Noise Detection and Cancellation

TEAM: BROGRAMMERS

TEAM MEMBERS:

KAVYA KHATTER

DAKSHA SINGHAL

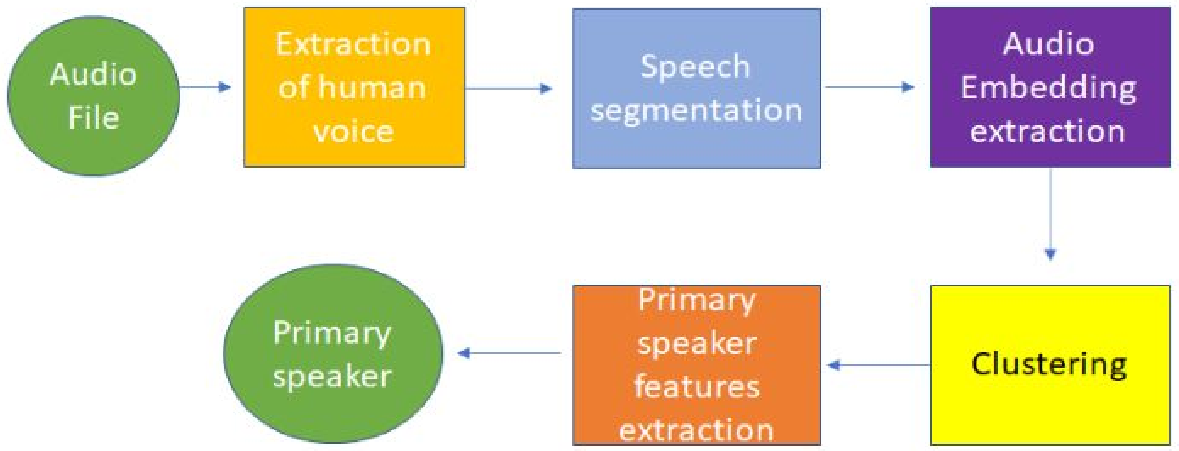
SHIVANGI RAJ

Overview

Identifying the primary speaker is a major problem nowadays. Given a noisy input signal, the aim is to filter out voice Interactions in Indian Houses & Neighborhoods noise without degrading the signal of interest. You can imagine someone using google audio search while a piece of music is playing in the background. In this situation, a speech denoising system has the job of removing the background noise to improve the speech signal. Besides many other use cases, this application is especially important for video and audio conferences, where noise can significantly decrease speech intelligibility.

In this project, we tackle the problem of cleaning the audio files as the first step towards finding the primary speaker using Deep Convolutional Neural Networks (CNNs) and AI.

Approach


Audio File
Extraction of human voice
Speech segmentation
Audio Embedding extraction
Clustering
Primary speaker features extraction
Primary speaker

This problem has 3-segments :

1. Removing the environmental noise from audio file
2. Checking if there is more than one speaker
3. Extraction of primary speaker audio

Removing the Environmental Noise from Audio File

Dataset

For removing the environmental noise we have used Urbansound8k dataset which contains small snippets (<=4s) of sounds. However, there are 8732 labelled examples of ten different commonly found urban sounds. The complete list includes:

0 = Wind

1 = car_horn

2 = children_playing

3 = dog_bark

4 = drilling

5 = engine_idling

6 = gun_shot

7 = jackhammer

8 = siren

9 = street_music

Given a noisy input signal, we aim to build an AI model that can extract the clean signal and return it to the user. Here, we focused on source separation of regular speech signals from ten different types of noise often found in an urban street environment using the essentials of Deep Learning

A deep learning model is used to take input audio and detect the type of noises present in the audio. Then, a 'noise reducer' is used to remove the similar kind of audio from the input file and creates a noise-free clean audio file.

Preprocessing

Preprocessing performed on the data includes:

- **Converting an audio file into a spectrogram**

  At first, it may seem a little strange to classify audio files as images. Images are 2-dimensional after all, and audio files have a single time dimension (with a possible 2nd dimension for channels, for example, stereo vs mono). Every audio file also has an associated sample rate, which is the number of samples per second of audio. If a 3-second audio clip has a sample rate of 44,100 Hz, that means it is made up of 3*44,100 = 132,300 consecutive numbers representing changes in air pressure.

- **Spectrogram to MFCC**

  The Mel-frequency Cepstral Coefficients (MFCCs) and the constant-Q spectrum are two popular representations often used on audio applications.

- **Feature extraction on Mel spectrograms**

  Mel spectrograms transform the frequency bins into the MEL scale. This greatly reduces the size of each transform as compared to the original.

- **Reshaping to 2D to CSV form**

  Audio data, in its raw form, is a one-dimensional time-series data. Images, on the other hand, are two-dimensional representations of an instant moment in time. For these reasons, audio signals are often transformed into (time/frequency) 2D representations.

