Audio Speech-Separation

Final Report

Khaled Khalil	(1003183654)
Akino Watanabe	(1004133653)
Suri Xiang	(1003211044)

1987 words; penalty 0%

Introduction

Audio recognition technology has increasingly been applied to solve a wide variety of real-world problems. Meanwhile, speech separation - splitting an audio into its original sources - is a critical challenge in audio recognition[1], especially when the audio records many people's voices and background noises all at the same time (the Cocktail Party Effect)[2]. Hence, this project aims to create a machine learning (ML) model that can split an audio recording of two people talking over each other to its original audio sources.

ML is a reasonable approach for this project because a neural network has the capability to detect nonlinear, complex patterns of audio signals. By specifically using long short-term memory (LSTM) layers, the model will learn the temporal structure of the audio, which is essential in speech separation[1].

Therefore, the project's outcome, an automatic speech-separation model, could be used in many technologies or services. For example, the model's output can be converted to text files so that hearing-impaired people could understand the content of audio speeches by reading text transcripts. The separated audio files could also be used for simultaneous translation in videos or conferences, classification, transcription, and speech enhancement.

Background & Related Work

There were two major related works that inspired us when approaching the project.

Firstly, Google published research regarding audio-visual speech-separation[2] in 2018. The model had a multi-stream neural network-based architecture; the visual stream captured facial features, and the audio stream inputted a spectrogram to the model and learned audio features by a dilated CNN[2]. The streams were then combined as an audio-visual representation, while the model outputted a time-frequency mask for each speaker so that the specific person's audio was enhanced while all other sounds were suppressed.

Secondly, in 2018, Columbia University published research regarding the development of TasNet, a Time-domain Audio Separation Network[3]. The paper proved that converting an audio data to a spectrogram through short-time Fourier transform (STFT) would introduce an upper bound on separation performance. Therefore, TasNet directly modelled the audio signals, created audio masks using deep LSTM network to incorporate time-dependencies of the audio segments, and finally decoded the masks to reconstruct the audio sources using transpose convolutions.

Both the audio-visual model and TasNet outperform previous works on speech-separation tasks. Nevertheless, the computations of such deep/complex learning model are expensive. In this project, we referenced the concepts of masking, waveform, and time dependencies of the audio segments when developing models.

Data Processing

For this project, the TIMIT corpus[4] was used to create the datasets, which included 6300 sentences spoken by 630 speakers from 8 different dialect regions across the United States. The sentences were phonetically diverse and emphasized the dialectal variations [Figure 1]. The speeches were initially separated into the train and test sets within the corpus. The test set included 168 speakers with about 27% of all the sentences in the TIMIT corpus [Figure 2].

