Voice conversion generative model for resolving Dysarthria

Changhyeon, Lee (20225278)
UNIST AIGS
changhyeon@unist.ac.kr

Kwangryeol, Park (20225466)
UNIST AIGS
pkr7098@unist.ac.kr

Abstract

We aim to convert the voice of a speaker with dysarthria into the voice of a non-impaired speaker using generative models. Dysarthria is a disorder which affects an individual's speech intelligibility. This makes it difficult for the speaker to communicate with others. We will use a deep generative model to transform the voice of the speaker for smooth communication. We introduce Speech-to-Text and Text-to-Speech pipeline for pronounce conversion. In the pipeline, the spoken transcript is obtained from the Speech-to-Text model. Text-to-Speech model which is trained on speaker's voice data generates non-impaired voice data using the output of Speech-to-Text model. For the data augmentation for Text-to-Speech model, we introduce voice conversion model which learns the feature of speaker's pronunciation. Finally, personalizing our pipeline and using it for people with dysarthria can reduce the communication problems.

1 Introduction

There are lots of restrictions on daily conversation and social activities because dysarthria makes communication difficult. In order to solve this problem, there are previous studies which convert the voice data into other domains. However, not much research on dysarthria speech to normal speech and its Korean speech data version. The reason is that Korean words consist of the combination of vowels and consonates, so that the pronunciation rules are more complicated compared to other languages. Moreover, there are difficulties in analyzing and processing the speech data due to the intonation. Therefore, we identify the characteristics of Korean speech data, use preceding models and fine-tune them by considering the Korean dysarthria data (2). Through this project, people with speech impairments and our society will be able to achieve better harmony than now.

2 Related work

E2E-DASR. An end-to-end dysarthric automatic speech recognition (DASR) system (10). It uses a spatio-temporal DASR system utilizing Spatial Convolutional Neural Network (SCNN) and Multi-Head Attention Transformer (MHAT) (15) to visually extract the speech features.

3 Methodology

3.1 Data

In this project, two types of data were used. In order to make the models learn the accurate distribution of Korean speech data and Korean text data, we used the target dysarthria speech data and standardized Korean speech data. The data has been publicly released.

3.1.1 Dysarthria speech

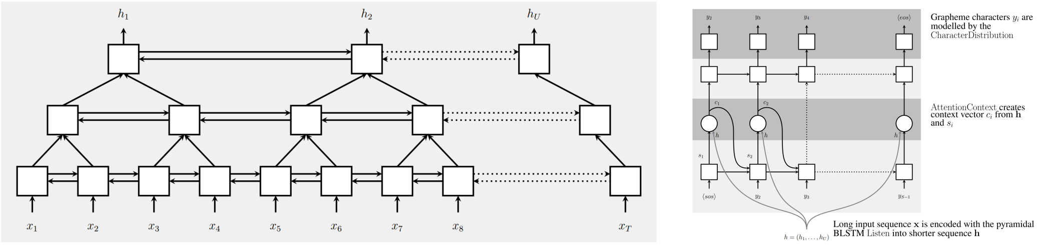
We use the Korean dysarthria dataset provided by AIHub (1). The dataset consists of wav format audio file and its script data. It is obtained from various ages, regions, genders and diseases. We manually pre-processed the given data because it needs cutting mute and de-noising, and obtain about 100 minutes dataset. Since patients with dysarthria caused by various diseases uttered several predetermined scripts, given data contains the characteristics of dysarthria resulting from different diseases. Also, the Speech-to-Text or Text-to-Mel task requires meta-data processed from RAW voice data. The data we use includes the speaker's raw voice data, utterance transcription data, and token data which is matched one-to-one with the transcription text. It is also necessary to convert raw data into a spectrogram format with features in the frequency domain rather than using it as it is. The data converted to spectrogram is processed by the artificial intelligence model in the form of 2D data such as images so that we can use some methods used by image-domain i.e., CNN. Another, Attention is generally used in NLP task models such as the seq2seq model. Attention focuses on where the given data is represented in another given data.

