University of Sheffield

Speaker Diarization System for the DIHARD Challenge

Mangesh Hambarde

Supervisor: Dr. Thomas Hain

A report submitted in fulfilment of the requirements
for the degree of MSc Computer Science with Speech and Language
Processing

in the

Department of Computer Science

August 31, 2019

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: ____

Signature: ______________________________________

Date: ____

Abstract

Speaker Diarization is the task of finding out “who spoken when?” given an audio recording. It is an important field because it is a crucial preprocessing step for many areas in speech processing like speech recognition. Over recent years, the availability of fast computing power and emergence of massive amounts of multimedia data has boosted the field. The demand for various kinds of speech technology, and hence diarization, has become greater than ever. The performance of diarization systems has also improved significantly in the past decade or so thanks to the continued efforts of the research community. But even so, the absolute numbers tell a different story and it seems that there is still a long way to go. There is still a need for big improvements so that speaker diarization technology can be deployed in real world applications. This makes it an exciting research area which has a lot of potential to improve in the next few years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Speaker diarization is the task of finding out “who spoke when?” in an audio recording with an unknown amount of speakers. It aims to find all segments of speech within the recording, possibly overlapping, along with their intra-recording speaker identities. It acts as an important upstream preprocessing step for most tasks in speech processing, like speech recognition, speech enhancement, speech coding etc.

With increase in computing power, speech processing technologies have achieved incredible advances in the past decade that were not possible earlier. This has increased interest in Rich Transcription (RT) technologies that can be used to automatically index the enormous amount of audio and video information that is generated in the modern world. Since speaker diarization is an important part in any RT system, there is a great deal of research interest in the area.

Diarization is not an easy problem since the output is affected by several factors like the application domain (broadcast news, meetings, telephone audio, internet audio, restaurant speech, clinical recordings etc), types and quality of microphones used (boom, lapel, far-field), inter-channel synchronization problems, overlapping speech, etc. These days, most of the research focuses on the meeting speech domain, since most problems that exist in speech recognition are encountered in this domain. The meeting scenario is thus often termed as “speech recognition complete”.

The DIHARD challenge was created to establish standard datasets for diarization and create performance baselines for comparison, thus encouraging further research. The challenge focuses on “hard” diarization, combining several domains of speech like broadcast speech, meeting speech, telephone speech, and many more. Creating a system for the challenge can be a rewarding experience since it gives a chance to learn about state-of-the-art speaker diarization techniques.

1.1 Speaker Diarization

1.2 Motivation and Objectives

1.3 Report Outline

Chapter 2

Literature Review on Speaker Diarization

2.1 Introduction

2.2 Feature Extraction

2.3 Speech Activity Detection

2.4 Segmentation

2.5 Clustering

2.5.1 Speaker Representation

GMM

i-vectors

x-vectors

2.5.2 Agglomerative Clustering

2.5.3 Distance metrics

BIC

PLDA

2.6 Evaluation

Diarization Error Rate

Jaccard Error Rate

2.7 Kaldi toolkit

Chapter 3

DIHARD challenge setup

3.1 Task definition

3.2 Evaluation Tracks

3.3 Scoring

3.4 Datasets

Chapter 4

Baseline setup

4.1 Overview

There are three software baselines provided by the DIHARD organizers, each for the parts of speech enhancement, speech activity detection and diarization. The speech enhancement baseline and the speech activity detection together are meant to be used in the case of system SAD (tracks 2 and 4), but since we only work with reference SAD, we do not need them. Thus we will only describe the diarization baseline in the following sections.

The diarization baseline is based on the best performing submission from John Hopkins University (JHU) in the previous year's DIHARD challenge (DIHARD I). There are 4 Kaldi recipes, each for an evaluation track, but we will focus only on the recipe for Track 1 since we only work with single channel and gold speech segmentation.

