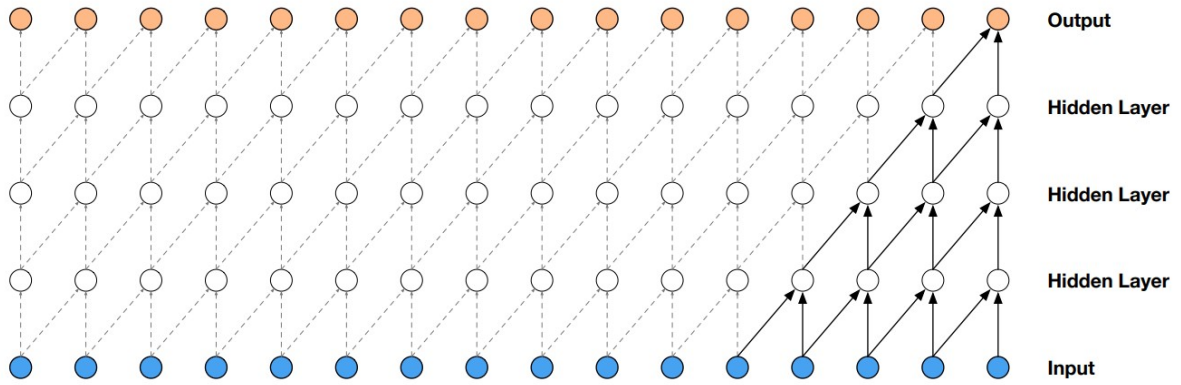


Figure 2.2: Architecture of the WaveNet [15].

dicted by two fully connected layers followed by two unidirectional Gated recurrent unit (GRU)[3] layers and another fully connected layer, where GRU is a flow-controlling mechanism applied on RNN. A WaveNet is applied to synthesize the final audio. Deep Voice achieves a MOS of 2.67 in English language. The second version of Deep Voice[5] added multi-speaker support and enlarged the original architecture to achieve better performance. Deep Voice 2 with an 80-layer WaveNet achieved a MOS of 3.53 in English. The third version of Deep voice is a complete redesign. Deep Voice 3[17] applied an encoder-decoder architecture. Encoder-decoder model is a way of using RNN for sequence-to-sequence problems. Typically the encoder in the pair transforms input information into compressed representations and the decoder generate output sequences step by step from these representations. In Deep Voice 3, the encoder is a Fully Convolutional Network (FCN)[13], a modified CNN that can preserve spatial information. The FCN transforms textual features into a compact representation. The decoder is another fully convolutional network that converts the learned representation into audio representation. An attention mechanism is applied to achieve this process. Attention mechanism is a common technique in machine learning that mimics cognitive attention to better emphasize the important part in the input information. It is widely applied in enhancing parts of the input data and diminish the rest. Deep Voice 3 achieves a MOS of 3.78.

Tacotron[28] is a TTS system proposed by Google in 2017. Its structure is similar to Deep Voice 3 as Deep Voice 3 is a modification of Tacotron. It uses a encoder-decoder model with attention mechanism. The structure of Tacotron is demonstrated in Fig-

ure 2.3. Its encoder has a pre-net and a CBHG module. The pre-net is a name for a series of non-linear transformation applied on the original input. The structure of the CHBG module is presented in Figure 2.4. CHBG consists of a 1-D convolution bank, a highway network [22] layer, and a Birectional GRU layer. A highway network is an architecture designed to ease gradient-based training of very deep networks. Its decoder has multiple pre-net, RNN layers, and CHBG module. It achieved a MOS of 3.82 in English.