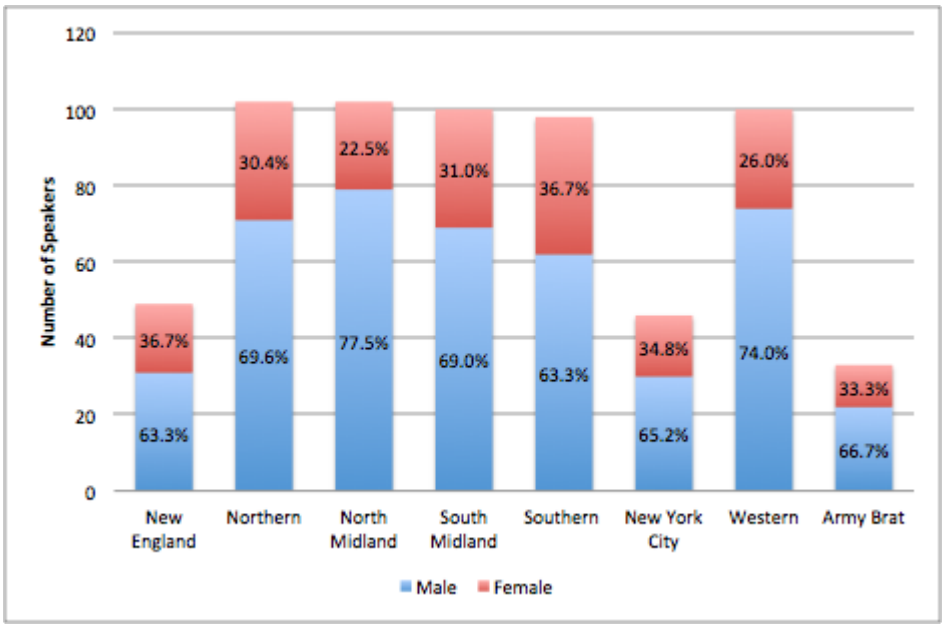

Figure 1. *Dialect and Gender Distribution of All the Speakers in the Corpus*

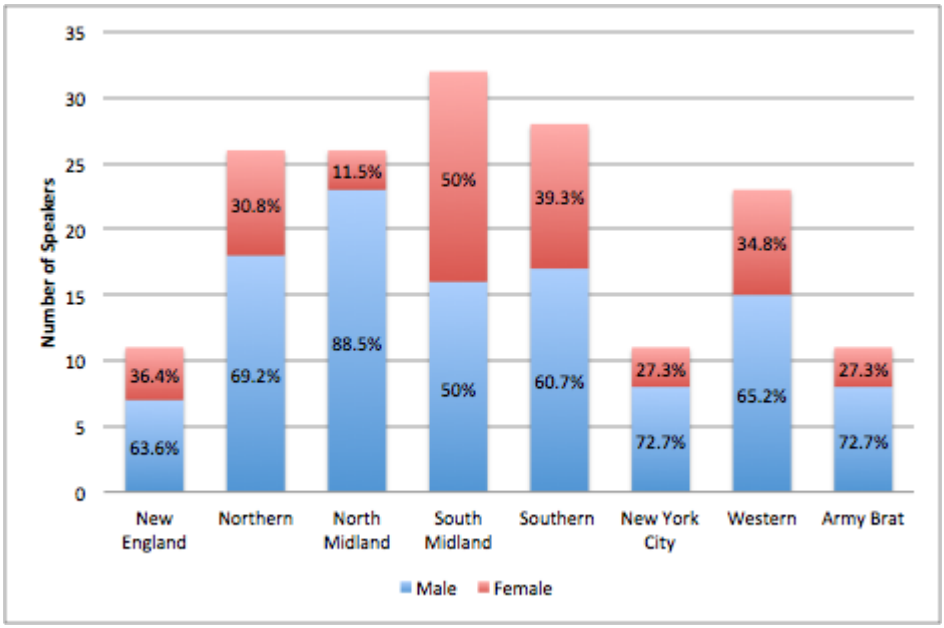

Figure 2. *Dialect and Gender Distribution of the Speakers in the Test Set*

Each sentence data had four files, including an audio file (.WAV, a SPHERE file), an orthographic transcription of the uttered words (.txt), a time-aligned word transcription (.wrd), and a time-aligned phonetic transcription (.phn). For this project, we mainly used the audio file, which contained a *waveform* represented by a tensor; each number in the tensor was the voltage difference of the audio signal at the specific time step.

Using the TIMIT corpus, we generated training and test datasets by randomly selecting two audio files from the same corpus set, extracting the contents of the .WAV files in waveforms, overlying the longer waveform on the shorter one, and exporting them as .wav files using the Pydub library.

Therefore, the overlay waveform had the same length as the shorter one. We then used the training set of the TIMIT corpus to create the training (7500 files) and validation (2500 files) sets for this project, and used the test corpus to create 2500 files as the test set. The model took the waveform tensors as inputs and original spoken sentences as the ground truths used during calculating the losses.

Architecture

One of the key challenges we faced was batching with variable waveform sizes. To solve this problem, we preprocessed the input data and divided them into smaller equally-sized segments. After trying many segment sizes, we decided to use one second because it was long enough to contain coherent and intelligible English content, while short enough to reduce the complexity of the training process. Based on the TA's suggestion, we deleted the last segment from the waveform if it did not make a full second.