- **train/test split and saved as CSV**

Modelling

The model based on Deep Convolutional Neural Network (DCNN) architectures included the following stages:

- Retrieving the data from test train split CSV files from the last step
- Reshape the data to One Hot to CNN required form
- Model formation and compilation
- Saving the model with test score

Observation and Results

Testing

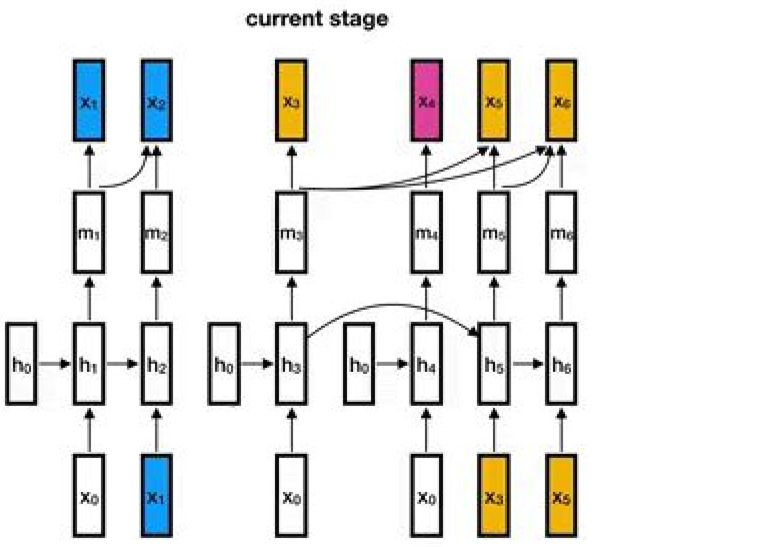
- Inputs audio file and Preprocess it
- Predicts the Noises present
- Removes corresponding noises using 'noise reducer'
- Saves the final WAV file

Training Accuracy: 97.1%	Training Loss: 0.07
Validation Accuracy: 79.5%	Validation Loss: 0.4

Assumption

SNR should not be less than1.

Checking if there is more than one speaker



Generative path 1: Switch to speaker BLUE

Preprocessing

After Segmentation and removing non-human audio, the next step is to extract features i.e. identify the components of the audio signal that are good for identifying the linguistic content and discarding all the other stuff which carries information like background noise, emotion etc.

- **MFCC**

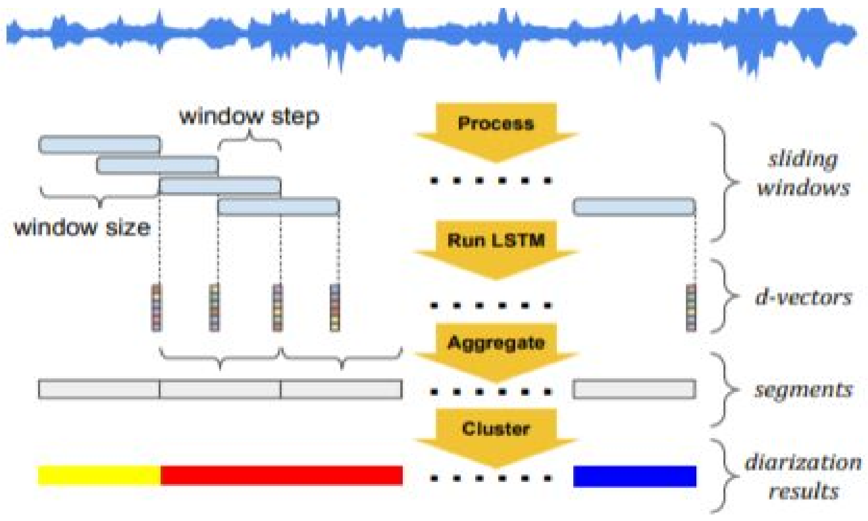
  Representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear MEL scale of frequency.

- **D-Vectors**

  A speaker identifier RNN is trained using the utterances from a development set. When the speaker identifier RNN is trained, the input acoustic feature is concatenated with context frames, meaning that a few of the previous frames and the ensuing frames are also placed into the speaker identifier RNN. Context frames are concatenated because a single frame does not have sufficient information about the speaker.

- The output layer is removed. Then, enrollment and test utterances are forward-propagated through the speaker identifier RNN to the last hidden layer.

- The activations of the last hidden layer are used as d-vector for the given input. However, this d-vector is extracted from a few frames and is therefore considered as being frame-level. Only one d-vector is extracted from one utterance— obtained by element-wise averaging frame-level d-vectors extracted from the same utterance.
- Speaker models are then composed by element-wise averaging the enrollment utterances' d vectors for a given speaker. Finally, cosine similarity scoring is applied between a speaker model and a given test d-vector and similarity is measured.