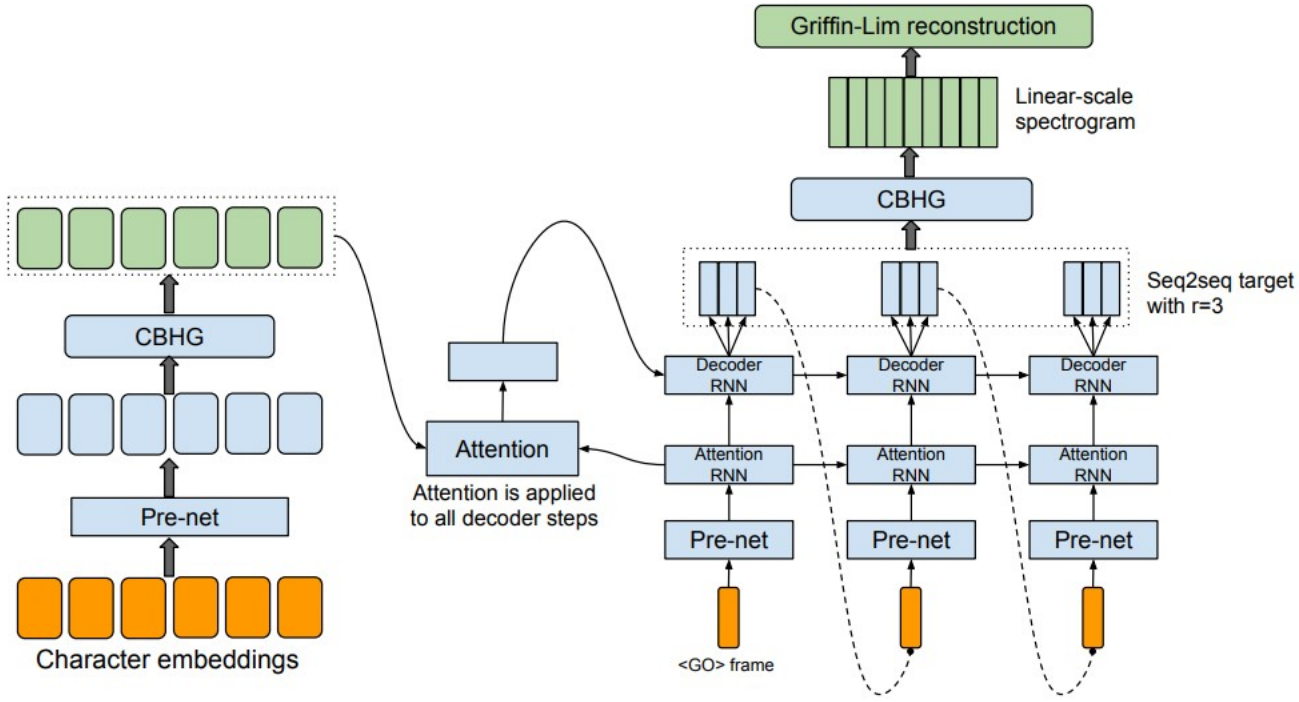


Figure 2.3: Architecture of the Tacotron 1 [28]

Tacotron 2 [20] improve and simplifies the original architecture with no major differences. Its structure is presented in Figure 2.5.In the encoder, 3 convolutional layers and a bidirectional LSTM replaces PreNets and CHBG modules. LSTM is a neural network stands for Long short-term memory. It is commonly applied to process sequences of data. The decoder has similar structure but simplified. While the original one has 6 RNN layer, 3 pre-net layer and 1 CBHG module, the simplified decoder has one RNN, 2 unidirectional LSTMs, and 5-layer Convolutional Post-Net. An improved wavenet based on PixelCNN++[18] is used as the Vocoder, and Tacotron 2 achieved a MOS of 4.53.

VITS[9] is the latest breakthrough in TTS task, which enables single-stage training in-

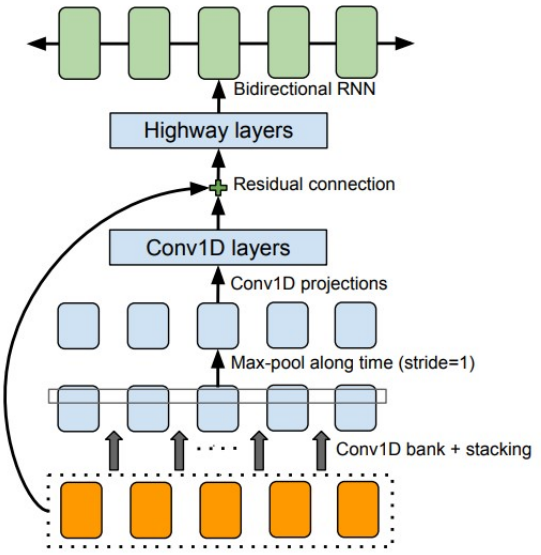


Figure 2.4: Architecture of CHBG in Tacotron [28]

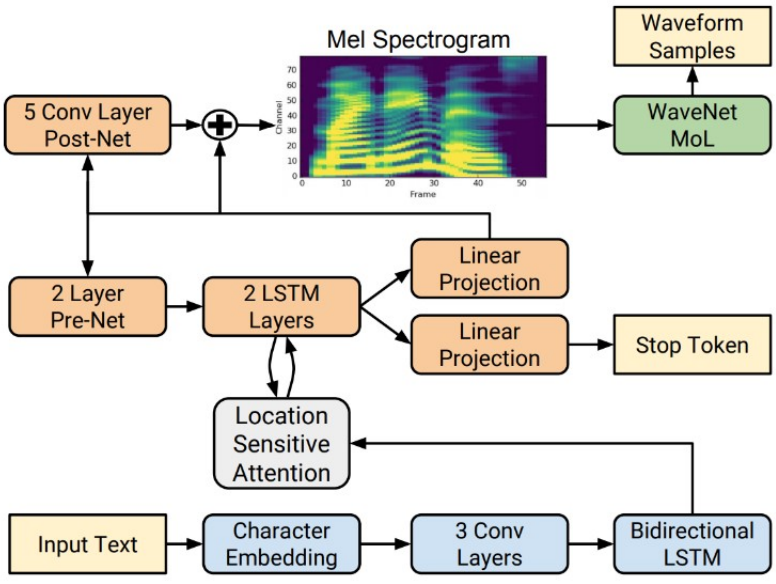


Figure 2.5: The architecture of Tacotron 2 [20]

stead of a parallel TTS system. Here single stage means it trains the middle producet of prediction, the Mel-spectrogram, together with the vocoder, which transforms the predicted waveform into audio signals. As a middle product, the Mel-spectrogram has been viewed as the final prediction since WaveNet. The vocoder is trained separately and in past research like Tacotron and Deep Voice, the performance of the vocoder is a controlled variable in comparisons. Though there have been many attempts of combination, VITS is the first single stage algorithm whose performance can rival parallel ones. The combination is achieved with a conditional variational autoencoder (cVAE). VAE is a machine learning model used to learn representation of high dimensional data such as image or audio, by mapping them in low dimensions. cVAE is a modified version of it. Its output is generated based on conditions on the input, such as label or category. VITS uses a transformer encoder and a HiFi-GAN V1[10] decoder. Its structure is shown in Figure 2.6. It achieves a MOS of 4.43.