3.1.2 Korean Single Speaker Speech.

Korean Single Speaker Speech (KSS) dataset (11) is designed for the Korean text-to-speech task. It consists of audio files recorded by a professional female voice actoress and their aligned text extracted from books. The data consists of a wav file with a sampling rate of 22,050 and a script file used to utter each voice data. We used the data as pre-train data, and each model was trained over 400k steps.

3.2 Model

3.2.1 Speech to Text


h_1
h_2
h_U
x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_T
y_2
y_3
y_4
$\langle eos \rangle$
Grapheme characters y_i are modelled by the CharacterDistribution
c_1
c_2
AttentionContext creates context vector c_i from h and s_i
h
h
h
s_1
s_2
$\langle sos \rangle$
y_2
y_3
y_{S-1}
$h = (h_1, \ldots, h_U)$
Long input sequence x is encoded with the pyramidal BLSTM Listen into shorter sequence h

(a)

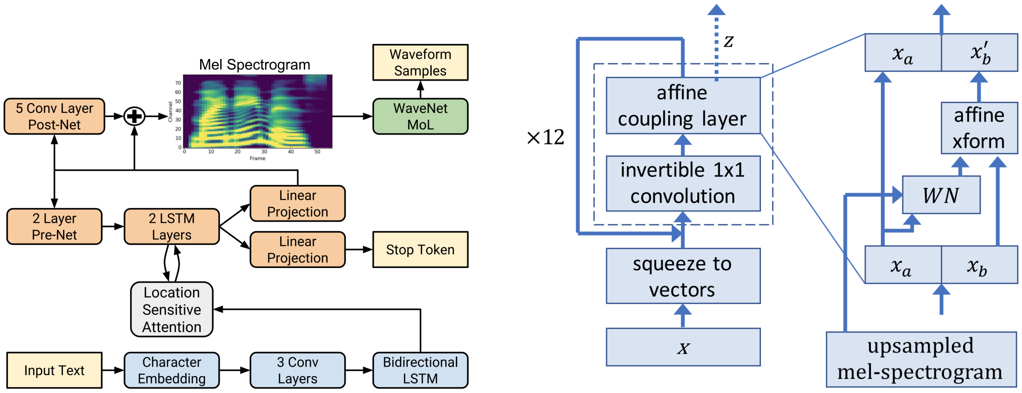
(b)

Figure 1: Listen, Attend and Spell (LAS) model (4) : (a) Listener is a pyramidal BLSTM encoding input sequence x into high level features h. (b) Speller is an attention-based decoder generating the y characters from h. Each figure is taken from the LAS paper.

We use LAS (4) which is an end-to-end model for converting speech data to transcriptions. Unlike the previous speech-to-text model, it is not depend on Hidden Markov Model and there is no independence assumption on label sequence. LAS consists of two parts. Figure 1(a) is pyramid bidirectional LSTM (BLSTM) based encoder called Listener and Figure 1(b) is attention based decoder, Speller. The encoder takes dysarthria voice data and makes high level feature data thought three stacked BLSTM layers. The decoder generates character distribution and predicts character from MLP layer with softmax, which takes the decoder context and state. The decoder context is from Attention Context layer (3; 5) which takes the output of encoder and decoder state. The decoder state is computed from stacked LSTM layers whose inputs are the previous decoder state, the output of encoder and decoder context.

3.2.2 Text to Mel-spectrogram

Tacotron2 (13) is used with WaveGlow and outputs mel-spectrogram for bridge data between the two models. Tacotron2 consists of attention based encoder and decoder. Input characters for encoder are represented by using a learned 512-dimensional embedding. The outputs of encoder are hidden features which is passed through 3 Conv layers, BLSTM and location-sensitive attention (5). The decoder which is based on autoregressive recurrent neural network generates mel-spectrogram from the output of encoder. The prediction from the previous time step is first passed through a Pre-Net containing two fully connected layers. The output of Pre-Net and the attention output of encoder is concatenated and pass into LSTM layer. The concatenation of the LSTM output and the attention context vector is projected through a linear transform to predict the target mel-spectrogram frame. Last but not least, Post-Net takes the mel-spectrogram and predicts a residual to add to the prediction to improve the overall reconstruction. See Figure 2(a).



(a) (b)

Figure 2: (a) Block diagram of the Tacotron2 () architecture. (b) WaveGlow () network from paper.

