OPTIMIZING SIMILARITY THRESHOLD FOR ABSTRACT SIMILARITY METRIC IN SPEECH DIARIZATION SYSTEMS: A MATHEMATICAL FORMULATION

Jagat Chaitanya Prabhala, Dr. Venkatnareshbabu K and Dr. Ragoju Ravi

Department of Applied Sciences National Institute of Technology Goa, India

ABSTRACT

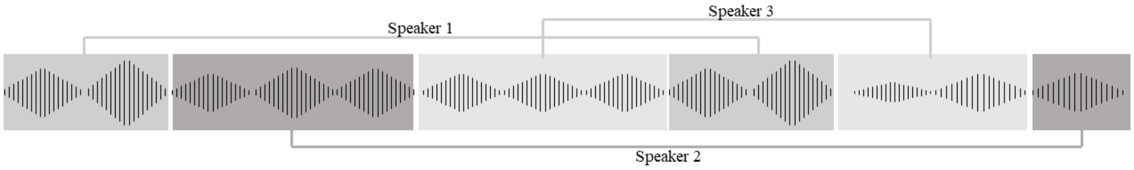
Speaker diarization is a critical task in speech processing that aims to identify "who spoke when?" in an audio or video recording that contains unknown amounts of speech from unknown speakers and unknown number of speakers. Diarization has numerous applications in speech recognition, speaker identification, and automatic captioning. Supervised and unsupervised algorithms are used to address speaker diarization problems, but providing exhaustive labeling for the training dataset can become costly in supervised learning, while accuracy can be compromised when using unsupervised approaches. This paper presents a novel approach to speaker diarization, which defines loosely labeled data and employs x-vector embedding and a formalized approach for threshold searching with a given abstract similarity metric to cluster temporal segments into unique user segments. The proposed algorithm uses concepts of graph theory, matrix algebra, and genetic algorithm to formulate and solve the optimization problem. Additionally, the algorithm is applied to English, Spanish, and Chinese audios, and the performance is evaluated using well-known similarity metrics. The results demonstrate that the robustness of the proposed approach. The findings of this research have significant implications for speech processing, speaker identification including those with tonal differences. The proposed method offers a practical and efficient solution for speaker diarization in real-world scenarios where there are labeling time and cost constraints.

KEYWORDS

Speaker diarization, speech processing, signal processing, x-vector, graph theory, similarity matrix, optimization

1. INTRODUCTION

Speech processing is a fundamental area of research in which the goal is to analyze and understand spoken language. It can be divided into two primary categories: speech recognition and speaker recognition. While speech recognition is concerned with the content of the speech signal, speaker recognition aims to identify the speaker(s) within a conversation. Speaker diarization is an important step in multi-speaker speech processing applications that falls under the latter category.



DOI : 10.5121/mathsj.2023.10201

In speaker diarization, the speech audio is segmented into temporal segments that contain the voice of a single speaker. These segments are further clustered into homogeneous groups to identify each speaker and the boundaries/frames of their speech. This process is becoming increasingly important as an automatic pre-processing step for building speech processing systems, with direct applications in speaker indexing, retrieval, speech recognition with speaker identification, and diarizing meetings and lectures.

This research paper proposes a novel approach to diarization that can handle scenarios where the number of speakers is unknown and their identities are not known beforehand. Specifically, we explore the use of loosely labeled speech data and formalize a mathematical model to search for the appropriate similarity threshold to diarize the speakers. Our approach utilizes pre-trained x-vector based voice verification embeddings, along with some simple concepts of graph theory for clustering and a genetic algorithm to identify the optimal similarity threshold for speaker diarization that minimizes the error rate. We evaluate the effectiveness of our approach on speech data in various languages, using different frame lengths and similarity metrics. The results demonstrate that our method can achieve high accuracy in speaker diarization even with unknown speakers and unknown number of speakers, which has significant implications for speech processing applications in a variety of fields.

This paper will begin by reviewing the existing literature on x-vectors and brief approach for diarization for labeled and unlabeled data followed by description of the proposed methodology in which we introduce some definitions and mathematical formulation. After describing the methodology we also illustrate the approach with a dummy example followed by results on real audio data and hence conclude the findings and applications in the closing section

2. RELATED WORK

Extensive research has been conducted on speaker diarization and clustering of speech segments using i-vector similarity or x-vector embeddings, along with various similarity metrics, such as cosine similarity [1]. Bayesian Information Criterion (BIC) is a widely used method for speaker diarization model selection [2][3], which can also estimate the number of speakers in a recording. Other Bayesian methods have also been proposed for this purpose [4][5]. Hierarchical Agglomerative Clustering (HAC) is a commonly employed unsupervised method for speaker cluster learning; however, the techniques for determining the optimal number of clusters are not sophisticated due to the problem being posed as an unsupervised learning problem.

Supervised learning-based clustering using Probabilistic Linear Discriminant Analysis (PLDA) of x-vectors and Bayesian Hidden Markov Model (BHMM)-based re-segmentation are other approaches that have been explored in this field [6]. However, these methods may not be suitable for loosely labeled data.