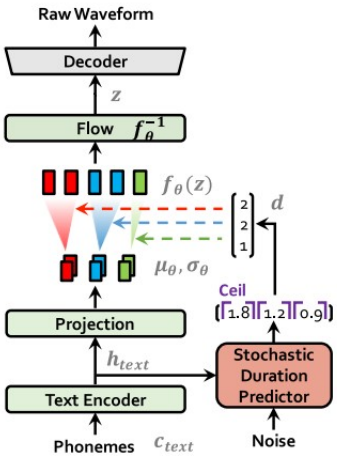


Figure 2.6: Architecture of the VITS

Name	Description	Publication
Unit Selection	Select units from existing corpus database	Hunt et.al in 1996 [8]
Statistical Parametric Synthesis	Hidden Markov Model to predict wave form	Zen et.al in 2009 [31]
Wavenet	Autoregressive PixelRNN	Oord et al. in 2016[15]
Fast Wavenet	Reduce the time complexity of the Wavenet from $O(2^L$ to $O(L)$	Paine et.al in 2016 [16]
Deep Voice	Multilayer GRU encoder-decoer+ hybrid of CNN and RNN+ WaveNet	Baidu in 2017[2]
Deep Voice 2	Same structure as Deep Voice with more wavenet layers	Baidu in 2017[5]
Deep Voice 3	FCN Encoder-Decoder	Google in 2017[17]
Tacotron	CHBG and RNN Encoder-Decoder	Google in 2017[28]
Tacotron 2	LSTM and RNN encoder-decoder	Google in 2018[20]
VITS	cVAE+GAN	Kim et al. in 2021[9]

Table 2.1: Algorithms in Text-to-Speech (TTS) tasks

Chapter 3

MATERIAL AND METHODS

In this project, as explained in the introduction, the chatbot framework used is Nonebot. The added TTS module is integrated into the chatbot as a ”plug-in” following the developing guidance of this framework. The chatbot is deployed on a Ubuntu 20.04 server provided by Duke Kunshan University Virtual Computing Managing (DKU VCM) service.

The part of the English data set used for the Text to speech task training is from the open corpus database VCTK[4]. VCTK is an open English corpus database offered by University of Edinburgh. This database includes the speech data of 109 native English speakers with various accents. Each speaker reads about 400 sentences from newspapers. Each speaker reads a different set of newspaper sentences. Part of VCTK data set in the Ubuntu file system is presented in Figure 3.2 as an example. This data is applied in the experiment of Jaehyeon Kim et. al.[9] to compare their VITS algorithm with its predecessors in the field of Text-To-Speech algorithms. It is also used in Google's work on WaveNet. Other parts of English, Chinese, and Japanese corpus dataset were collected from open sources such as public videos and games. In data cleaning and marking, this project applied different approaches as VITS and Tacotron2 do in their experiments. Apart from using international phonetic alphabet (IPA) to mark the corresponding tones, more signs such as directional arrows (”↑,↓”), tilde (”∼”), question marks (”?”), and exclamation mark (”!”) were applied to better mark the tone and emotion in the original voice sample. Part of the cleaned corpus in displayed below in

Figure 3.1 as an example. Following the experiments in the Tacotron 2 and VITS research, the data set is transferred into 16-bit PCM with sample rate 22500Hz.

sg0_single/SG0_00_01_MAY0002.ogg_2.wav|o↑ka↓riN, da↑ijo↓obu? ʃi↑Qka↓ri ʃI↑te ʃi↑nana↓ide.
sg0_single/SG0_00_01_MAY0005.ogg_5.wav|da↑me↓dayo! mu↑rijii su↑ru no↑wa, yo↓kunai yo↓u!
sg0_single/SG0_00_01_MAY0006.ogg_6.wav|ko↑Nna bo↓roboroni na↑Qte↓ru o↑ka↓riN, mi↑terare↓naimoN.
sg0_single/SG0_00_01_MAY0007.ogg_7.wav|do↓oʃIte? do↓oʃIte mi↓raino ko↑to↓o, o↑kariNhIto↓rini o↑ʃItsuke↓runo?
sg0_single/SG0_00_01_MAY0009.ogg_9.wav|o↑ka↓riNga, no↑zoNda wa↓kejanainoni!
sg0_single/SG0_00_01_MAY0010.ogg_10.wav|so↑reni mo↑oiʧido ya↓QtaQte, ma↑ta o↑ka↓riNga ki↑zu ʦu↓kudakedaQte, o↑mo↓una.
sg0_single/SG0_00_01_MAY0011.ogg_11.wav|mi↓raino ko↑to↓o, hI↑tohIto↓ride ka↑eyoo↓naNte, ki↑Qto mu↓rina N↓dayo.
sg0_single/SG0_00_01_MAY0015.ogg_15.wav|mo↓o, ga↑Nbara↓nakUtemo i↓ikarane?
sg0_single/SG0_00_01_MAY0016.ogg_16.wav|na↑itemo i↓i N↓dayo, o↑ka↓riN.
sg0_single/SG0_00_01_MAY0017.ogg_17.wav|ma↓yu ʃi↑iwa so↓bani i↑ru↓karane o↑ka↓riN.
sg0_single/SG0_01_01_MAY0001.ogg_19.wav|a, so↑rena↓ra, a↑kIha↓barani i↑Qtemo i↓ikanaa?
sg0_single/SG0_01_01_MAY0002.ogg_20.wav|ru↑ka↓kuNto fe↓risUʧaNgane, hI↑sabisani o↑ka↓riNni a↑ita↓iQte.
sg0_single/SG0_01_01_MAY0003.ogg_21.wav|a, fe↓risUʧaN to↓ruka ku↑Nda. tu↑Qtu↓ruu.