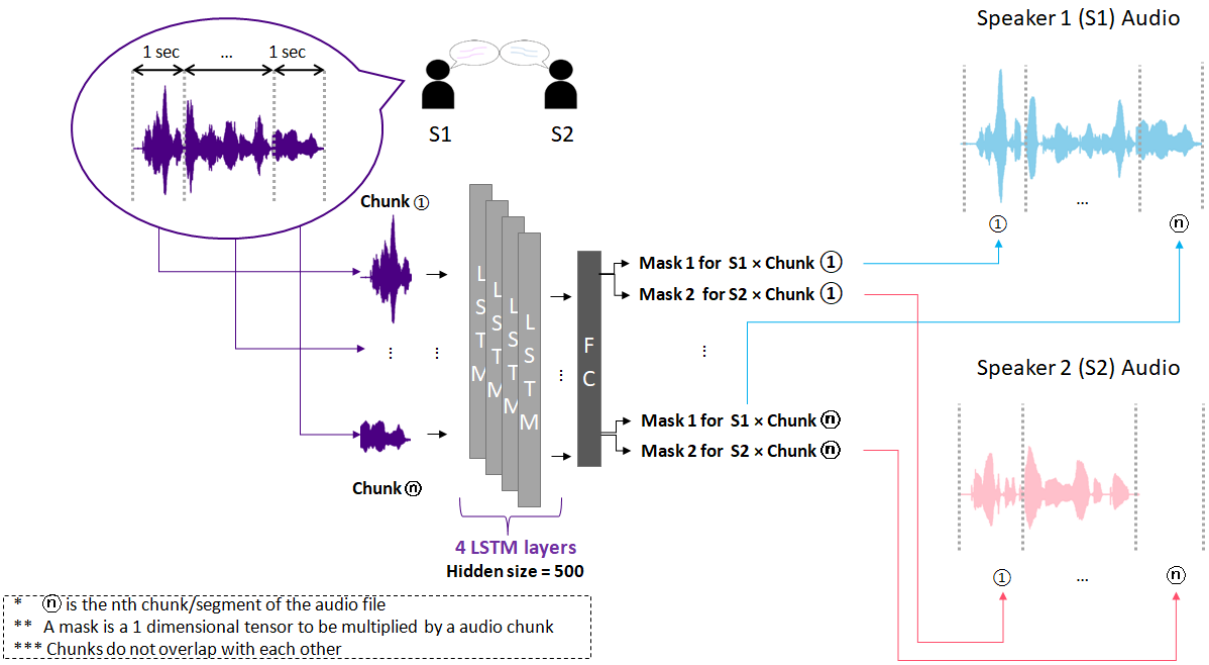

Figure 3. *Overview of the created model (LSTM and FC) and its inputs and outputs*

After the preprocessing, the actual network took the batches of audio segments as inputs. The network consisted of four LSTM layers with input_size=16000 and hidden_size=500 followed by a fully connected layer. The model took the given segment of the speech mixtures and created a pair of masks, each with the same shape as the input segment. The masks were then multiplied with the input segment to reconstruct two audio waveform tensors, one for each speaker. The model then concatenated the outputs of the input segments from the same overlay audio file into one audio sentence for each speaker represented as waveform tensors so that the audience could listen to the entire separated speeches from one overlay audio.

We used a loss function that combined two types of loss: Mean Squared Error (MSE) loss and Kullback–Leibler (KL) loss. Since the speakers are unlabelled, it is difficult to know which mask corresponds to which speaker. To solve this issue, we used the minimum loss between all the possible loss combinations of masks and ground truths. Reconstruction loss was also implemented to insure that the sum of the two outputs of the model add up to the original file. Reconstruction loss is defined

as □□□□□□□□□□(□□□□1 ∗ □□□□□□□□ + □□□□2 ∗ □□□□□□□□, □□□□□□□□). The final loss function was the sum of minimum loss and reconstruction loss.

This model was then trained for five epochs with the learning_rate=0.0001 and batch_size=64.

Baseline Model

Due to the inherent complexity of speech-separation, simple non-ML models or neural networks (e.g. only 1D fully convolutional layer) were ineffective on assessing the quality of the models based on experiments. Hence, we used a simple RNN model as the baseline model.

The model had one RNN layer (100 hidden units), followed by a fully connected layer with input_size=100 (hidden size of the previous layer) and the output_size=2*16000. Hence, the model took the batch_size number of overlay segments of tensors as input and outputted the masks for each speaker in a tensor with size batch_size*16000*2. It used Adam optimizer and MSE loss. Loss was calculated by summing the MSE of pred1 with output1, and pred2 with output2.

The model was trained on unsegmented datasets. Therefore, the overlay files were segmented during the training. It was trained for ten epochs with the learning_rate=0.001 and batch_size=32.