Our proposed methodology is better suited for training models with loosely labeled data. The existing methods are not optimal for our data as it is neither fully labeled nor completely unlabeled. Therefore, we propose a new approach that can effectively train models with loosely labeled data for speaker diarization and clustering of speech segments.

3. METHODOLOGY

Def: Training audio data is called ‘loosely labeled’ if each audio file has information about the number of speakers in the audio but do not have detailed annotation of who spoke when.

Given set of multi-speaker audio files $D = \{(A_k, t_k) \mid k = 1,2\ldots n\}$ where $A_k$is k^{th}multi-speaker audio and $t_k$is number of speakers in A_k. We define audio data where only the number of speakers is known but no information about temporal segmentation of speaker is given as loose labels. Below [ref: Image 1] is process diagram for training and prediction

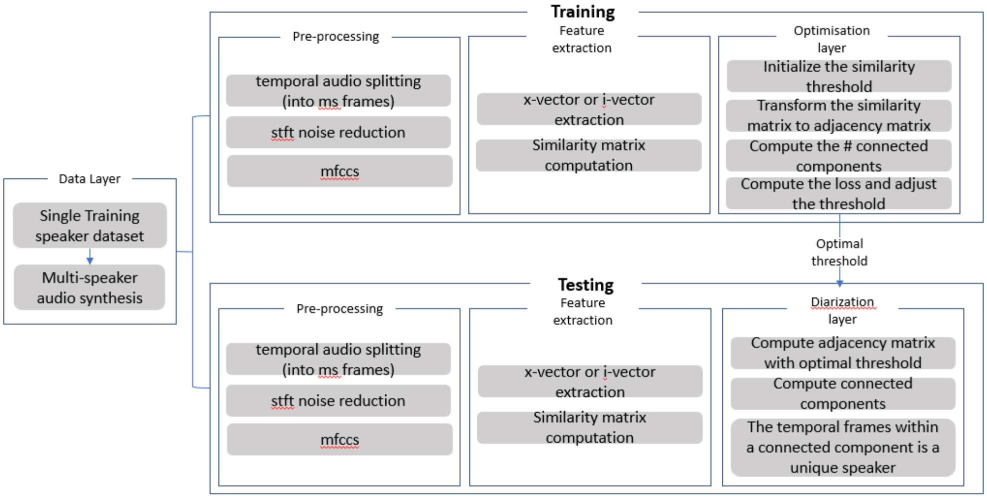

Training
Pre-processing
temporal audio splitting (into ms frames)
stft noise reduction
mfccs
Feature extraction
x-vector or i-vector extraction
Similarity matrix computation
Optimisation layer
Initialize the similarity threshold
Transform the similarity matrix to adjacency matrix
Compute the # connected components
Compute the loss and adjust the threshold
Data Layer
Single Training speaker dataset
Multi-speaker audio synthesis
Optimal threshold
Testing
Pre-processing
temporal audio splitting (into ms frames)
stft noise reduction
mfccs
Feature extraction
x-vector or i-vector extraction
Similarity matrix computation
Diarization layer
Compute adjacency matrix with optimal threshold
Compute connected components
The temporal frames within a connected component is a unique speaker

Image 1

3.1. Detailed Methodology

3.1.1. Data

As a first step we used single speaker audios from multiple languages i.e SLR38-Chinese, SLR61-Spanish, SLR45-English &SLR83-English datasets provided in openslr data [7] as input to generate synthetic multi-speaker audio generation i.e., each audio has more than one speaker.

The table below has more details about the synthesized dataset

Language	Total # of files	Average duration	Average # of speakers
English	6844	103s	3.2
Spanish	2106	53s	2.7
Chinese	5823	154s	4.5

3.1.2. Pre-Processing & Feature Extraction

Split the above generated audio files into small temporal segments with length of 15ms and use pretrained model in the kaldi project [8] to extract x-vector embeddings of the temporal segments. The block diagram [ref: Image2] of x-vector training architecture is as follows

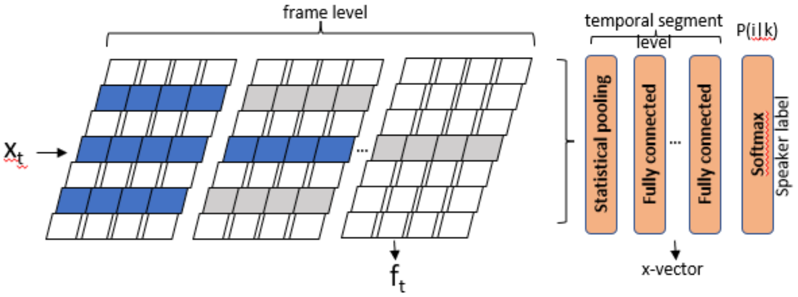


Image2

By end of the data processing step kth multi-speaker audio file A_k results in a stream of x-vectors x_{k1},x_{k2}, ...,x_{kn} where x_{ki} represents the x-vector of the ith temporal segment

3.1.3. Problem Formulation

Further we compute similarity matrix

$$S_k = S(A_k) = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nn} \end{bmatrix}$$ where $S_{ij} = s(x_{ki},x_{kj})$ for an abstract similarity function ‘s’