Figure 3.1: Part of the cleaned corpus

The mean opinion score (MOS) is used to compare performance of TTS systems as introduced in the preceding section. Since subjective MOS is used in most works in TTS field as introduced above, this project will also apply subjective MOS to evaluated the final trained TTS system. The latest guideline of ITU [25] specified test form and test environment of the MOS test. The survey will take voice only form [23] since this project does not have video involved. The test environment cannot be specified as this project does not have a physical base. The main concern in the guideline is to specify the lab setup, the content presented, and the devices used, which are not determined at this stage in this project. Concerning the test form, there are several methods to choose, as listed in Table 3.1. Based on available resources of this project and with the concern of unbiasedness, in this project, I will not use the survey as an evaluation method. Instead, I will evaluate the result with objective metrics, which will be further explained in the following part.

The MOS of different algorithms introduced above is listed in Table 3.2. These scores are accessed from Paperswithcode [21].

According to their performance, tacotron 2 and VITS will be used in this project to train on the data set and obtain the final TTS systems. Their MOS will be compared to determine the final choice. In tacotron2, mel spectrograms is used as an intermediate results. It is calculated by short-Time Fourier transfrom (STFT) from the predicted

```
./wav48/p376:
total 130M
-rwxrwxrwx 1 sean sean 317K  2月   6  2022 p376_001.wav
-rwxrwxrwx 1 sean sean 497K  2月   6  2022 p376_002.wav
-rwxrwxrwx 1 sean sean 817K  2月   6  2022 p376_003.wav
-rwxrwxrwx 1 sean sean 657K  2月   6  2022 p376_004.wav
-rwxrwxrwx 1 sean sean 809K  2月   6  2022 p376_005.wav
-rwxrwxrwx 1 sean sean 665K  2月   6  2022 p376_006.wav
-rwxrwxrwx 1 sean sean 569K  2月   6  2022 p376_007.wav
-rwxrwxrwx 1 sean sean 865K  2月   6  2022 p376_008.wav
-rwxrwxrwx 1 sean sean 565K  2月   6  2022 p376_009.wav
-rwxrwxrwx 1 sean sean 1.2M  2月   6  2022 p376_010.wav
-rwxrwxrwx 1 sean sean 889K  2月   6  2022 p376_011.wav
-rwxrwxrwx 1 sean sean 517K  2月   6  2022 p376_012.wav
-rwxrwxrwx 1 sean sean 533K  2月   6  2022 p376_013.wav
-rwxrwxrwx 1 sean sean 665K  2月   6  2022 p376_014.wav
-rwxrwxrwx 1 sean sean 881K  2月   6  2022 p376_015.wav
-rwxrwxrwx 1 sean sean 673K  2月   6  2022 p376_016.wav
-rwxrwxrwx 1 sean sean 393K  2月   6  2022 p376_017.wav
-rwxrwxrwx 1 sean sean 465K  2月   6  2022 p376_018.wav
-rwxrwxrwx 1 sean sean 645K  2月   6  2022 p376_019.wav
-rwxrwxrwx 1 sean sean 425K  2月   6  2022 p376_020.wav
-rwxrwxrwx 1 sean sean 781K  2月   6  2022 p376_021.wav
-rwxrwxrwx 1 sean sean 653K  2月   6  2022 p376_022.wav
-rwxrwxrwx 1 sean sean 1.2M  2月   6  2022 p376_023.wav
-rwxrwxrwx 1 sean sean 781K  2月   6  2022 p376_024.wav
-rwxrwxrwx 1 sean sean 449K  2月   6  2022 p376_025.wav
-rwxrwxrwx 1 sean sean 209K  2月   6  2022 p376_026.wav
-rwxrwxrwx 1 sean sean 345K  2月   6  2022 p376_027.wav
-rwxrwxrwx 1 sean sean 413K  2月   6  2022 p376_028.wav
-rwxrwxrwx 1 sean sean 465K  2月   6  2022 p376_029.wav
-rwxrwxrwx 1 sean sean 577K  2月   6  2022 p376_030.wav
-rwxrwxrwx 1 sean sean 449K  2月   6  2022 p376_031.wav
-rwxrwxrwx 1 sean sean 389K  2月   6  2022 p376_032.wav
-rwxrwxrwx 1 sean sean 281K  2月   6  2022 p376_033.wav
```

Figure 3.2: Part of VCTK data set

Dimension	**Forms**
Stimulus	Single/Double/Multiple
Time presented	Once/Twice/Multiple
Presence of reference clip	Informed/Hidden/No
Interactivity of subjects	Single/Parallel vote
Presence of anchor clips	Yes/No

Table 3.1: MOS Test Forms

Name	MOS
Unit selection method	No MOS
Statistical Parametric Synthesis	No MOS
Wavenet	4.21
Fast Wavenet	4.21
Deep Voice	2.67
Deep Voice 2	3.53
Deep Voice 3	3.78
Tacotron	3.82
Tacotron 2	4.53
VITS	4.43

Table 3.2: MOS of TTS algorithms

characters. Mathematically, in discrete form,it is written as :

$$\textbf{STFT}\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]\omega[n-m]e^{-i\omega n} \tag{3.1}$$

