Speaker Identification from Audio using K-Nearest Neighbors

Emmanouil Lykos

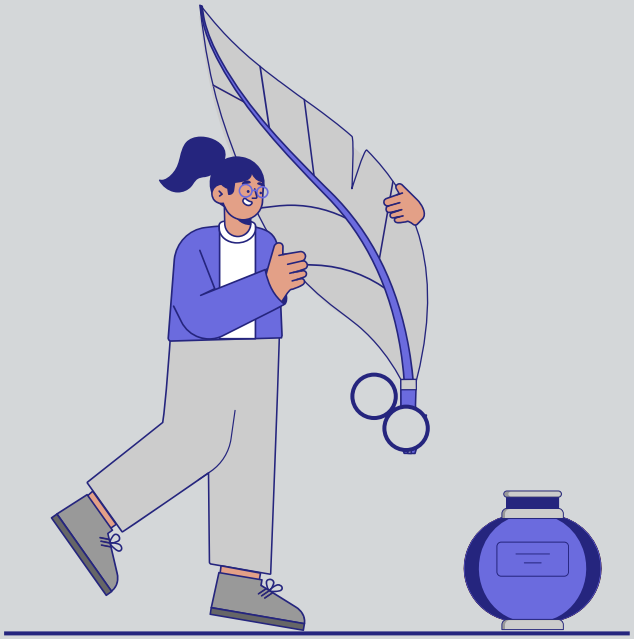
Table of contents

01.
Introduction

Speaker Identification

Introduction

- Speaker Identification is the task of identifying the speaker at some given point in time of an audio or video segment.
- Can be used in order to annotate shows or podcasts and also to provide info in subtitles on who is speaking in addition on what's being said.
- In this presentation, the task will be handled as closed-set classification task, but it would be more appropriate to be handled as an online open-set classification problem.

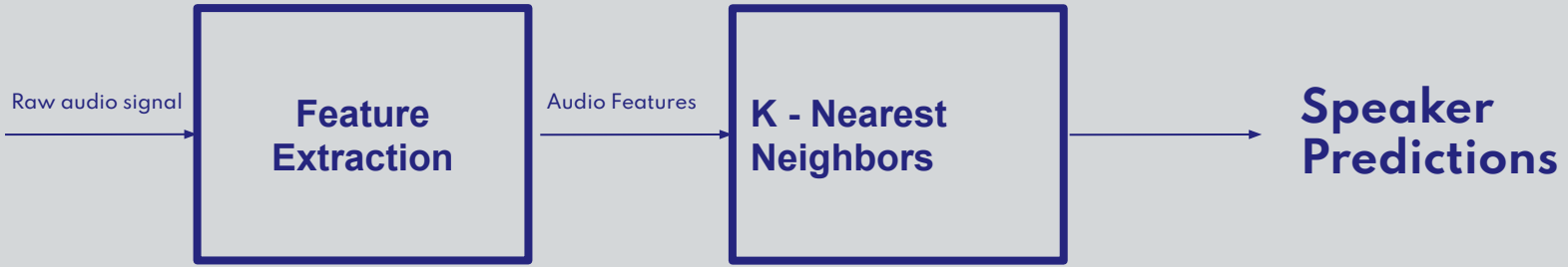


02.

Methodology

Methods to convert audio signals to feature vectors and classification method

Speaker Identifier's Pipeline



Audio Features-Short-Term Features

- pyAudioAnalysis Library's, Short-Term Features are used which are the common time and spectral domain features that can extracted from an audio signal.
- This returns a feature vector of size 64 of each time window of the signal.
- Then, each instance is represented by the mean vector of those.
- Finally, we remove the constant features and we normalize each vector.

Audio Features-Wav2Vec2

- Wav2Vec2 is a Transformer model that is like BERT that extracts representation from each time window of the signal.
- The pretrained version on 60,000 hours of unlabeled audio from *Libri-Light* dataset was used.
- This pretrained model returns a feature vector of size 1024 of each time window of the signal.
- Then, each instance is represented by the mean vector of those.
- Finally, we only normalize each vector.

