Unsupervised Speaker Identification

Thivakkar Mahendran
University of Massachusetts Amherst
tmahendran@umass.edu

Varun Prasad
University of Massachusetts Amherst
varunanantha@umass.edu

1. Abstract

Filled by Varun

2. Introduction

A voice recorder makes it super convenient to take notes or preserve the minutes of a meeting. In order to make the experience better, there are tools that can be used to automatically convert speech to text. One area where this tool currently fails at is identifying the speaker. The problem that we are trying to solve is to identify the speaker. Our ultimate goal is to build a project where we are able to record a meeting and which particular person speaks at a specific time and what they speak.

It would be cumbersome and time consuming to record each speaker's voice beforehand and train a speaker recognition model on the speakers' voice. The goal is to make this tool predict the speaker without prior training on the speaker's voice. This would be an unsupervised learning project.

For example, if two speakers were talking in a meeting we would like our tool to give an output:
Speaker1: Hey, How are you?
Speaker2: I'm doing well.
Speaker2: I was able to complete the tasks that we talked about last week
Speaker1: That's great.

These two speakers' voices have not been trained before but still the program would identify the two speakers just by getting the features from their voice.

3. Background/Related Work

Filled by Varun

4. Approach

Real-world applications of such a system would require an unsupervised learning approach to the problem since it is extremely difficult to re-train the model with the voices of all the speakers in every particular situation, and also there would be no labels to train the model on. It is also difficult to use an entirely unsupervised approach during the training phase because there would be absolutely no way to know whether the model is trained properly. Hence, evaluation of the model would become almost impossible.

Because of this, we have chosen to go with a hybrid approach where we train the model using a supervised approach over labelled data, and then use an unsupervised approach for the final classification. The first phase was done by building a fully-connected neural network and training over the 8-speaker dataset and random speaker dataset which will learn to classify the different speakers.

The second phase was implemented by cutting off the final classification layer of the neural net, and getting the embedding vector. The embedding vector at every interval, in our case 1 second, was compared to previous embeddings using a distance metric and a clustering algorithm to decide if the speaker is the same or not.

4.1. Dataset

The dataset that we intially planned to use was VoxCeleb, which is a large scale audio-visual dataset of human speech. The dataset contains 7,000+ speakers, 1 million+ utterances, and 2,000+ hours of audio. The total size of the dataset was around 250 GB. The dataset was really huge in terms of computational complexity and also space required to store and train the model. After weeks of trying to use the dataset, we decided to build our own dataset.

We decided to create our own dataset so we could tailor the dataset to exactly suit our project. We built a pipeline to scrape audio from YouTube videos, and then split a whole audio into chunks of 1 second audio clips. We decided to create 1 second audio clips because when this model is getting used in the real world, we want to recognize the speaker instantly with as little of a delay as possible. According to the article [1] published by virtualspeech, an average person speaks about 150 words per minute, which is about 2.5 words per second. This is enough to extract useful features from the person's speech.

4.1.1 Dataset of 8 speakers

The YouTube videos that we chose to include in our dataset were speeches/monologues from celebrities. We thought this would be the best way to build a labeled dataset of audios of different people. The dataset 1 includes 7 celebrities (Obama, Hillary, Ivanka, Trump, No Speaker, Modi, Xi-Jinping, and Chadwick-Boseman) and one "no speaker" class which includes multiple background noises without anyone speaking. We included the "no speaker" class so that the model can recognize when no one is speaking. We wanted the dataset to be as diverse as possible, that is why we included speakers of both genders, different races, and also different languages. The current dataset has a size of 4.1+ hours.

	Gender	**Language**	**Race**	**Length**
Obama	Male	English	Black	19.5 mins
Hillary	Female	English	White	57.5 mins
Ivanka	Female	English	White	17.9 mins
Trump	Male	English	White	41.6 mins
Modi	Male	Hindi	Asian	32.4 mins
Xi-Jinping	Male	Chinese	Asian	11.18 mins
Chadwick	Male	English	Black	27.1 mins
No Speaker	N/A	N/A	N/A	39.6 mins

Table 1. Details of the 8 speaker dataset

4.1.2 Dataset of random speakers

We wanted to experiment on the quality and quantity of our dataset to see the impact on the speaker recognition model. The dataset that we created before only had 8 speakers but each speaker had multiple videos combining to more than 30 mins each. In the dataset of random speakers, we got audio files of length 2-3 minutes of **50 speakers**. In this dataset we increased the amount of speakers but limited to 1 audio file (2-3 minutes) per speaker.