This function is a Fourier transform with the transformed function multiplied by a window function that is non-zero for a short period of time. The window function used in Tacotron 2 is a "50 ms frame size, 12.5ms hop"[20] Hann window function. The encoder converts embedded sequence into feature representations and the decoder predict a spectrogram. The output of the final convolutional layer is passed into a single bi-directional LSTM layer to generate the final results. The final results is generated by a modified WaveNet. The WaveNet output is passed through a ReLu activatation, which simply takes the maximum of 0 and the input. IN Tacotron2, RNN and LTSM play key roles. RNNs are a class of neural networks used to handle series of data. Suppose the input vector is $[x_1, x_2, \cdots, x_n]^T$. At each step t, RNN computes y_t and hidden states $h(t)$ to process the next input, where

$$h_t = f(W_{xh}x_t + W_{hh}h_t - 1 + b_h) \tag{3.2}$$

$$y_t = g(W_{hy}h_t + b_y) \tag{3.3}$$

In these equations, x_t, y_t denote the input and output vectors at time t, h_t is the hidden states at time t. W_{xh}, W_{hh}, W_{hy} are weight matrices connecting the corresponding layers (e.g. W_{xh} is connecting the input layer and the hidden layer). b_h, b_y are biases on the hidden layer and output layer. f and g are activate functions. LSTM is a type of RNN that contains multiple gating mechanisms. These mechanisms help control the information flow through the network. At each time step, the LSTM unit takes input and previous hidden layer to produce an output and a new hidden layer. It contains an input gate, a forgot gate, and an output gate. The input gate controls how much new input is added to the memory cell c_t. The forgot gate controls how much previous value c_{t_1} should be retained. The output combines the activated result from the previous gates and current input. The forgot gate determines how much c_t will remain. Combing all outputs gives the new memory unit c_{t+1}. LSTM is helpful in modeling long-term dependencies in time series.

VITS shares similar encoder-decoder structure. Specifically, VITS extracts the phonemes from the text using an attension matrix. cVAE stands for conditional variational autoencoder. It is called conditional variational as it has an extra input to maximize the conditional log probability. GAN [6] stands for generative adversarial networks. Its structrue is presented in Figure 3.3. It uses random input to generate a sample and select a real sample. The discriminator compare the entire data set to determine which one is true.

Though the evaluation of MOS in the literature cited was all subjective. In this project, we apply an objective metric to obtain MOS, i.e., MOSNet [12]. In the voice conversion (VC) community, objective performance and evaluation criteria often fail to reflect the ”naturalness” of the result, but subjective approaches are expensive and unreliable [12]. MOSNet operates through a Deep Learning (DL) regression model to predict MOS. MOSNet accommodates a Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BLSTM) to extract representative features. As in many other CNN applications, following that CNN part and a fully connected (FC) layer is employed to predict the corresponding score. In the reported results [12], MOSNet

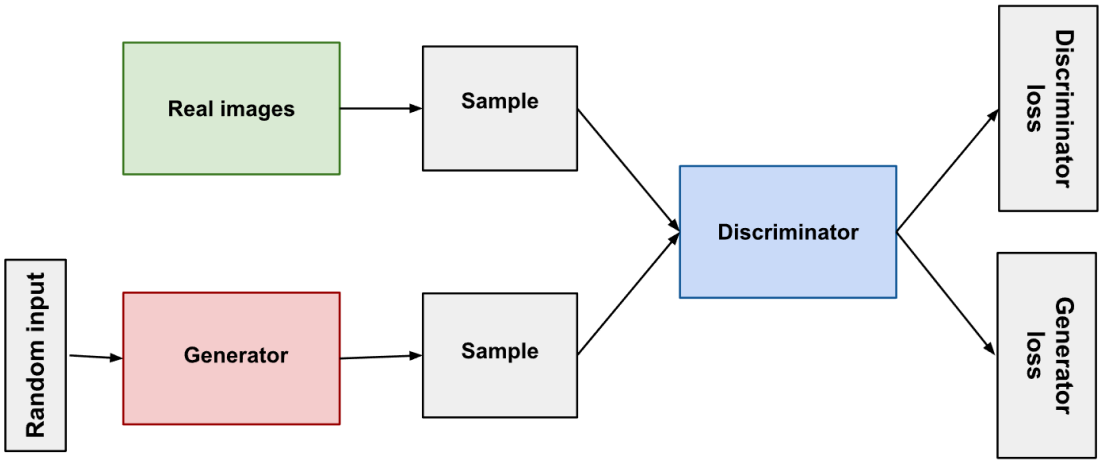


Figure 3.3: Structure of GAN [6]

achieves an accuracy of 69.6%. Since their trained model has been publicized, the evaluation of generated voice sample will be processed directly by the algorithm provided.

Chapter 4

RESULTS

RESULTS

The MOS score of the TTS algorithm is given by MOSNet tests. With MOSNet, I randomly tested 100 samples from the training data, where audio files are given by real human speech. The average score of the training is 2.62. For each algorithm trained (VITS and Tacotron 2), I generated 10 samples and tested them with MOSNet, the average scores are 2.31 and 2.39 respectively.

Following the evaluation process in Shen et al's research, I randomly selected 100 samples from the evaluation dataset. These data is not included in the training process and the ground truth are not known in the inference process. Corresponding sentences are used to generate inference samples .Different from previous research, in this project MOSNet objective evaluation is used instead of subjective evaluation. Table 4.1 shows the comparison between the original samples and the inference samples generated by both algorithms.