Define a function G_p: $\{S_k\} \rightarrow \{Adj_k\}$ from set of similarity matrix $\{S_k\}$ to set of adjacency matrix $\{Adj_k\}$as matrix G_p (S_k) = 1 if S_{ij}>=p and 0 otherwise. Clearly one can see that $G_p(S_k)$ is adjacency matrix with temporal segments as nodes

Define a function H:$\{Adj_k\} \rightarrow \mathbb{Z}^+$ as $H(Adj_k)$ = # of connected components [12]

The objective is to $\underset{p}{minimize}\sum_{k=1}^{l}\left(HoG_p(S_k)-(t_k+1)\right)^2$ represented as $\underset{p}{minimize}$ Q(p). We use t_k+1 to account for silence segments

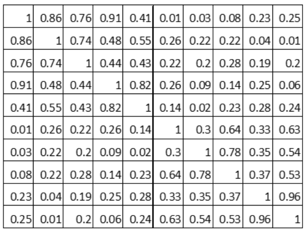
For the optimal ‘p’, $G_p(S_k)$ is an adjacency matrix corresponding to kth audio file A_k and each connected component of this adjacency matrix is a unique user resulting in diarization of the multi-speaker audio

Given $HoG_p(S_k)$ and t_k are positive integers Q(p) is a discrete quadratic function and at least one global optimal is guaranteed and due to property of quadratic curve Q(p) doesn’t have any local optimal. As Q(p) is neither continuous not differentiable we cannot use gradient descent or sub-gradient descent algorithms to find the optimal ‘p’. We have used genetic algorithm [11] to search the optimal p.

3.2. Illustration of Proposed method

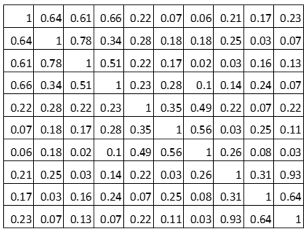
Let {(A_1,2),(A_2,3), (A_3, 2)} be the set of similarity matrix created from audio files and corresponding number of speakers with arbitrary similarity matrix. For ease of illustration assume the similarity metric is bounded between [-1,1]. Let matrices A_1, A_2, A_3 be as represented in the following tables [Image 3]

A_1=



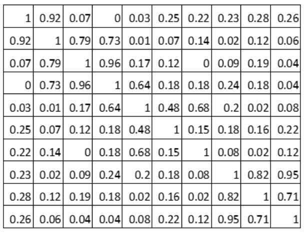
1	0.86	0.76	0.91	0.41	0.01	0.03	0.08	0.23	0.25
0.86	1	0.74	0.48	0.55	0.26	0.22	0.22	0.04	0.01
0.76	0.74	1	0.44	0.43	0.22	0.2	0.28	0.19	0.2
0.91	0.48	0.44	1	0.82	0.26	0.09	0.14	0.25	0.06
0.41	0.55	0.43	0.82	1	0.14	0.02	0.23	0.28	0.24
0.01	0.26	0.22	0.26	0.14	1	0.3	0.64	0.33	0.63
0.03	0.22	0.2	0.09	0.02	0.3	1	0.78	0.35	0.54
0.08	0.22	0.28	0.14	0.23	0.64	0.78	1	0.37	0.53
0.23	0.04	0.19	0.25	0.28	0.33	0.35	0.37	1	0.96
0.25	0.01	0.2	0.06	0.24	0.63	0.54	0.53	0.96	1

A_2=



1	0.64	0.61	0.66	0.22	0.07	0.06	0.21	0.17	0.23
0.64	1	0.78	0.34	0.28	0.18	0.18	0.25	0.03	0.07
0.61	0.78	1	0.51	0.22	0.17	0.02	0.03	0.16	0.13
0.66	0.34	0.51	1	0.23	0.28	0.1	0.14	0.24	0.07
0.22	0.28	0.22	0.23	1	0.35	0.49	0.22	0.07	0.22
0.07	0.18	0.17	0.28	0.35	1	0.56	0.03	0.25	0.11
0.06	0.18	0.02	0.1	0.49	0.56	1	0.26	0.08	0.03
0.21	0.25	0.03	0.14	0.22	0.03	0.26	1	0.31	0.93
0.17	0.03	0.16	0.24	0.07	0.25	0.08	0.31	1	0.64
0.23	0.07	0.13	0.07	0.22	0.11	0.03	0.93	0.64	1