3.2.3 Mel-spectrogram to Speech

WaveGlow (12) which converts the mel-spectrogram to speech data is used with Tacotron2. The model is based on flow model and combines the two models, Glow (9) and WaveNet (14). The model consists of invertible 1x1 CNN and affine coupling layer (6). Following Glow model, the weights of invertible 1x1 CNN are initialized to be orthonormal so that it can be invertible. The log scale determinant Jacobian of this transformation joins the loss function. The affine coupling layer takes the output of CNN. In the layer, the half of the input channels, x_a in Figure 2(b) are pass to the output of the layer and WN transformation at the same time. The output of WN and another input half channels, x_b pass to affine transformation layer whose outputs, x'_b are concatenated with x_a. WN can be any transformation which has invertible parameters due to the unchanged form of x_a, and also x'_b can be inverted to x_b.

3.2.4 Voice conversion

MaskCycleGAN-VC (8) is a non-parallel voice conversion model which does not need parallel corpus. The model is trained on Filling in Frames task in order to alleviate the problem of CycleGAN-VC2 (7) which does not have sufficient ability. In overall pipeline of FIF shows Figure 3, the model use mask m to input x. The mask, m where the black part is zero and others are one is randomly determined based on a predetermined rule (8). Next, the MaskCycleGAN-VC converts $\hat{x}$ and m into y' using $G^{mask}_{X\to Y}(\text{concat}(\hat{x}, m))$. By using conditional information m, the $G^{mask}_{X\to Y}$ can fill the information of masked region. After generatesy', the model uses $G^{mask}_{Y\to X}(\text{concat}(y', m'))$ to reconstruct the x'' with all-one matrix m' and y'. Finally, the model applies the cycle-consistency loss for the original, x and reconstructed mel-spectrograms, x':

$$\mathcal{L} = \mathbb{E}_{x\sim P_X, m\sim P_M}[||x'' - x||_1] \tag{1}$$

4 Pipeline

4.1 Pre-train process

We pre-trained four models. First, the Speech-to-Text model was trained on dysarthria data so that the model can properly learn the distribution between the characteristics of dysarthria voice data and

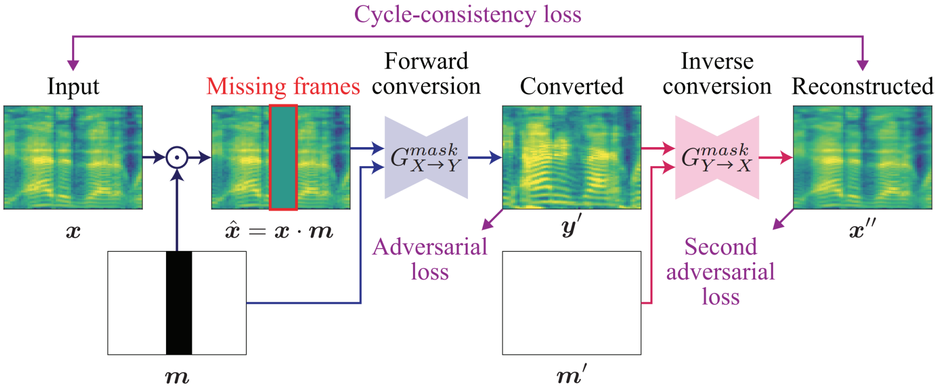


Figure 3: Pipeline of Filling in Frames (FIF). We encourage the converter to fill in the missing frames (surrounded by the red box) based on the surrounding frames through a cyclic conversion process.