Name	MOSNet
Sample Score	2.77±0.06
Tacotron 2	2.56±0.035
VITS	2.63±0.08

Table 4.1: MOS Comparison for samples

As proposed by Lo et.al[12], in Voice conversion community, objective measures such as Mel-cepstral distance (MCD) are also widely applied to automatically evaluate the inference quality. Since the prediction of Mel-spectrogram is the key step in the learning process of both algorithm, I also applied this metric to objectively verify the similarities between sample data and the inference. The following Figure 4.1 and Figure 4.2 give a direct visual comparison between one sample data and its inference. For this metric, similarly, I took 100 random samples and used the corresponding corpus to generate inferences. Calculation process of MCD is completed in 2 steps. As MCD is used to compare the similarities between 2 speech samples with Mel-frequence cepstral coefficients (MFCCs) of the sample, a metric to measure features in voice, the first step of calculating MCD is calculate MFCC. This calculation process contains noise-cancellation, framing, windowing, Fourier transform, Mel filter, Compression, and Discrete Cosine Transform.

$$MFCC(X(t)) = \mathscr{A}(log(Mel(\int_{-\infty}^{\infty} w(t)x(t)e^{-i\omega t}dt))) \tag{4.1}$$

where $\mathscr{A}$ denotes the Discrete Cosine Transform and *Mel* represents the Mel-filter function. In this process, the Discrete Fourier transform takes signals in time domain and transforms then in to signals in frequency domain. The Mel filter uses the Mel-scale, which scaled signals in different frequencies, to decompose signals into separate frequency bands that mimics human perception of sound. The log and discrete cosine transform resized the processed data to get the final result of MFCC. Then, for 2 MFCC m and n, we have the MCD follows:

$$MCD(\mathbf{m}, \mathbf{n}) = \sqrt{(\frac{(\mathbf{m}\text{-}\mathbf{n})}{||\sqrt{\mathbf{1}\mathbf{m}^2}/\sqrt{len(m)}||})^2 - (\frac{(\mathbf{m}\text{-}\mathbf{n})}{||\sqrt{\mathbf{1}\mathbf{n}^2}/\sqrt{len(n)}||})^2} \tag{4.2}$$

The tested average distance is 16.10 for VITS inferences and 16.99 for Tacotron 2 inferences.

Applying the NoneBot framework, I have developed a chatbot integrated with the speech system trained. It is running on an Ubuntu 20.04 server and working on the Chinese social platform Wechat and QQ. At this stage of this project, on the Japanese corpus database, both VITS and tacotron2 were trained for 20000 epochs. Currently, one English, one Chinese, and one Japanese single speaker model have been applied to the