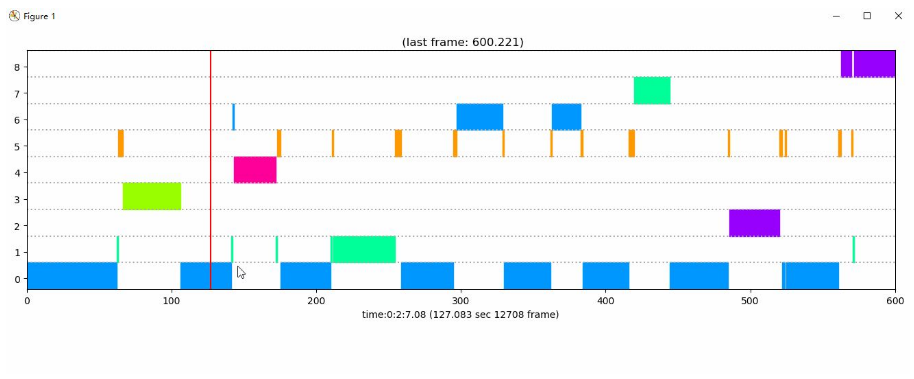
Modelling

- The purpose of this stage is to associate or cluster segments from the same speaker together.
- The clustering ideally produces one cluster for each speaker in the audio with all segments from a given speaker in a single cluster. The predominant approach used in diarization systems is hierarchical, agglomerative clustering with a BIC based stopping criterion.
- Extracted d-vectors are clustered to determine which segments were produced by the same speaker. Clustering was based on cosine similarity :
- After clustering, we compute one d vector for each cluster and reclassify the individual segments The process is repeated until convergence.

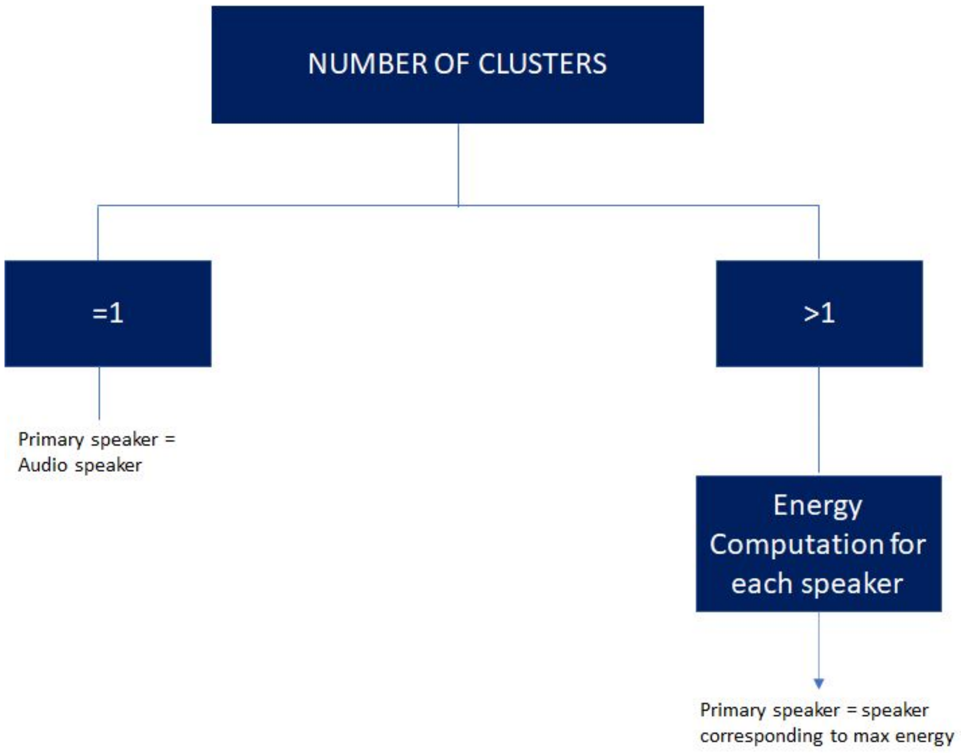
Observation and Results

Sample Output format:

```
========= 0 ========
0:00.288 ==> 0:04.406
0:07.699 ==> 0:16.461
0:33.921 ==> 0:35.8
========= 1 =========
0:04.406 ==> 0:07.699
0:16.461 ==> 0:19.594
0:30.371 ==> 0:33.921
0:41.19 ==> 0:44.185
========= 2 =========
0:19.594 ==> 0:30.371
0:35.8 ==> 0:41.19
```



Extraction of primary speaker audio


NUMBER OF CLUSTERS
=1
>1
Primary speaker = Audio speaker
Energy Computation for each speaker
Primary speaker = speaker corresponding to max energy

Energy

Once we had the Speaker turns from Speaker Diarization, speaker energy was extracted using the root mean square of the amplitude of the audio signal over a sliding time window for each audio track. Speaker turns were then chunked and added over the particular speaker. Speaker with the maximum speaking energy was considered the primary speaker. This method was applied on the assumption that the Primary Speaker is the one closest to the microphone.

Automatic Speech Recognition (ASR)

After retrieving the primary speaker from the audio file, ASR was implemented to get the text.

We have assumed that the speaker speaks in Hindi and therefore the text is produced in that language.

Results and Conclusion

We have used the WER(Word error rate) as an evaluation metric for our model and Obtained a score of 37.5%.

$$WER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}$$

where

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- C is the number of correct words,
- N is the number of words in the reference (N=S+D+C)

Acknowledgement

We would like to thank Flipkart and PES University for giving us this opportunity. We explored the domain deeply and enjoyed working on this topic.