Automated Transcript Generation for Video Conferences

Nikolas Lee, Jia Wern Yong {nikolas, jayyong}@stanford.edu

MOTIVATION

With the rise of Deep Learning, speech recognition has shown great results in recent years. As people who use video conferencing software on daily basis, we aim to demonstrate the viability of generating meeting transcripts using speech recognition combined with a preprocessing step of identifying unique speakers in the meeting. If successful, participants in the meeting can focus on engaging fully in the conversation without spending effort on note taking.

DATA SET AND SETUP

Our main dataset for speech recognition comes from Google Research's Speech Commands Data Set[1] and TIMIT[2].

In a video conference call, audio can be captured from each user's computer (mic and system audio). The data from one conversation between N people hence comprises of N sets of audio for each speaker's in which the speaker's microphone is noticeably louder compared to the system audio containing the voices other N-1 speakers.

Our problem can be split into two main tasks. The first is a preprocessing step to identify which the parts of the conversation for every unique speaker. The second is performing speech recognition on the respective preprocessed audio.

UNIQUE SPEAKER IDENTIFICATION

Method

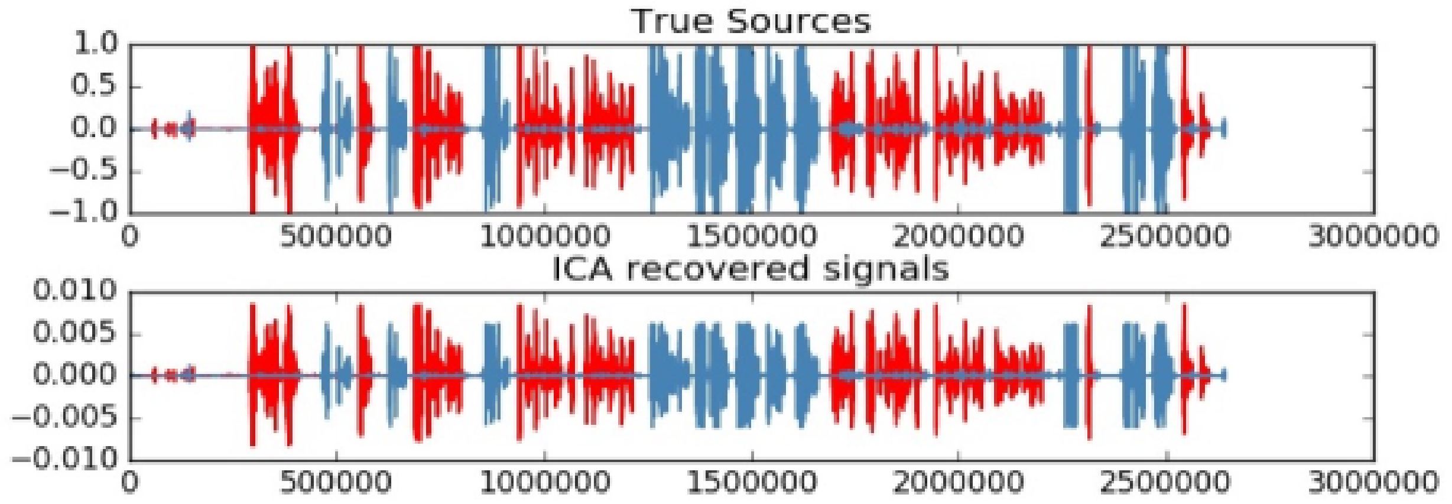
Independent Component Analysis

In order to help us to quickly perform ICA, we employed the FastICA library from sklearn decomposition[3].

Preliminary test: We first got sound data from three separate individuals, then mixed each one up with a 'fixed' mixing matrix A as a simulation.

Currently: Perform ICA on sound files from two speakers with mixed mic and system audio.

Results



DISCUSSION

The splitting of audio with ICA turned out well, resulting in sets of audio mainly containing the main speaker with the voices of the other speakers being inaudible.

While we could not get speech recognition to perform the way we had hoped, it can be seen that this preprocessing set improves the output for the purposes of transcript generation.

SPEECH RECOGNITION

Method

(Preliminary) Feature Extraction and Gaussian Naive Bayes Classifier: As a preliminary step, we calculated the MFCCs and ran a Gaussian Naive Bayes model on the features extracted from the MFCCs.

(Currently) CTC with BiRNN: We used Mozilla's DeepSpeech library, which implements a BiRNN CTC model as described in a Baidu Research paper[4]. We trained the model on different amounts of data and also tried a pre-trained model that we found.

Results

The model which trained on the most data, had a WER of 0.74. Feeding the preprocessed audio into a commercial speech recognition system resulted in slightly better results.

Currently, the results that we got from speech recognition are not satisfactory for our application. We expect the results of the speech recognition model to improve with more data.

FUTURE

We plan to obtain more data to improve our speech recognition system to reduce reliance on commercial software and also build an actual webapp for our application.

REFERENCES

[1] Google Research. Speech Commands Data Set. https://research.googleblog.com/2017/08/launching- speech-commands-dataset.html.
[2] TIMIT Dataset. (https://catalog.ldc.upenn.edu/ldc93s1)
[3] Scikit-Learn. Blind source separation using FastICA. http://scikit-learn.org/stable/autoexamples/decomposition/ploticablindsourceseparation.html.
[4] G. Alex. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. ICML, 2006.