A_3=

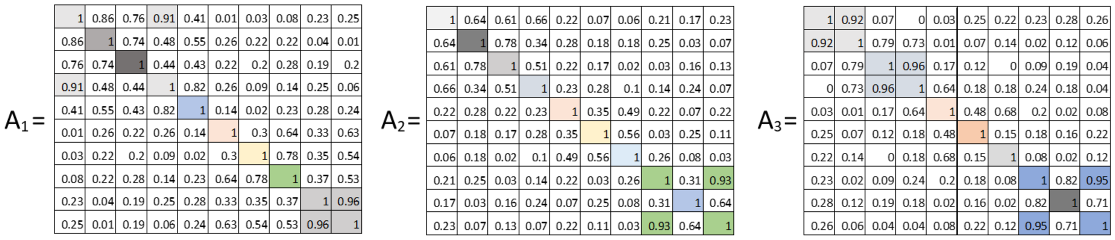


1	0.92	0.07	0	0.03	0.25	0.22	0.23	0.28	0.26
0.92	1	0.79	0.73	0.01	0.07	0.14	0.02	0.12	0.06
0.07	0.79	1	0.96	0.17	0.12	0	0.09	0.19	0.04
0	0.73	0.96	1	0.64	0.18	0.18	0.24	0.18	0.04
0.03	0.01	0.17	0.64	1	0.48	0.68	0.2	0.02	0.08
0.25	0.07	0.12	0.18	0.48	1	0.15	0.18	0.16	0.22
0.22	0.14	0	0.18	0.68	0.15	1	0.08	0.02	0.12
0.23	0.02	0.09	0.24	0.2	0.18	0.08	1	0.82	0.95
0.28	0.12	0.19	0.18	0.02	0.16	0.02	0.82	1	0.71
0.26	0.06	0.04	0.04	0.08	0.22	0.12	0.95	0.71	1

Image3

Let’s evaluate number of connected components and Q(p) for each threshold ‘p’ where p = 0.9,0.7,0.3for all the matrices in [ref: Image 3]

For p = 0.9, connected components of A_1, A_2, $A_3$are 8, 9, 7 respectively. The similar colored edges [ref: Image 4] for each matrix is a connected component



A_1=

1	0.86	0.76	0.91	0.41	0.01	0.03	0.08	0.23	0.25
0.86	1	0.74	0.48	0.55	0.26	0.22	0.22	0.04	0.01
0.76	0.74	1	0.44	0.43	0.22	0.2	0.28	0.19	0.2
0.91	0.48	0.44	1	0.82	0.26	0.09	0.14	0.25	0.06
0.41	0.55	0.43	0.82	1	0.14	0.02	0.23	0.28	0.24
0.01	0.26	0.22	0.26	0.14	1	0.3	0.64	0.33	0.63
0.03	0.22	0.2	0.09	0.02	0.3	1	0.78	0.35	0.54
0.08	0.22	0.28	0.14	0.23	0.64	0.78	1	0.37	0.53
0.23	0.04	0.19	0.25	0.28	0.33	0.35	0.37	1	0.96
0.25	0.01	0.19	0.06	0.24	0.63	0.54	0.53	0.96	1

A_2=

1	0.64	0.61	0.66	0.22	0.07	0.06	0.21	0.17	0.23
0.64	1	0.78	0.34	0.28	0.18	0.18	0.25	0.03	0.07
0.61	0.78	1	0.51	0.22	0.17	0.02	0.03	0.16	0.13
0.66	0.34	0.51	1	0.23	0.28	0.1	0.14	0.24	0.07
0.22	0.28	0.22	0.23	1	0.35	0.49	0.22	0.07	0.22
0.07	0.18	0.17	0.28	0.35	1	0.56	0.03	0.25	0.11
0.06	0.18	0.02	0.1	0.49	0.56	1	0.26	0.08	0.03
0.21	0.25	0.03	0.14	0.22	0.03	0.26	1	0.31	0.93
0.17	0.03	0.16	0.24	0.07	0.25	0.08	0.31	1	0.64
0.23	0.07	0.13	0.07	0.22	0.11	0.03	0.93	0.64	1

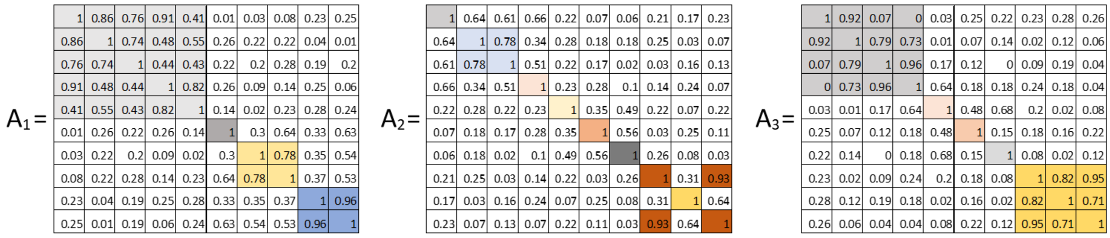
A_3=

1	0.92	0.07	0	0.03	0.25	0.22	0.23	0.28	0.26
0.92	1	0.79	0.73	0.01	0.07	0.14	0.02	0.12	0.06
0.07	0.79	1	0.96	0.17	0.12	0	0.09	0.19	0.04
0	0.73	0.96	1	0.64	0.18	0.18	0.24	0.18	0.04
0.03	0.01	0.17	0.64	1	0.48	0.68	0.2	0.02	0.08
0.25	0.07	0.12	0.18	0.48	1	0.15	0.18	0.16	0.22
0.22	0.14	0	0.18	0.68	0.15	1	0.08	0.02	0.12
0.23	0.02	0.09	0.24	0.2	0.18	0.08	1	0.82	0.95
0.28	0.12	0.19	0.18	0.02	0.16	0.02	0.82	1	0.71
0.26	0.06	0.04	0.04	0.08	0.22	0.12	0.95	0.71	1