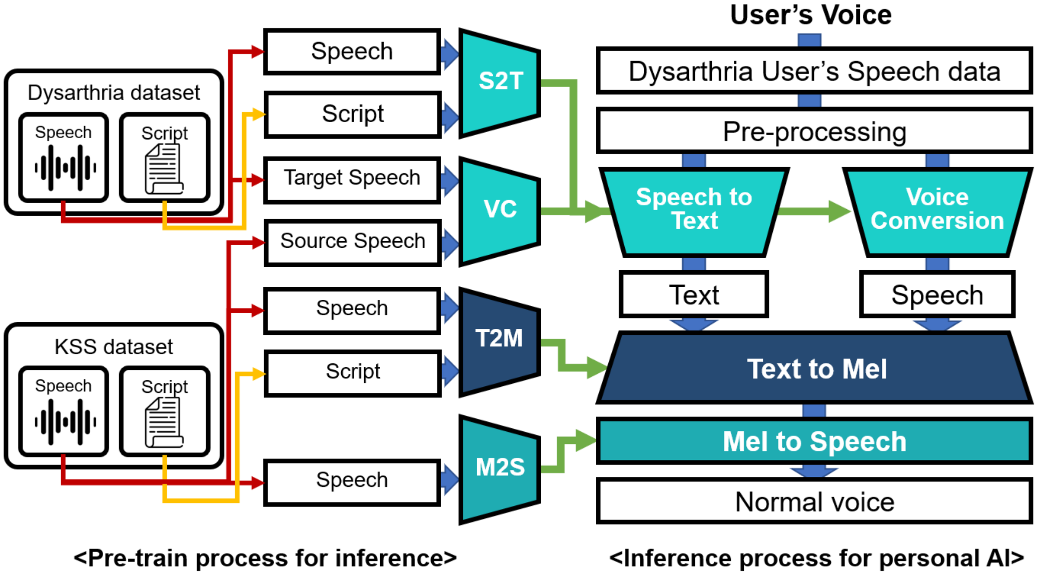


Figure 4: The figure shows the pipeline configuration of our project. We pre-trained four models using dysarthria data and KSS data. Using a pre-trained model, it can be re-trained and used through the user's voice data and script.

text data. The used data are obtained from cerebral palsy patients among dysarthria data. It consists of data in which about 200 to 300 lines were uttered by on patient. Each data was pre-processed and classified by sentence to form a total of 3,000 voice data. In the case of dysarthria caused by cerebral palsy, the model was able to learn well enough because there are the same level of features that the model can learn. Second, the voice conversion model learned the speech characteristics of dysarthria patients in order to increase the amount of data and the number of sentences. In the case of source speech, KSS data and the data uttered by project facilitators were used. For target speech, some patient data selected from dysarthria data was used. About 7,000 epochs were trained for proper speech conversion performance, and it took more than 20 hours to pre-train each model. Third, in the case of Text to Mel-spectrogram, pre-training was performed using KSS data. During learning, the audio signal waveform data is converted into mel-spectrogram and used. In addition, the sentences paried with each mel-spectrogram are divided into phoneme units to learn the mel-spectrogram of each phoneme. In the case of text to mel-spectrogram, there was no specific evaluation index to check whether pre-learning was properly performed. To check the performance, we confirmed the degree of learning by checking the actually generated voice signal based on each 10,000 steps. As a result, we selected a model that trained 480,000 steps. Fourth, the mel-spectrogram to speech model was pre-trained using KSS data. The model is trained using speech data as input. The model needs the huge amount of dataset for learning the distribution between speech data and mel-spectrogram. KSS was made with accurate pronunciation and was more effective in learning because of the huge amount of data. In addition, a voice signal was generated and confirmed every 10,000 steps. Finally, the model that trained about 450,00 steps was selected.

4.2 Retrain process per user

After configuring the pipeline using the above-mentioned pre-train models, it must go through a re-learning process to reflect the voice information of each user. In the case of the Speech-to-Text model, it has already been trained with voice data for dysarthria, so it can sufficiently convert the voice uttered by a user with dysarthria into text. However, since the Text-to-Mel model is trained with KSS data, it has voice information of KSS data. In order to dilute it and overlay the user's voice information, the user must utter and relearn according to the set script. The Mel-to-Speech model also needs to be retrained with user speech data. The voice conversion model, which we previously used to generate data of various sentences and large amounts of data, may or may not be used in the re-training process. The reason is that if a dysarthmic speaker has difficulty uttering a large amount of data, it can be used to increase the amount of data by learning a voice conversion model using some data. Briefly, re-training is carried out in order to learn the user's voice information in the Text-to-Mel model and the Mel-to-Speech model.