The baseline code modifies and reuses the `egs/dihard_2018` recipe that was checked into Kaldi by the researchers at JHU. It does this by copying over new scripts and data that is needed to the `egs/dihard_2018` directory and running the recipe from there.

This baseline code is in turn modified to make it easier to run with different trained models and parameters. All the diarization runs are run in this way.

4.2 Initial segmentation

The initial segmentation step deals with identifying speech and non-speech segments from the recording files using the reference SAD which is provided in the form of HTK label files. This results in a bunch of segments which are known to be containing only speech. These are called "utterances" in Kaldi terminology and act as keys in the `utt2spk`, `feats.scp` and `segments` files. These reside in a Kaldi "data directory", one for each dev and eval.

4.3 Features

The baseline then extracts 30 dimensional MFCC features for each of the every 10 ms using a 25 ms window. It uses the standard `steps/make_mfcc.sh` Kaldi script for this. Later, cepstral mean and variance normalization (CMVN) with a 3 second sliding window is applied using the `apply-cmvn-sliding` Kaldi tool.

4.4 Main segmentation

After feature extraction, the utterances are uniformly divided into smaller 1.5 second subsegments with a 0.75 second overlap. This creates another Kaldi data directory with newer keys corresponding to each subsegment. An x-vector is extracted from each of these subsegments in the next step.

4.5 Speaker representation for utterances

The baseline extracts an x-vector from each subsegment using an x-vector extractor that is trained on the datasets Voxceleb I and II, along with added augmentation. The augmentation is done by additive noise (music, babble) from the MUSAN dataset and reverberation from the RIR dataset. The extraction script is `egs/callhome_diarization/v1/diarizati`

The baseline uses x-vectors [1] to represent speakers for each utterance. The x-vector extractor is a neural network that has a 512-dimensional embedding layer from which the x-vectors are extracted. This neural network is trained using data. The neural network is trained to discriminitively predict the correct speaker, given a chunk of frames at a time from the training data.

4.6 Domain adaptation

To adapt the extracted x-vectors to the DIHARD domain, they are normalized with a global mean and whitening transform that is learned from the DIHARD development set.

4.7 Scoring

The x-vectors are scored using a PLDA backend that is trained on a subset of Voxceleb consisting of segments of at least 3 seconds duration. This is done using the existing script `egs/callhome_diarization/v1/diarization/score_plda.sh`. These scores are stored as affinity matrices which give the scores between any pair of x-vectors.

4.8 Clustering

The x-vectors are then clustered using agglomerative hierarchical clustering (AHC) and a parameter sweep is done on the dev set to find the threshold that maximises the DER on the dev set. This threshold is then used for clustering the x-vectors of the eval set. The script used for clustering is `egs/callhome_diarization/v1/diarization/cluster.sh`.

4.9 Diarization output

The clustering output is used to generate RTTMs using `egs/callhome_diarization/v1/diarization/m` The RTTMs give a flat segmentation of the recordings with no overlap. Since the x-vectors were extracted from segments that were overlapping, care needs to be taken when two adjacent segments are assigned to a different speaker. The script places the speaker boundary midway between the end of the first segment and the start of the second segment.

Chapter 5

Experiments and Results

5.1 Baseline results

5.2 Using Existing Pre-trained Models

5.2.1 Kaldi VoxCeleb x-vector model

5.2.2 Kaldi VoxCeleb i-vector model

5.3 Training Models on In-Domain Data

5.3.1 No Augmentation

5.3.2 Augmentation with Noise and Reverberation

5.4 Manual Speech Activity Detection

5.4.1 WebRTC

5.4.2 SHoUT toolkit

5.4.3 Energy-based

5.4.4 DNN-based

5.5 Speech Enhancement

5.5.1 Baseline

5.6 Alternative Distance Metrics for Clustering

5.6.1 BIC

5.6.2 Cosine

Chapter 6

Conclusions

Bibliography

[1] Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 5329–5333.