Image4

Q(0.9) = $(8-3)^2 + (9-4)^2 + (7-3)^2$
Q(0.9) = 66

For p = 0.7, connected components of A_1, A_2, A_3 are 4, 8, 5 respectively. The similar colored edges [ref: Image 5] for each matrix is a connected component



A_1=

1	0.86	0.76	0.91	0.41	0.01	0.03	0.08	0.23	0.25
0.86	1	0.74	0.48	0.55	0.26	0.22	0.22	0.04	0.01
0.76	0.74	1	0.44	0.43	0.22	0.2	0.28	0.19	0.2
0.91	0.48	0.44	1	0.82	0.26	0.09	0.14	0.25	0.06
0.41	0.55	0.43	0.82	1	0.14	0.02	0.23	0.28	0.24
0.01	0.26	0.22	0.26	0.14	1	0.3	0.64	0.33	0.63
0.03	0.22	0.2	0.09	0.02	0.3	1	0.78	0.35	0.54
0.08	0.22	0.28	0.14	0.23	0.64	0.78	1	0.37	0.53
0.23	0.04	0.19	0.25	0.28	0.33	0.35	0.37	1	0.96
0.25	0.01	0.19	0.06	0.24	0.63	0.54	0.53	0.96	1

A_2=

1	0.64	0.61	0.66	0.22	0.07	0.06	0.21	0.17	0.23
0.64	1	0.78	0.34	0.28	0.18	0.18	0.25	0.03	0.07
0.61	0.78	1	0.51	0.22	0.17	0.02	0.03	0.16	0.13
0.66	0.34	0.51	1	0.23	0.28	0.1	0.14	0.24	0.07
0.22	0.28	0.22	0.23	1	0.35	0.49	0.22	0.07	0.22
0.07	0.18	0.17	0.28	0.35	1	0.56	0.03	0.25	0.11
0.06	0.18	0.02	0.1	0.49	0.56	1	0.26	0.08	0.03
0.21	0.25	0.03	0.14	0.22	0.03	0.26	1	0.31	0.93
0.17	0.03	0.16	0.24	0.07	0.25	0.08	0.31	1	0.64
0.23	0.07	0.13	0.07	0.22	0.11	0.03	0.93	0.64	1

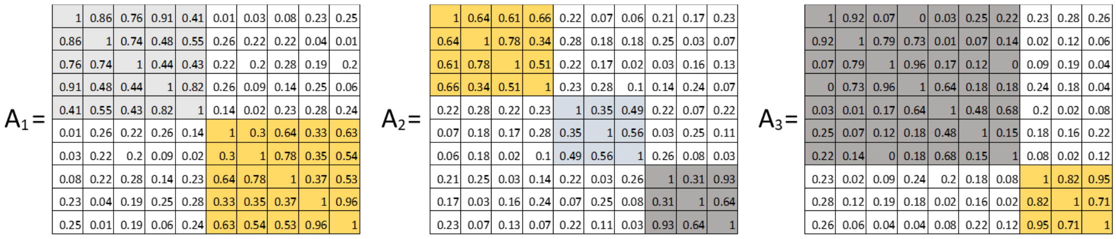
A_3=

1	0.92	0.07	0	0.03	0.25	0.22	0.23	0.28	0.26
0.92	1	0.79	0.73	0.01	0.07	0.14	0.02	0.12	0.06
0.07	0.79	1	0.96	0.17	0.12	0	0.09	0.19	0.04
0	0.73	0.96	1	0.64	0.18	0.18	0.24	0.18	0.04
0.03	0.01	0.17	0.64	1	0.48	0.68	0.2	0.02	0.08
0.25	0.07	0.12	0.18	0.48	1	0.15	0.18	0.16	0.22
0.22	0.14	0	0.18	0.68	0.15	1	0.08	0.02	0.12
0.23	0.02	0.09	0.24	0.2	0.18	0.08	1	0.82	0.95
0.28	0.12	0.19	0.18	0.02	0.16	0.02	0.82	1	0.71
0.26	0.06	0.04	0.04	0.08	0.22	0.12	0.95	0.71	1

Image5

Q(0.7) = $(4-3)^2 + (8-4)^2 + (5-3)^2$
Q(0.7) = 21

For p = 0.3, connected components of A_1, A_2, $A_3$are 2, 3, 2 respectively. The similar colored edges ref: [Image 6] for each matrix is a connected component



