Unsupervised Speaker Identification

Thivakkar Mahendran
University of Massachusetts Amherst
tmahendran@umass.edu

Varun Prasad
University of Massachusetts Amherst
varunanantha@umass.edu

1. 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.

2. Problem Statement

2.1. Dataset

The dataset that we 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. At the end of the project, we are planning to experiment on the seconds of each audio clip to see its effect on the accuracy of the model.

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. To start with, the dataset 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-Boseman	Male	English	Black	27.1 mins
No Speaker	N/A	N/A	N/A	39.6 mins

Table 1. Details of the dataset

We are planning to add more diverse speakers to make the model predict better. We will experiment with the dataset to see the impact on the model. For example, we will train the model on all male voices or only English speakers to see the impact on the model.

2.2. Expected Result and Evaluation

The project is split into two parts, the first part is training the model to recognize and predict the 8 different speakers from the dataset and the second part would be to modify the model to recognize different speakers who have not been trained before. We want the model to differentiate different speakers and label each speaker as "speaker#". 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 using the features to predict the speaker by using the model. So every 1 second, the program will output the current speaker or "no speaker" if no one is talking.

For the first part of the project we will be performing the five-fold cross validation to train and evaluate the model. We will be evaluating the model by getting the accuracy, precision, and recall. We would like the model to perform at least 90% accuracy on the test dataset so the model would work better in the second part of the project.

Evaluating the second part of the project would be slightly harder because we don't have a labeled dataset of speakers talking at every timestamp. We either need to build our own dataset to make it work for this specific use case or we have to manually evaluate the model. One possible way to evaluate is to play an interview video between two or more people and physically check if the model predicts the speaker speaking at that very point of time.

We are expecting several factors to impact the accuracy of the model. For example, the speaker's gender, age, race, language, dialect of a language could affect the accuracy of the model. Multiple speakers speaking at the same time might also impact the accuracy of the model. These are some of the factors that we are interested in evaluating. These results would help us improve the model in the future based on its weakness.

3. Technical 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 will be done by building a fully-connected neural network and training over the 8-speaker dataset specified in section 2.1, which will learn to classify the different speakers. The second phase can then be implemented by cutting off the final classification layer of the neural net, and considering the embedding of the penultimate layer: The embeddings at every interval can be compared to previous embeddings using some distance metric and a classification method, like k-nearest neighbors, can be used to decide if the speaker is the same or not.

Feature selection is an important part of any deep learning project. 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.

We are continuing to experiment with various other features that are useful in audio processing, like spectral entropy, pitch and spectral flux, to see if the model would benefit from including any of them. We are also experimenting with other classification methods that can be used instead of k-nearest neighbors which might allow the model to better distinguish between different speakers.

4. Preliminary Results

For initial experimentation, we have constructed a 3-layer network (layer 1: size=12, activation=ReLU; layer 2: size=8, activation=ReLU; output layer: size = # of speakers (8), activation=softmax). Using an Adam optimizer and categorical cross-entropy loss, we are able to achieve around 75-80% accuracy over our dataset of 8 speakers. Our model has a total of 10,748 trainable parameters out of which layer 1 uses 10,572, layer 2 uses 104 and layer 3 uses 72 parameters respectively. We will continue to experiment with different network architectures and optimizers to see how they influence the results.

Table 2 shows the confusion matrix for the different classes (speakers): Obama(1), Hillary(2), Ivanka(3), Trump(4), No Speaker(5), Modi(6), Xi-Jinping(7), and

	1	2	3	4	5	6	7	8
1	0.735	0.016	0.033	0.022	0.000	0.010	0.058	0.124
2	0.017	0.924	0.000	0.001	0.003	0.055	0.000	0.001
3	0.010	0.000	0.988	0.002	0.000	0.000	0.000	0.000
4	0.014	0.006	0.002	0.805	0.000	0.166	0.006	0.000
5	0.000	0.757	0.000	0.000	0.243	0.000	0.000	0.000
6	0.004	0.079	0.000	0.081	0.000	0.832	0.002	0.002
7	0.001	0.000	0.000	0.001	0.003	0.000	0.945	0.049
8	0.004	0.000	0.000	0.000	0.001	0.000	0.004	0.991

Table 2. Confusion matrix

Chadwick-Boseman(8). It is seen from the confusion matrix that the model is able to do a good job predicting each of the speakers, but it falls short when it comes to predicting the ”No Speaker” class. The model is incorrectly predicting the audio clips where there is no speaker as ”Hillary”. We intend to explore further why this is happening, and aim to improve the model performance by either altering the network layers or adding more training data for the ”No Speaker” class, or possibly both.

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.