4.3 Inference process for personal AI

After the pipeline composed of the pre-trained model is retrained with the user's voice, the pipeline can actually operate. When the user's voice is received as input, the Speech-to-Text model generates text from the voice based on the voice information of speakers with dysarthria. The text is generated as a mel-spectrogram through a Text-to-Mel model. And, since the Mel-to-Speech model has the speech information of the user's voice and the correct pronunciation learned in advance, it can generate a voice with the user's voice and accurate pronunciation from the generated mel-spectrogram.

5 Experiments

5.1 Inference test

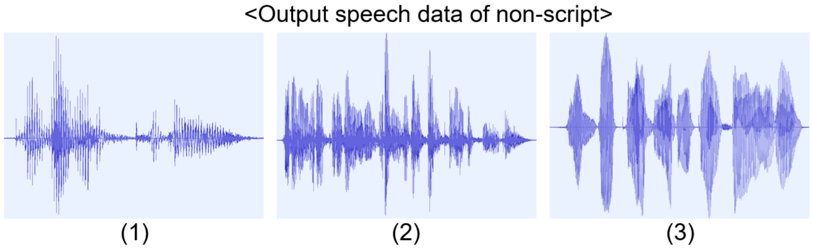
We tested our pipeline by retraining a pipeline composed of pre-trained models with specific dysarthria patient data. Using data from a 62-year-old man and voice data from a 44-year-old woman, they created personalized AI models for each. We checked the results of four cases: audio signals for learned sentences, audio signals for unlearned sentences, simple sentences, and long sentences. We tried to classify the results based on pronunciation and differences between real and generated voices. **The data that generated various sentences can be checked in the attached file like the corresponding report.**

5.1.1 Experiment using data from a 62-year-old male

The degree of dysarthria in the patient is not severe, and speech is often slurred. The number of sentences uttered by the patient is 200, and the amount of data was increased by using a voice conversion model from that data. As a result, it was retrained with about 1,000 data. In addition, 100,000 steps were re-trained for each of the Text-to-Mel model, which had pre-trained about 480,000 steps, and the Mel-to-Speech model, which had pre-trained about 450,000 steps.

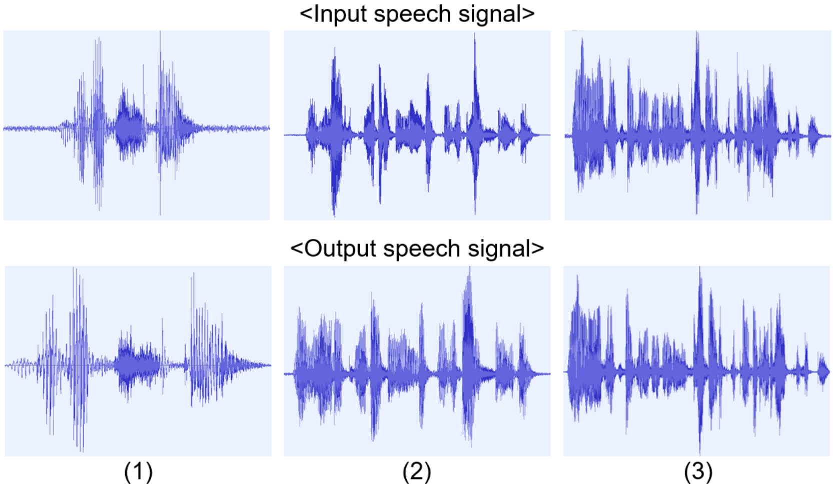
Learned sentences. In the case of learned sentences, some noise is generated, and the pronunciation is slightly more accurate or the same. Both long and short sentences produced adequate speech cues.

Unlearned sentences. Sentences that were not learned were included in the KSS dataset, but in the case of sentences that were not retrained, voice signals were generated with the speaker's voice appropriately. In this case, both long and short sentences could be generated appropriately. However,



(1) “배가 아프다.” (2) “꼬리를 늘어뜨리고 있는 호랑이에게 땅콩을 줬습니다.”
(3) “행복한 하루 되세요.”

Figure 5: In this Figure, sentences that did not exist in the script were generated and represented as audio signal waveforms. In the case of (1) and (2), the user’s voice was appropriately generated for short and long sentences. However, in the case of (3), it was properly generated with the voice of the KSS dataset. The Korean text below is the transcript of the signal.