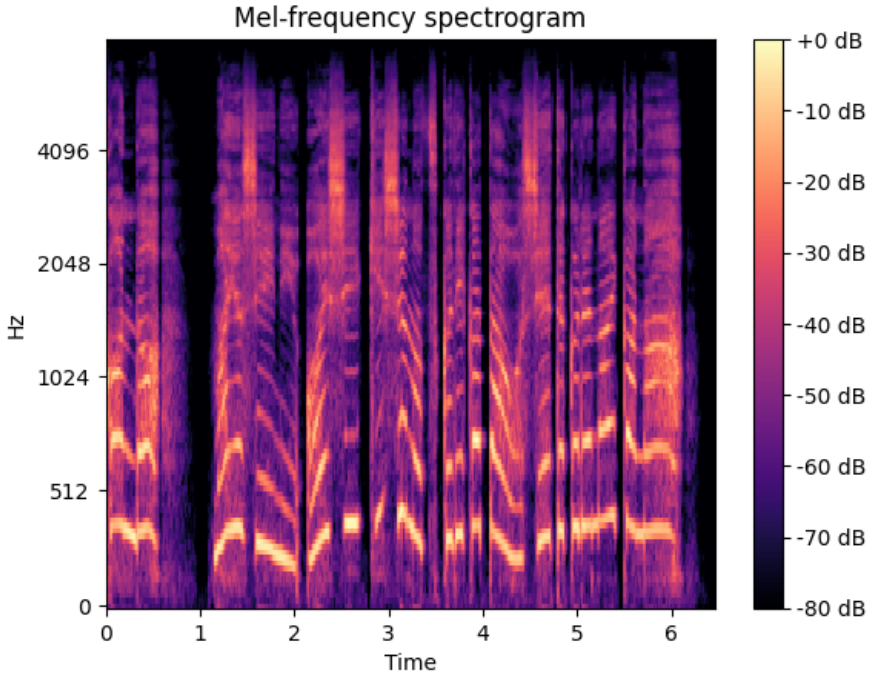


Figure 4.1: Mel-spectrogram of sample data

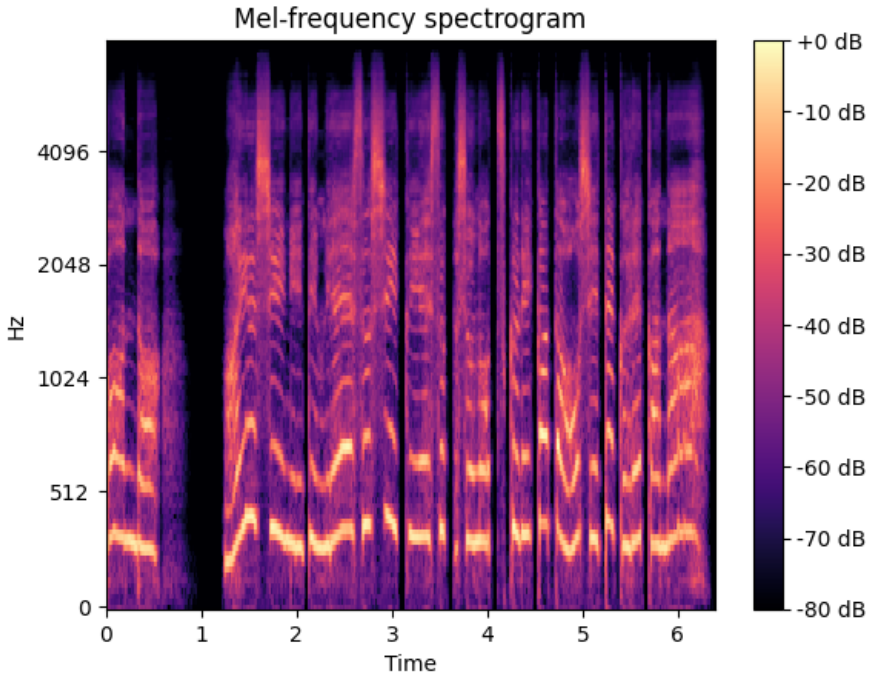
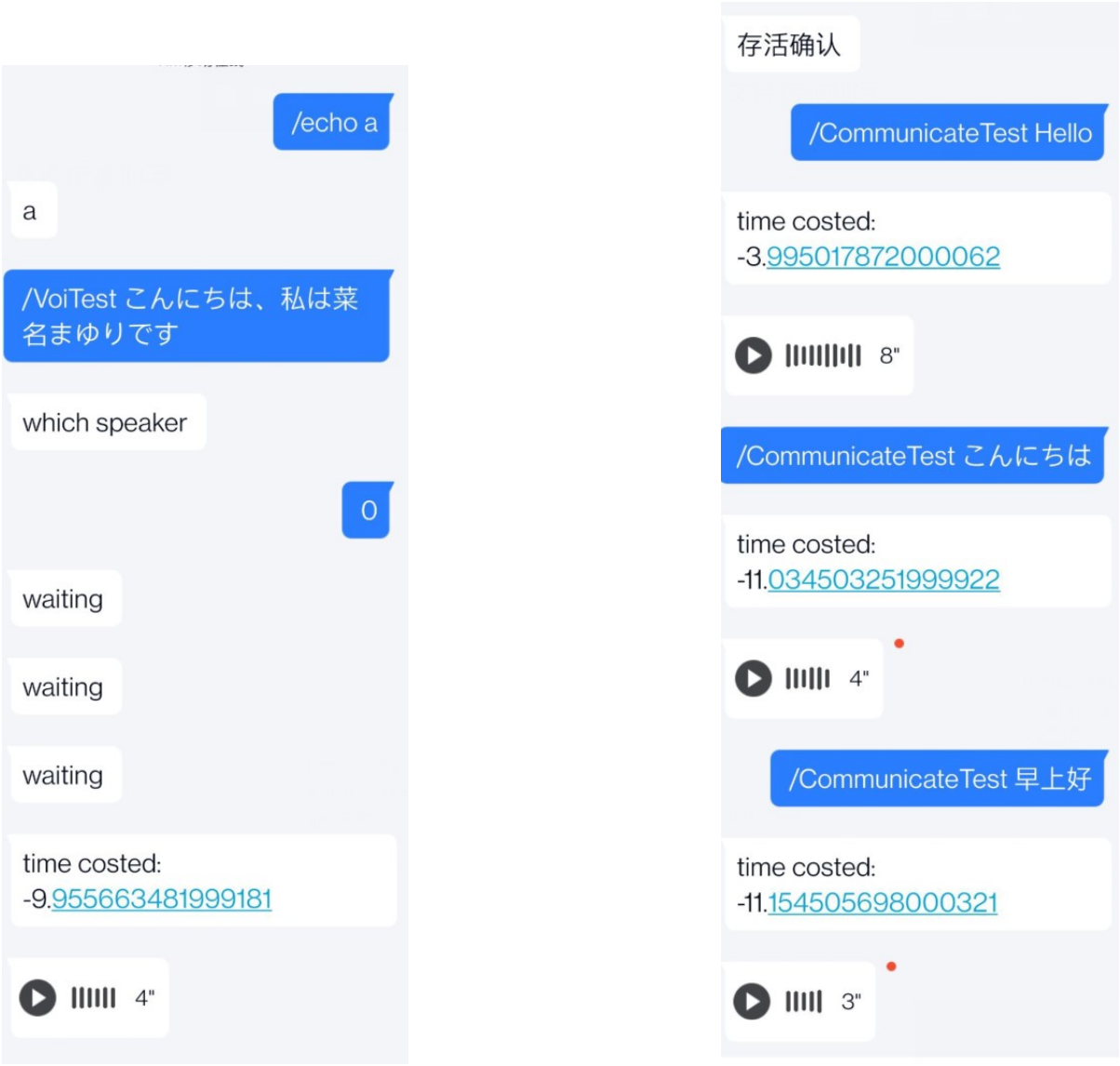


Figure 4.2: Mel-spectrogram of inference data

chatbot. They are now able to generate natural voice samples. The chatbot will generate answers from records, preset values, or NLP api responses. The following screenshot gives a short demo of the working process of the chatbot. As the testing account is borrowed from my friend, a command start is required to have a conversation with the chatbot. In normal working case, the chatbot can work properly without them. As shown, the chatbot can repeat the input, generate speech from the input, and make responses to the input. As demonstrated in Figure 4.3a, the blue text box is sent from the user and the white chat box is the message got from the chatbot response. The first message is used to test if the bot is running by using a command to let it repeat a single letter. The second message send from user is a self introduction in Japanese. At this stage, the chatbot will not recognize language and automatically select speaker. Therefore, the third message is sent to select a speaker. Then a speech response is generated by the chatbot. In the process, the chatbot will also send the waiting time. Figure 4.3b shows a simple communication, with each message sent by user as a greeting respectively, the chatbot will response a greeting from the preset values of greeting messages. As shown, the average response time for the chatbot at this stage is about 10 seconds.