A_1=

1	0.86	0.76	0.91	0.41	0.01	0.03	0.08	0.23	0.25
0.86	1	0.74	0.48	0.55	0.26	0.22	0.22	0.04	0.01
0.76	0.74	1	0.44	0.43	0.22	0.2	0.28	0.19	0.2
0.91	0.48	0.44	1	0.82	0.26	0.09	0.14	0.25	0.06
0.41	0.55	0.43	0.82	1	0.14	0.02	0.23	0.28	0.24
0.01	0.26	0.22	0.26	0.14	1	0.3	0.64	0.33	0.63
0.03	0.22	0.2	0.09	0.02	0.3	1	0.78	0.35	0.54
0.08	0.22	0.28	0.14	0.23	0.64	0.78	1	0.37	0.53
0.23	0.04	0.19	0.25	0.28	0.33	0.35	0.37	1	0.96
0.25	0.01	0.19	0.06	0.24	0.63	0.54	0.53	0.96	1

A_2=

1	0.64	0.61	0.66	0.22	0.07	0.06	0.21	0.17	0.23
0.64	1	0.78	0.34	0.28	0.18	0.18	0.25	0.03	0.07
0.61	0.78	1	0.51	0.22	0.17	0.02	0.03	0.16	0.13
0.66	0.34	0.51	1	0.23	0.28	0.1	0.14	0.24	0.07
0.22	0.28	0.22	0.23	1	0.35	0.49	0.22	0.07	0.22
0.07	0.18	0.17	0.28	0.35	1	0.56	0.03	0.25	0.11
0.06	0.18	0.02	0.1	0.49	0.56	1	0.26	0.08	0.03
0.21	0.25	0.03	0.14	0.22	0.03	0.26	1	0.31	0.93
0.17	0.03	0.16	0.24	0.07	0.25	0.08	0.31	1	0.64
0.23	0.07	0.13	0.07	0.22	0.11	0.03	0.93	0.64	1

A_3=

1	0.92	0.07	0	0.03	0.25	0.22	0.23	0.28	0.26
0.92	1	0.79	0.73	0.01	0.07	0.14	0.02	0.12	0.06
0.07	0.79	1	0.96	0.17	0.12	0	0.09	0.19	0.04
0	0.73	0.96	1	0.64	0.18	0.18	0.24	0.18	0.04
0.03	0.01	0.17	0.64	1	0.48	0.68	0.2	0.02	0.08
0.25	0.07	0.12	0.18	0.48	1	0.15	0.18	0.16	0.22
0.22	0.14	0	0.18	0.68	0.15	1	0.08	0.02	0.12
0.23	0.02	0.09	0.24	0.2	0.18	0.08	1	0.82	0.95
0.28	0.12	0.19	0.18	0.02	0.16	0.02	0.82	1	0.71
0.26	0.06	0.04	0.04	0.08	0.22	0.12	0.95	0.71	1

Image6

Q(0.3) = $(2-3)^2 + (3-4)^2 + (2-3)^2$
Q(0.3) = 3

Clearly for threshold p = 0.2 connected components are 1,1,1 respectively for A_1, A_2, A_3 and hence
Q(0.2) = $(1-3)^2 + (1-4)^2 + (1-3)^2$
Q(0.2) = 17

For the illustrated example in [3.2] above, p = 0.3 is the best threshold among the threshold enumerated as Q(0.3)minimum.

In real life use case when data is large, we use genetic algorithm [11] to find the optimal value of ‘p’ instead of enumeration and exhaustive search

4. RESULTS & DISCUSSION

4.1. Evaluation Metric

As a standard we use mean DER(diarization error rate) as evaluation metric
DER(A) = $E_{spkr} + E_{miss} + E_{fa} + E_{ovl}$
where,
E_{spkr} is percentage of scored times the speaker ID is assigned to wrong speaker
E_{miss} is percentage of scored times the non-speech segment is categorized as speaker segment
E_{fa} is percentage of scored times a speaker segment is categorized as non-speech segment
E_{ovl} is percentage of times there is overlap of speakers in the segment

Mean DER = $\frac{1}{n}\sum_{k=1}^{n} DER(A_k)$

4.2. Similarity Metrics

The following similarity metrics are considered in this paper to test the proposed approach

4.2.1. Cosine Similarity

Cosine between two vectors X and Y is defined as $Cos(X,Y) = X.Y/|X||Y|$

4.2.2. Euclidian distance (L_2 norm)

The L_2 between two vectors X and Y is defined as $||X,Y||_{L2} = \sqrt{(X-Y)^2}$

4.2.3. Manhattan distance (L_1 norm)

The L_1between two vectors X and Y is defined as $||X,Y||_{L1} = abs(X-Y)$

4.2.4. Kendall's tau similarity

The kendall’s tau between two vectors X and Y is defined as$\tau(X,Y) = \frac{n_C - n_d}{n(n-1)/2}$ where n_c is # of concordant pairs and n_d is # of discordant pairs and n is length of vector

Note that Cosine and kendall’s tau similarity are bounded metrics and we can interpret segments are more similar when the value is higher, L_1& L_2 distance metrics are not bounded above and segments are closer if the value is closer to 0

4.3.Experimental Results

Test results when training & test is 90% and 10% respectively [ref: Image 7]

