CS698 Machine Listening Final Project Proposal:

Conformer-based Speaker Recognition Model

Ethan Oh
eo238@njit.edu

Overview

This project's objective is to build a speaker recognition model based on the conformer model proposed in [1]. Conformer is an expansion on the Transformer model proposed by [2], where a convolution mechanism is integrated to the Transformer. The Multi-head Attention networks give the Transformer good global awareness. With the addition of a Convolution network, the Transformer gains better awareness of local relationships. I seek to use these features to our benefit as the idea is to see if the model can learn the unique relationships between component frequencies of a voice to identify the speaker. Consequently, a preprocessing step of frequency decomposition using discrete Fourier transforms is a major component of this model.

Motivation

Speaker recognition is a very useful tool with many applications. Smart devices like Amazon's Alexa can employ it to distinguish user preferences. A custom menu for the recognized user can be loaded for quick access to "favorited" or "frequently used" apps. Even responding to the recognized user by name can add a stand-out level of intimacy with the device not achievable with the majority of other products we use. User-based restrictions can also be automatically deployed in such cases where a guardian does not wish a child to access certain apps or features of the device.

Speaker recognition is also an interesting tool when it comes to speech recognition in multi-speaker settings. The motivation behind this project is to build a speaker recognition model that can eventually be able to identify multiple speakers simultaneously speaking. Limitations in time and skill may not allow this project to achieve this, but we endeavor to at least build a model that can identify individual speakers in isolation as a first step.

Background

Papers I will use for reference:

[1] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, R. Pang, “Conformer: Convolution-augmented Transformer for Speech Recognition” arXiv:2005.08100 [eess.AS], May 2020
[2] F. Ye, J.Yang, “A Deep Neural Network Model for Speaker Identification” *Appl. Sci.*, 11,3603, April 2021
[3] X. Xiang, S. Wang, H Huang, Y. Qian, K. Yu, “Margin Matters: Towards More Discriminative Deep Neural Network Embeddings for Speaker Recognition” arXiv:1906.07317v1 [eess.AS], June 2019
[4] S. Konam, S. P. P. Selvaraj, “Deep Learning for Speaker Recognition” [Online Serial] Available: https://saiprabhakar.github.io/files/lstm_speaker.pdf. [Accessed Oct 10, 2023]

https://www.mathworks.com/help/audio/ug/speaker-identification-using-pitch-and-mfcc.html

Commercially available references:

There are several speaker identification products available on the market today. Google offers one as a Service as a Product via their Google Cloud and is used by companies as Avaya and Genesys for their customer service platform. Microsoft has a similar offering via their cloud, Azure.

Proposed Approach

The model will take speech audio .wav files from the Arctic training dataset as input and preprocess them using discrete Fourier transform for frequency decomposition via Numpy’s fft function. The grouping of frequencies per .wav file will be formatted as lists with their speaker id labels, where each pairing of list and label will be a row in a .csv format saved onto a HDF5 file using Pandas. This step is to decrease computational overhead and circumvent the model from conducting frequency analysis for the same .wav file at each epoch. The HDF5 file will be used to provide input to a Conformer model (built using PyTorch) for training. The intuition employed by the model is as follows:

$$S_i = \sum_{n=0}^{N} f_{in} \bullet W_i$$

S is a speaker of set I, where each S can be identified by the sum of his component frequencies f_i. N is a hyperparameter and is the total number of frequencies I would like the model to use to

define S_i. W is the learned weights for each frequency for each S_i. The assumption is that every S_i is unique in all of S.

[4] Outlines a method for feature extraction, which I will use for processing the frequencies. Here, Mel Frequency Cepstral Coefficients (MFCC) and their differentials and accelerations were derived from the sound waves samples. MFCC is a cepstral representation of the sound wave, where the frequencies are equally spaced on the mel scale and better represents the human auditory system's responses. MFCC coefficients were extracted from every frame using filter-bank channels and the differentials and accelerations of the MFCC coefficients were calculated. These values were combined to form the input feature vectors for the model.

Before processing frequencies into MFCC, I will experiment with limiting the frequency range to 275 hz to 3.5k hz, compliant with telephony narrowband. This is an effort to reduce the data size and to minimize processing while maintaining a usable frequency range. If necessary, another optimization step can be to apply a voice activity detector to remove non-speech frames, as described in [3].

Datasets

Google's Fleura dataset was considered a candidate as it is a compilation of audio speech clips totalling 5gb. But due to its binary labeling of categories as male or female, it may not be usable without relabeling each audio clip with a speaker identifier. Carnegie Mellon University has developed the Arctic dataset, a collection of speech clips encoded in .wav format and available on Kaggle. Arctic is smaller than Fleura (at 1gb) but is pre-classified into 19 speaker id's.

Evaluation

Arctic provides a test dataset, which I will use for evaluation. The test dataset is a random compilation of speech audio by the same speakers used to create the training set but of unheard utterances. The test set will be our ground-truth, which will compare that to the predicted classifications by the Conformer model to derive an accuracy ratio.

Challenges

As a new ML practitioner, taking on the challenge of building a Conformer model from scratch will be a challenge. Another challenge is that I am also new to audio and signal processing and will need to learn how to configure a model in this domain.

Milestones

Phase 1: Data collection - 10/13
Phase 2: Data preprocessing - 11/1
Phase 3: Model building - 11/15
Phase 4: Model training - 12/1
Phase 5: Analyze and build report on model outputs - 12/15
Phase 6: Present report - 12/20