(a) demo 1

(b) demo 2

Figure 4.3: demos of the chatbot

Chapter 5

DISCUSSION AND CONCLUSION

In this project, I have studied and tested the performance of different Text-to-Speech (TTS) systems, existing in the literature. Training on open data in different language and evaluating on the objective metric MOSNet, the implemented TTS system with IPA cleaners have shown high MOSNet score in relation to the sample speeches. With the Mel-cepstral Distance (MCD), I also compared the similarities between the sample data and inference data from different TTS systems. Both evaluations shows that VITS have better performance. By integrating the TTS system in a chatbot, I explored the possibility of having more natural interactions between users and the software. I believe this project is a meaningful exploration of the pure AI system applications.

On the other hand, in the process of the project, I also realized limitations of this project. First, thuogh MOSNet is an objective TTS evaluation system, its evaluation score of real-human speech is not full in the metric of MOS. This makes it hard for me to have comparisons of my TTS systems with those from previous research. Also, the accuracy of MOSNet is about 70%. As an algorithm for an regression task, this accuracy is not very reliable. Second, MCD only presents an relative comparison. It cannot offer an absolute evaluation for each system. Third, in the generating process, the time costed is long, which can reduce user friendliness of the chatbot. Based on these limitations, in future work, I think the response time could be further improved by using better hardware, storing generated data for future use, or apply quicker inference algorithms. Also, subjective MOS can be tested as a reference for better evaluation.

REFERENCES

[1] Eleni Adamopoulou and Lefteris Moussiades. “An overview of chatbot technology”. In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer. 2020, pp. 373–383.

[2] Sercan Ö Arık et al. “Deep voice: Real-time neural text-to-speech”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 195–204.

[3] Junyoung Chung et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling”. In: *arXiv preprint arXiv:1412.3555* (2014).

[4] University of Edinburgh. *VCTK*. https://datashare.ed.ac.uk/handle/10283/2950. Accessed: 2022-12-16.

[5] Andrew Gibiansky et al. “Deep voice 2: Multi-speaker neural text-to-speech”. In: *Advances in neural information processing systems* 30 (2017).

[6] Ian Goodfellow et al. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[7] Harry Henderson. *Artificial intelligence: mirrors for the mind*. Infobase Publishing, 2007.

[8] Andrew J Hunt and Alan W Black. “Unit selection in a concatenative speech synthesis system using a large speech database”. In: *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*. Vol. 1. IEEE. 1996, pp. 373–376.

[9] Jaehyeon Kim, Jungil Kong, and Juhee Son. “Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech”. In: *International Conference on Machine Learning*. PMLR. 2021, pp. 5530–5540.

[10] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17022–17033.

[11] Zachary C Lipton, John Berkowitz, and Charles Elkan. “A critical review of recurrent neural networks for sequence learning”. In: *arXiv preprint arXiv:1506.00019* (2015).

[12] Chen-Chou Lo et al. “Mosnet: Deep learning based objective assessment for voice conversion”. In: *arXiv preprint arXiv:1904.08352* (2019).

[13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[14] Keiron O’Shea and Ryan Nash. “An introduction to convolutional neural networks”. In: *arXiv preprint arXiv:1511.08458* (2015).

[15] Aaron van den Oord et al. “Wavenet: A generative model for raw audio”. In: *arXiv preprint arXiv:1609.03499* (2016).

[16] Tom Le Paine et al. “Fast wavenet generation algorithm”. In: *arXiv preprint arXiv:1611.09482* (2016).

[17] Wei Ping et al. “Deep voice 3: Scaling text-to-speech with convolutional sequence learning”. In: *arXiv preprint arXiv:1710.07654* (2017).

[18] Tim Salimans et al. “Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications”. In: *arXiv preprint arXiv:1701.05517* (2017).

[19] Rakesh Kumar Sharma and Manoj Joshi. “An analytical study and review of open source chatbot framework, RASA”. In: *International Journal of Engineering Research and* 9.06 (2020).

[20] Jonathan Shen et al. “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions”. In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2018, pp. 4779–4783.

[21] *Speech synthesis*. https://paperswithcode.com/task/speech-synthesis. Accessed: 2022-12-16.

[22] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. “Highway networks”. In: *arXiv preprint arXiv:1505.00387* (2015).

[23] Robert C Streijl, Stefan Winkler, and David S Hands. “Mean opinion score (MOS) revisited: methods and applications, limitations and alternatives”. In: *Multimedia Systems* 22.2 (2016), pp. 213–227.

[24] Alan Mathison Turing. “Mind”. In: *Mind* 59.236 (1950), pp. 433–460.

[25] International Telecommunication Union. *ITU-T P.800.1*. https://www.itu.int/rec/dologin_pub.asp?lang=s&id=T-REC-P.800.1-201607-I!!PDF-E&type=items. Accessed: 2022-12-16.

[26] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. “Pixel recurrent neural networks”. In: *International conference on machine learning*. PMLR. 2016, pp. 1747–1756.

[27] Mahesh Viswanathan and Madhubalan Viswanathan. “Measuring speech quality for text-to-speech systems: development and assessment of a modified mean opinion score (MOS) scale”. In: *Computer speech & language* 19.1 (2005), pp. 55–83.

[28] Yuxuan Wang et al. “Tacotron: Towards end-to-end speech synthesis”. In: *arXiv preprint arXiv:1703.10135* (2017).

[29] Joseph Weizenbaum. “ELIZA—a computer program for the study of natural language communication between man and machine”. In: *Communications of the ACM* 9.1 (1966), pp. 36–45.

[30] Yongyu Yang. *Nonebot/nonebot2*. https://github.com/nonebot/nonebot2. Accessed: 2022-12-16.

[31] Heiga Zen, Keiichi Tokuda, and Alan W Black. “Statistical parametric speech synthesis”. In: *speech communication* 51.11 (2009), pp. 1039–1064.