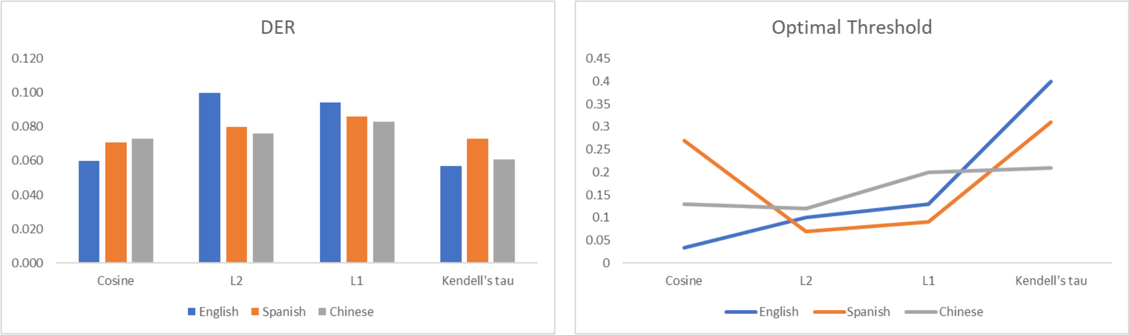


Image7

Test results when training & test is 80% and 20% respectively [ref: Image 8]

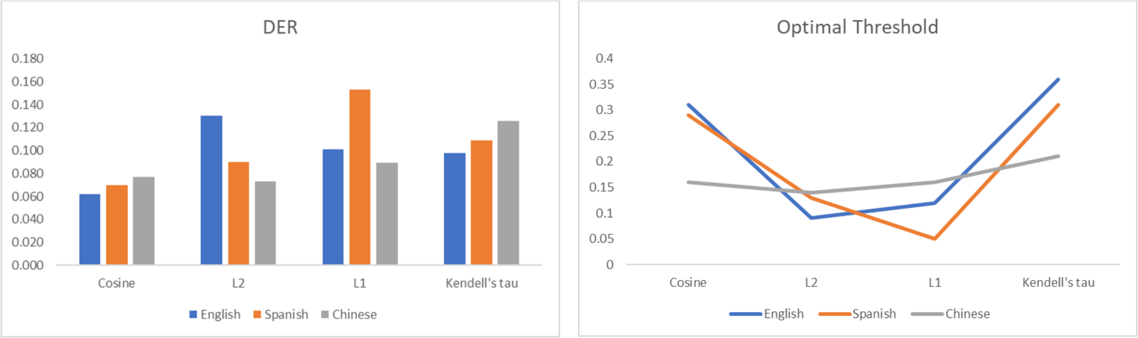


Image8

Test results when training & test is 70% and 30% respectively [ref: Image 9]

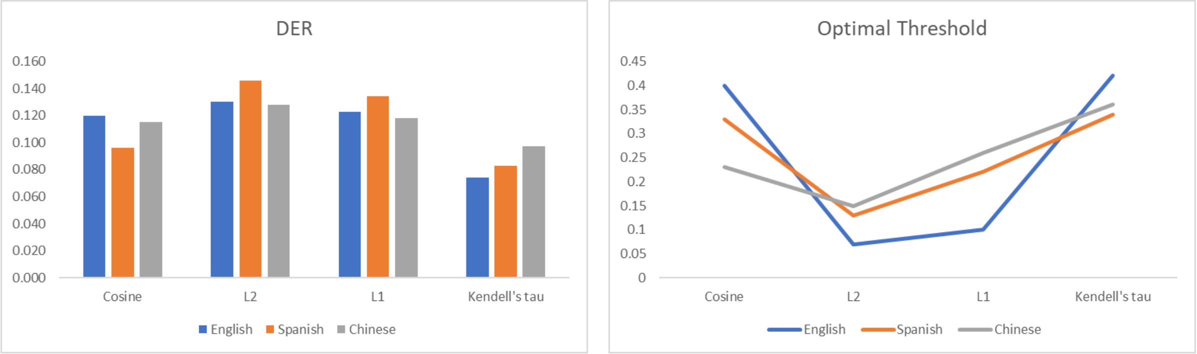


Image9

Test results when training & test is 60% and 40% respectively [ref: Image 10]

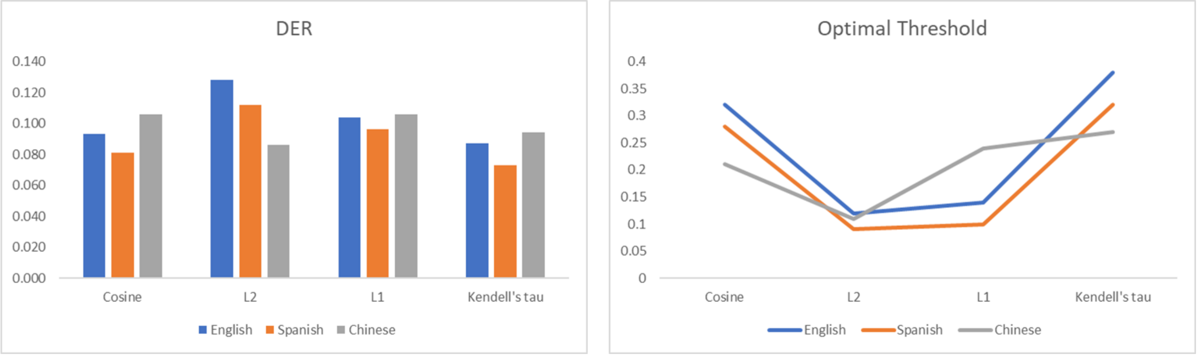


Image10

Test results when training & test is 50% and 50% respectively [ref: Image 11]