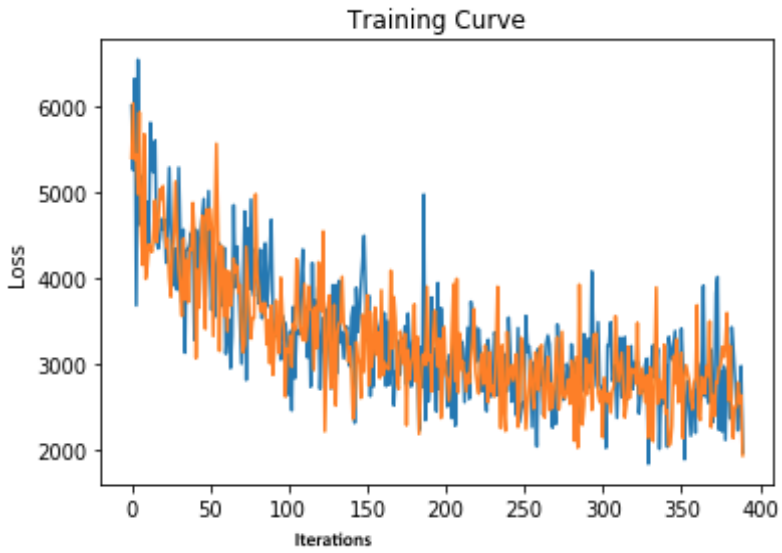

Figure 4. *Training curves (blue training and orange validation losses) of the baseline model*

Quantitative Results

Comparing loss values across all models proved to be an effective way to quantify and evaluate models' performance. However, since different loss functions resulted in different ranges of loss values, each model was trained and tuned using its own loss function. After each model obtained the optimal set of hyperparameters, all models were compared using the same test data loader with the same loss function, MSE loss. Although the model with the smallest loss value did not guarantee that the best performance, larger loss values were a good indicator of poor performance. By referring to the loss, we further investigated the models using qualitative metrics discussed in the next session. This quantitative metric also helped us focus on fewer well-performing models.

In order to understand how well a model was actually performing, we observed different loss values, such as minimum loss and reconstruction loss, on the test set as shown below. We introduced a new metric; separation. It represented the degree of separation between the two predictions by using criterion(prediction1, prediction2).

Table 1. *Comparison between baseline and final model with different loss functions*

	Baseline	**Final**
Min MSE Loss	6484.80	6398.35
Min KL Loss	3245.42	3611.82
MSE Reconstruction	596.01	7.69
KL Reconstruction	6472.95	7164.93
MSE Separation	625.75	17.73
KL Separation	1546.08	1758.68

Qualitative Results

Since the objective of this project was to separate two speeches, human evaluation would provide a better understanding about the output quality. Hence, we conducted test survey using 10 randomly sampled files from the test set, which were then separated using different models. Without knowing which model produced which outputs, five interviewees heard and ranked all sets of outputs based on overall noise level, degree of separation, clarity, etc. Based on the survey results, we discovered that humans were unable to distinguish the differences between the audio outputs generated by models with loss values ranging from 5000 to 7000. Hence, such differences in loss values were too small to have distinct audio differences when heard. Nevertheless, according to the survey, the LSTM model (shown above as ‘final’) with the lowest test loss was the top ranked model.

Discussion

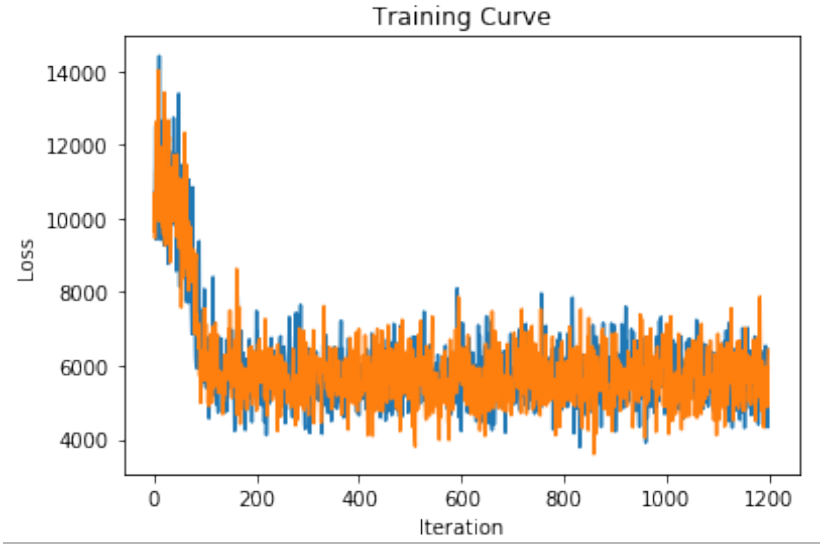

Figure 5. *Training curve (blue training and orange validation) for the final model with epoch=5, learning_rate=0.0001, batch_size=64*