The audios of the speakers were obtained from random youtube videos and voice recordings. We only chose audios which had only one speaker speaking in the whole audio file and checked if there were no background music or sound. We limited the audio file to 2-3 minutes so we could have multiple speakers. The table 2 is a snippet of the youtube videos used to create the random speaker dataset.

YouTube Title	**Length**
How to evaluate expressions with two variables	2:04
2020 Rock & Roll Hall of Fame Speech	2:31
A look at U.S. President-elect Joe Biden	3:32
Open Office Math	2:46
How to Get Stuff Done When You Have ADHD	4:45
How Online Math Tutoring via Skype Works	2:38

Table 2. A snippet of random speaker dataset

4.2. Supervised speaker recognition

4.2.1 Features

The first step to build a supervised speaker recognition model is to extract useful features from the audio files. These features would be the training data. We looked at multiple research papers to choose the best features for speech recognition. The features we chose, based on [2] and [3], to extract from the audio clips were:

- MFCC (Mel-Frequency Cepstral Coefficients): coefficients used to detect the envelope of audio signals. Sounds produced by humans can be accurately represented by determining the envelope of the speech signal.

- Zero-crossing rate: Number of times in a given time interval/frame that the amplitude of the speech signals passes through a value of zero. This feature is used to distinguish between periods of voiced and unvoiced sounds.

- Spectral roll off: Measure of the amount of the right-skewedness of the power spectrum. That is, the roll-off point is the frequency below which 85% of accumulated spectral magnitude is concentrated.

4.2.2 8 speaker recognition model

To build the 8 speaker recognition model we experimented with different layers and hyperparameters to get us the best result. We finally chose a 5-layer network. The first layer had the input size of 128 and the activation we chose was ReLU. The second layer has 64 as it's input with ReLU activation. The third layer has 32 as it's input with ReLU activation. The fourth layer has 16 as it's input with ReLU activation. The final layer has a size of 8 with softmax activation. The optimizer used for the model was Adam with a learning rate of 3e-4. The loss was categorical crossentropy and the metric to analyze the model was accuracy. The total trainable parameters were 123,768.

The figure 1 represents the model built to identify the 7 speakers and 1 "no-speaker" class.

4.3. Using embedding vector and clustering for unsupervised speaker recognition

4.4. No speaker recognition model

4.5. Live speaker recognition

A python script has been created to get a live stream of audio data from the computer, process every 1 second audio clip by extracting features on the audio and then the features get passed to the model to get the last layer embedding vector. The new embedding vector is added to a dictionary.

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_5 (Dense)              (None, 11847, 128)        112768
_________________________________________________________________
dense_6 (Dense)              (None, 11847, 64)         8256
_________________________________________________________________
dense_7 (Dense)              (None, 11847, 32)         2080
_________________________________________________________________
dense_8 (Dense)              (None, 11847, 16)         528
_________________________________________________________________
dense_9 (Dense)              (None, 11847, 8)          136
=================================================================
Total params: 123,768
Trainable params: 123,768
Non-trainable params: 0
_________________________________________________________________
```

Figure 1. 8 speaker model

This dictionary contains all the previous embedding vectors. That dictionary is then used to run the clustering algorithm to find which cluster that last added embedding vector belongs to. So every 1 second, the program will output the current speaker or "no speaker" if no one is talking. This is how an unsupervised speech recognition system works in real time.

5. Experiment

Filled by Varun

6. Conclusion

Filled by Varun

References

[1] Barnard, Dom. "Average Speaking Rate and Words per Minute." VirtualSpeech, VirtualSpeech, 20 Jan. 2018, virtualspeech.com/blog/average-speaking-rate-words-per-minute.

[2] Sharma, Usha, et al. "Study of Robust Feature Extraction Techniques for Speech Recognition System." 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015, doi:10.1109/ablaze.2015.7154944.

[3] "Analytical Review of Feature Extraction Technique for Automatic Speech Recognition." International Journal of Science and Research (IJSR), vol. 4, no. 11, 2015, pp. 2156–2161., doi:10.21275/v4i11.nov151681.