Speaker identification app using Deep learning and "VOX" celebrity dataset

Reinis Nudiens
LIACS, Leiden University, Niels Bohrweg 1
Leiden, 2333 CA
reinisnudiens@gmail.com

Speaker identification system based on existing project with different dataset was developed, with graphical interface and option to run identification test using microphone's input to check which person is speaking. Option to test the performance using test files from dataset was also implemented. Results, however, are worse than expected, meaning there is unsolved problem in the implementation.

Keywords: Speaker Identification; Deep Learning.

1. Introduction

Speaker identification is a system that identifies, which person, out of registered people, is speaking, based on a voice sample and it's characteristics. Not to be mistaken with speech recognition, which is speech-to-text system. Speaker recognition is a task that has been an actual problem for at least forty years. There are text-dependent and text-independent systems, that are trained on samples of people with or without pre-defined text that each person has to record while speaking out loud. Although this is one of the weakest forms of bio-metric forms of authentication it can also be used to authenticate user using the voice by setting certain similarity threshold, mostly this requires system to be text-dependent to increase the accuracy.
Applications of speaker identification are applicable anywhere that requires to identify someone from list of people, also could be used hand-in-hand with speaker verification to first find top people with highest similarity and then apply extensive algorithms to verify that it is the person speaking.

2. Related and previous work

Because of the fact that these systems date back to last century, there has been numerous papers and advancements in this field, which are also made in speech recognition and speaker verification fields. These similar fields use the similar principles, therefore advancements in one often means advancement in other.

3. Goal

Goal that is attempted to achieve with this project is creating computer application with graphical user interface that has option to take input of the microphone and try to identify which speaker is speaking, using deep learning methods. It also was decided to include option to run tests with dataset split in training/testing subsets.

4. Dataset

The dataset chosen for this project is VoxCeleb by Oxford university, that is dataset consisting of human speech clips extracted from Youtube interviews, see Ref. 1. More than 7000 speakers make up no fewer than one million voice clips totaling 2000 hours of data, see Fig. 1. This dataset is widely used in various fields of science, since it also includes actual video part where the person is talking.b Therefore applications are not limited to speaker identification/verification, with speech separation, emotion recognition, face generation projects and others showcased to the world.

Since the voice samples are cut out of a video, there is noticeable noise in large part of the samples. It would be wise to try and filter out the noise in order to improve the results.

While this dataset has framework for speaker identification, this project only uses the audio samples for 1221 speakers (subset of large dataset) with near 135 000 samples and 36.5GB in total, which is a subset of the larger dataset.

For one to acquire this dataset it is needed to apply online at official website describing intended usage and one's role. Then username and password is provided via email to be able to download it from official server.