Classification Model

- Now that we have our features set, we are ready to define the speaker identification model.

- Our model is just a simple K-Nearest Neighbors Classifier that works with the following way when it gets an instance to classify:
  - Find the k Nearest Neighbors of that instance.
  - The predicted speaker is the most prevalent speaker in those k nearest neighbors.

- This model was proposed because it was easy to understand what and why we do what we do.

03. Experimental Evaluation

Experimental Setting

- The dataset used for conducting experiments was the VoxCeleb 1 that contains samples from 1251 speakers.
- Due to computational limitations we only took samples from 250 speakers.
- The split of the dataset into train and test was done using Stratified Cross-Validation Split because we want to have on test set instances of all speakers.
- The accuracy of the different versions of our model, in terms of number of nearest neighbors we find, is shown in the next slide

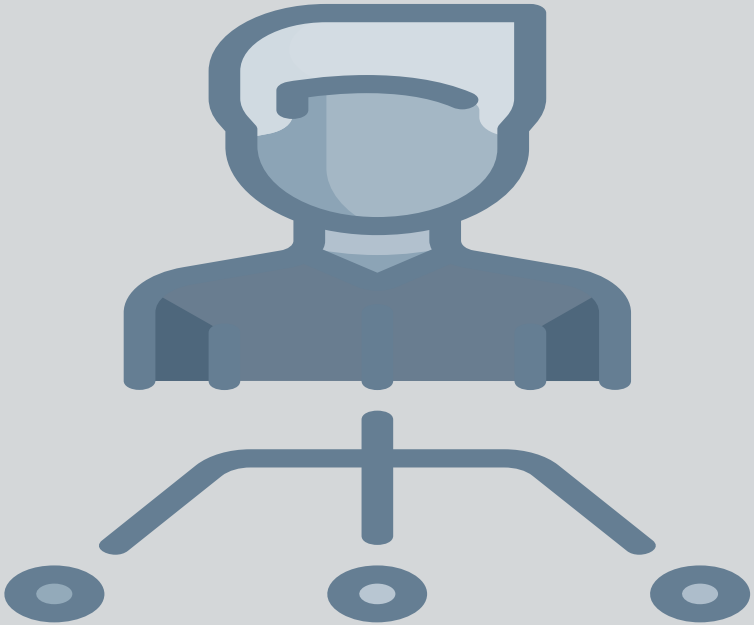
Accuracies on Test Set

Neighbors Number	Short-Term Features	Wav2Vec2
1	0.5484	0.3659
5	0.5173	0.3027
10	0.5068	0.3010
25	0.4634	0.2717
50	0.4132	0.2433
100	0.3547	0.200
200	0.2910	0.1632
350	0.2440	0.1323

Results Discussion

- Initially, it is clear that short-term features perform better than Wav2Vec 2 Features.
- Testing offline other classifiers we did not manage to yield better performance.
- Finally, the higher accuracy is yielded when we take into consideration only the nearest neighbors. This was not expected but this might probably happen because in that case is impossible to encounter any ties, hence the classifier does not do any random choices.

03. Conclusion & Further Work



Conclusion

- In this project we presented two Audio Feature Extraction techniques along with a classification model.
- We found out that for the nearest neighbors model the simple short-term features perform better than the other features when considering only the Nearest Neighbor with accuracy equal to 54%.
- Personally I believe that the predictive performance of the model is pretty decent for such a simple approach but still there is room for improvement either by generating another features or to look at completely different approaches.

Further Work

- We can use different Feature Extraction techniques like Autoencoders.
- This gives us the advantage to have representations of variable size which can be configured by us.
- We can also do the following Neural Network approaches:
  - Use LSTM to pass sequentially the extracted features that we mentioned before.
  - Use Convolutional Neural Networks that can take a spectrogram as an input and then predict the speaker like how is done on VGGVox.

Thank you for your time

Any Questions???