(1) “나무 아래 풀밭에는 메뚜기가 있습니다.” (2) “물이 차다.”
(3) “음운론을 가르치시던 담임 선생님은 흙에서 넓죽한 고구마를 캐셨다.”

Figure 6: In this figure, the upper part (Input speech signal) is the speech signal directly uttered by the dysarthmic speaker for each script. The lower part (Output speech signal) is the waveform of the speech signal generated through our pipeline. It was created properly, and there were cases where the change in pronunciation was prominent and cases where it was not.

for sentences that were never learned, voices were generated with the voices of the KSS dataset. Judging from the results, it seems that all phoneme features were not extracted from the user’s voice because a large amount of characters were not learned in the relearning process. A large amount of KSS dataset was created because it went through 480,000 steps.

5.1.2 Experiment using data from a 70-year-old female

The original pronunciation of patient is more severe. The number of sentences is about 300, and also the amount of data was increased by using the voice conversion model and produced about 1,00 data. The training was done as above experiment.

Learned sentences. Compared to the above experiment, the noise is reduced, and the pronunciation is slightly better or the same. Both long and short sentences produced adequate speech cues.

Unlearned sentences. In the case of female user’s voice, the quality of overall converted voice is better than the above male experiment. This is because the KSS dataset which was used for

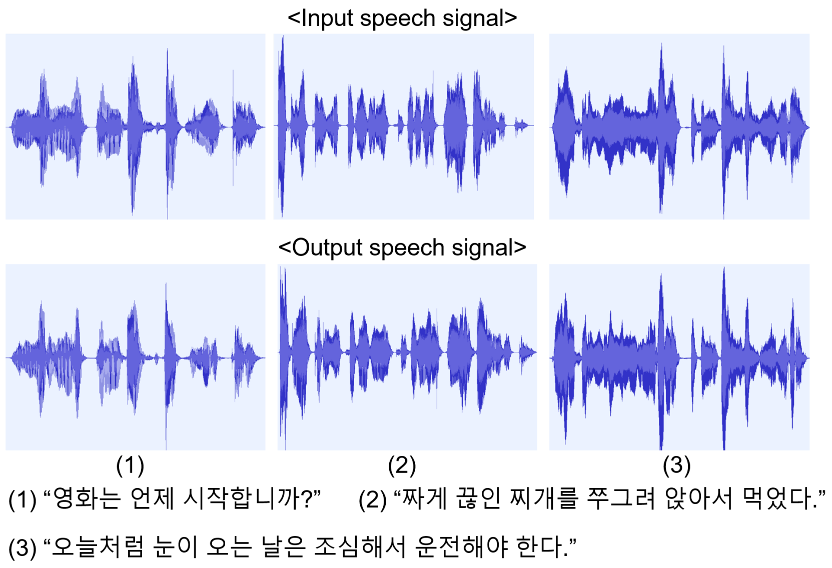


Figure 7: In this figure, the upper part (Input speech signal) is the speech signal directly uttered by the dysarthmic speaker for each script. The lower part (Output speech signal) is the waveform of the speech signal generated through our pipeline. It was created properly, and there were cases where the change in pronunciation was prominent and cases where it was not.

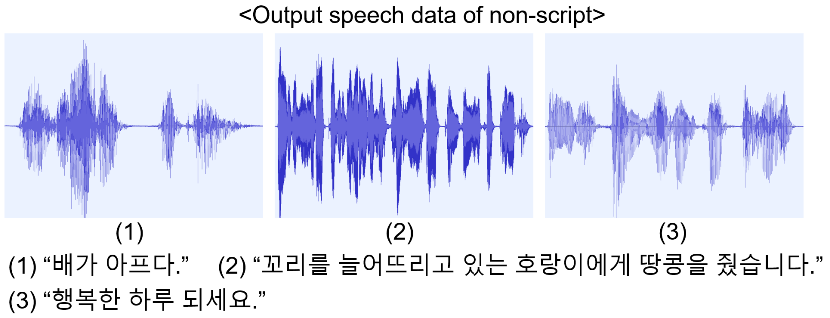


Figure 8: In this Figure, sentences that did not exist in the script were generated and represented as audio signal waveforms. In all of the three cases 70-year-old female, the user’s voice was appropriately generated compared to the above experiment. This is because the model pre-trained on KKS dataset which is spoken by female.

pre-training is female based. The model already is trained of the distribution of female voice so that it can show better result on female dataset.