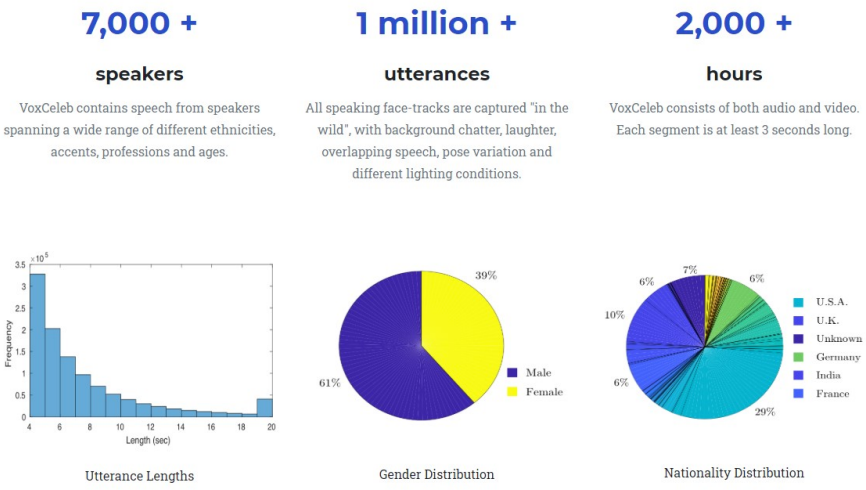

Fig. 1. VoxCeleb dataset statistics, note that subset of whole dataset was used

5. Pre-processing

Pre-processing is done in multiple steps as the dataset is in different format than the one used in project this project is based on. One of the most important differences was that file naming and subfolder system was different, for example Speaker_id/videoURL/recording_id.wav versus train/Speaker_id/recording_id.p, notice that file format is also different, meaning latter being stored as Python Pickle .p file, which contains log filterbank energies of the .wav file.
Restructuring of files and folders was done also using python, which was handy to also split the dataset into train and test sets in the ratio of 9:1 respectively. For extraction of log filterbank energies python_speech_features is used (see Ref. 3) and then saved as a pickle file while also saving speaker id in the pickle itself.

6. Deep Learning model

Deep learning model is based on project by Youngmoon Jung at KAIST, South Korea, see Ref. 2. That project uses smaller Korean-speaking dataset containing 3GB of data. Model contains a residual neural network or ResNet, which as an input takes log filterbank energies. Model is written in Python using PyTorch. After training, embedding or d-vector for each speaker is extracted from the last hidden layer.
Training accuracy was 92.51% as reported by PyTorch after 11 hours of training on GTX1070 graphical processing unit.

7. Results

An executable computer application was created using Python. User interface was built with library called Tkinter. With functional test option that tests one random speaker's recording and tries to predict out of 100 speakers (this was chosen trial and error, because higher number of speakers in the test equal lower accuracy). The process of comparing is quite trivial, compares the random test file's log filterbank to all speaker's embeddings by calculating cosine similarity and highest similarity score's speaker id is returned as predicted speaker.
However, only 26% accuracy over 100 randomly ran tests was achieved. While more speakers in the subset led to even worse results.

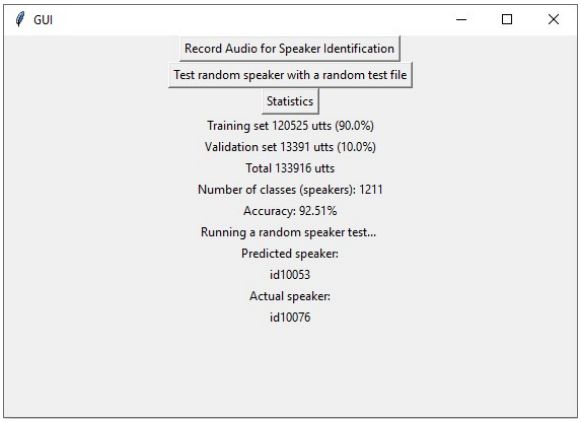

Fig. 2. End result of executable application in action

Another feature implemented was recording of microphone's input, which then is converted to log filterbank and afterwards compared against speaker embeddings by calculating cosine similarity. Again highest cosine similarity equal highest likeliness of recorded speaker is predicted correctly. This feature turned out to be useless with all the noise from microphone, accuracy being zero.

8. Conclusions

The goal can be considered achieved, but unfortunately, the results are weak, which might indicate that there is a problem in the code or that the noise from the dataset is making too much distortion. It is possible to hear other people talking in the background and other noises in the dataset, which undoubtedly affect the results, but the question is, if it is the only weak part of the chain in this project.
Since source project was done in Korea, it is in Korean language, making it hard to understand. Problem might be somewhere in the implementation as it was not completely clear how the source project needed to be adapted, because of this language difference.
To find root of the problem one should try to adjust the code so that it outputs top 3 most similar speakers, and if the actual speaker is mostly out of the top 3, the difference is enormous. This could mean there is some major problem with the implementation.
The ResNet possibly need to be adjusted.

References

1. *VoxCeleb dataset http://www.robots.ox.ac.uk/ vgg/data/voxceleb/*
2. *Speaker recognition project https://github.com/jymsuper/SpeakerRecognition_tutorial*
3. *Python speech features library https://python-speech-features.readthedocs.io/en/latest/*