The figure above showed the training curve of our best model. The model quickly reached its minimum loss, averaging around 6000. At this loss level, the model's predictions were two audio files, which were both extremely similar to the input overlay file and were not separated into individual speeches. By monitoring the metrics mentioned in the previous section, we determined that the initial drop in loss was the model learning how to output masks that did not result in noisy outputs. However, once the model reaches a loss of 6000, it stops learning. We believe that the complexity of the task and the simplicity of our model were the main factors contributing to such poor performance. We also believe that the model was unable to learn the complex temporal structure of audio that we hoped LSTM layers would be able to do.

Ethical Considerations

Because the model separated speeches without using other information of the speakers, there were no apparent ethical issues in the datasets with the focus on the United States and English. Nevertheless, as shown in figures 1 and 2, there were many more male speakers than female speakers. Since we randomly sampled to create the dataset, the gender and dialect distribution remained the same. This could potentially lead to a lower separation performance on female speaker's voice compared to male speakers.

Similarly, dialects that were less available in the dataset, such as "New England" and "New York City", could exhibit a worse performance.

Lastly, users of the model must be aware that the data used for training only contained English dialects from the US. Using the model on other languages or English accents might yield a much worse result.

Project Difficulty/Quality

Due to the nature of audio data, we spent much time on data processing and working with different audio file formats to ensure the least amount of information loss. We researched on how to convert SPHERE files to .wav files without losing or altering information. We also tried different representations of the audio content. Since STFT required computations of both magnitude and phase, which doubled the number of computations, we discovered that waveforms were computationally efficient more than a STFT. Ultimately, we generated a dataset of size 1GB, which required a long loading and training time.

Another major challenge we encountered was the complex patterns of audio data. To capture all the complicated data features, we created many architectures and tuned various hyperparameters. We created models using many different combinations of convolutional, RNN, LSTM, GRU, and fully connected layers. We also tested published neural networks that directly reconstructed split audios or producing masks and networks with two parallel output streams for each output audio[5]. Additionally, we examined many loss functions, such as MSE, Kullback–Leibler, reconstruction losses, Chamfer distance[6], and introduced our own criterion, combining weighted losses and penalties to better quantify the degree of separation.

In conclusion, the original file format and the nature of the audio data added extra complexity to the project. Even though the final model did not fully achieve the project objective, we still managed to design and develop models to incorporate the challenges mentioned above.

References

[1] D. Wang, Fellow, IEEE, J. Chen, "Supervised speech-separation Based on Deep Learning: An Overview." [Online serial]. Available: https://arxiv.org/pdf/1708.07524.pdf?fbclid=IwAR1QxfgmVxHa2APNnustn9crIhfimYvzOH7EIzJQqswHmlEO5mkG0FKfP_Q. [Accessed July 20, 2019].

[2] I. Mosseri, O. Lang, "Looking to Listen: Audio-Visual speech-separation," Google, 2018. [Online]. Available:

https://ai.googleblog.com/2018/04/looking-to-listen-audio-visual-speech.html?fbclid=IwAR0SnfFheZdfUiDg_dwH6nSIi4MYRrTAqT1DM2R9CixLhjwGx1GqulCoQFc. [Accessed August 11, 2019].

[3] Y. Luo, "TASNET: TIME-DOMAIN AUDIO SEPARATION NETWORK FOR REAL-TIME,

SINGLE-CHANNEL SPEECH SEPARATION," Department of Electrical Engineering, Columbia University, New York, NY, 2018. [Online Serial]. Available: https://arxiv.org/pdf/1711.00541.pdf. [Accessed August 11, 2019].

[4] P. Rémy, "The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus," philipperemy.github.io, Feb. 24, 2019. [Online]. Available: https://github.com/philipperemy/timit/blob/master/README.md. [Accessed August 12, 2019].

[5] Andabi, M. Kwon, "Deep Neural Network for Music Source Separation in Tensorflow," 2007. [Online]. Available: https://github.com/andabi/music-source-separation. [Accessed August

12, 2019].

[6] H. Su. CS 395. Class Lecture 17, Topic: "Seeing the unseen." Stanford University, Stanford, CA.