6 Future work

Our future work describes the problems we felt when constructing the pipeline and dealing with various generative models in speech and signal task, and what we think is necessary for more efficient learning and use. We also considered ways to overcome the limitations of our pipeline.

6.1 Create Korean dataset

A large number of studies are being conducted to learn and analyze the relationship between speech signals and sentences. However, research on Korean speech signals does not seem to be actively conducted. We think the reason is that there are not many Korean speakers, and the sentence structure of Korean is very different from English. In addition, a large amount of data is required to identify different sentence structures, but there is not much Korean speech data. In our case, we used KSS data, but since this data consists of female voices, we could not consider differences by gender or age. Therefore, if you create a data set composed of various voices and many sentences, it will be very helpful for future research.

6.2 Feature extraction of speech signal

In the case of spoken voice signals, it is often difficult to properly extract voice information from voice signals because they consist of various frequencies and contain different information depending on who uttered the voice. Moreover, in the case of dysarthria, it was difficult to extract information because the symptoms resulting from dysarthria could alter not only the individual's pronunciation but also the voice. Of course, if there is a large amount of speech data, information for each phoneme can be extracted, but this is inefficient. If we can identify the peculiarities of Korean voices, it is expected that we can restore or change individual voices to other voices.

6.3 Speech-to-Speech Model for Pronunciation Correction

Most of the speech-to-speech models are used to translate speech and convert it into another language. Models for translating and translating into other languages have been studied a lot and are known to have good performance. However, it seems that there are not well studied models for the purpose of pronunciation correction or voice correction. The machine translation models studied so far perform symmetric translations for different languages. Here, it is thought that a model that can symmetrically convert voice signals with appropriate voices and pronunciations, rather than other languages, can also have good performance if research is conducted. As for expected problems, it is easy to train because each language has its own pronunciation and characteristics, but in the case of voices and pronunciations, there may be difficulties because they can have various characteristics due to various factors.

6.4 Efficient training of generative models

Because the generative model used in our project has a complex structure and many parameters, it required a lot of time and data. In addition, repeated experiments were conducted for optimization by adjusting the structural hyper-parameters and loss function of the model. This process is a chronic problem with certain generative models. In order to solve this problem, it would be good to combine Experience Replay and Path Length Regularization, or representative model weight reduction methods such as quantization, model compression, and pruning.

7 Conclusion and Discussion

Establish a pipeline that changes pronunciation to normal while maintaining the voice of patients with dysarthria. Because the built pipeline consisted of four generative models, the results were poor if each model did not perform properly. However, the pipeline produced adequate speech for the learned sentences. It is difficult to generate speech signals for unlearned sentences. It would be impossible to train all sentences, but I think it will work properly if we can get a little more sentences and refined data. Although it may not be practical for reasons such as data, time, and accuracy of the model, the limitations and problems of generative models have been identified from a project that accomplishes one big goal by linking various generative models together. If we learn the model with more data and time, we will be able to build personalized artificial intelligence for users with pronunciation correction or dysarthria.

`https://github.com/KwangryeolPark/DGM.Project.Group25`

References

[1] Ai hub.

[2] Ai hub dysarthria speech data.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[4] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. Listen, attend and spell, 2015.

[5] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition, 2015.

[6] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp, 2017.

[7] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. Cyclegan-vc3: Examining and improving cyclegan-vcs for mel-spectrogram conversion, 2020.

[8] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. Maskcyclegan-vc: Learning non-parallel voice conversion with filling in frames, 2021.

[9] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions, 2018.

[10] Hendrik Klopries and Andreas Schwung. Extracting interpretable features for time series analysis: A bag-of-functions approach. *Expert Systems with Applications*, 221:119787, 2023.

[11] Kyubyong Park. Kss dataset: Korean single speaker speech dataset, 2018.

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

[13] 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, 2018.

[14] 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.

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