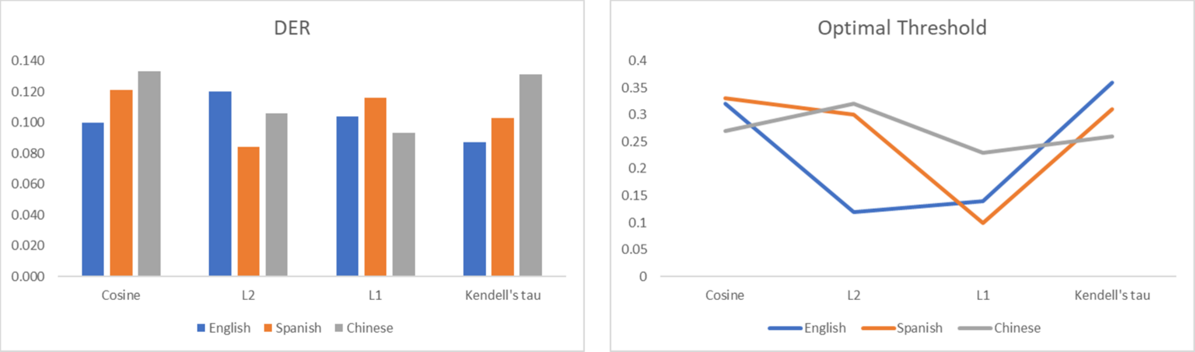
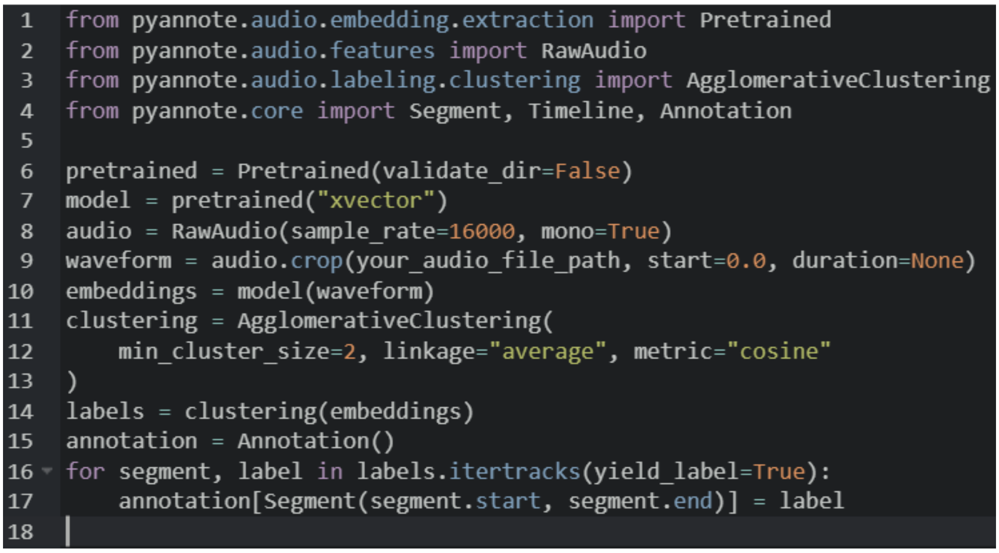


Image11

4.4. Code & Results with pretrained model [9] without threshold optimization

We have also used same test data and ran diarization algorithm [ref: code snippet 1] which uses pyannote library [9]. Please find the results below



```
from pyannote.audio.embedding.extraction import Pretrained
from pyannote.audio.features import RawAudio
from pyannote.audio.labeling.clustering import AgglomerativeClustering
from pyannote.core import Segment, Timeline, Annotation

pretrained = Pretrained(validate_dir=False)
model = pretrained("xvector")
audio = RawAudio(sample_rate=16000, mono=True)
waveform = audio.crop(your_audio_file_path, start=0.0, duration=None)
embeddings = model(waveform)
clustering = AgglomerativeClustering(
    min_cluster_size=2, linkage="average", metric="cosine"
)
labels = clustering(embeddings)
annotation = Annotation()
for segment, label in labels.itertracks(yield_label=True):
    annotation[Segment(segment.start, segment.end)] = label

```

Sample split	English DER	Spanish DER	Chinese DER
90%-10%	8.4%	9.6%	9.3%
80%-20%	7.8%	7.2%	6.4%
70%-30%	9.6%	10.4%	11.2%
60%-40%	10.2%	11.9%	9.1%
50%-50%	14.8%	12.7%	13.8%

5. CONCLUSION

In conclusion, this paper has presented a generic approach for identifying speakers and segmenting speech into homogenous speaker segments using loosely labeled data. We have found that similarity metrics such as cosine and Kendall's tau perform better than distance metrics such as L1 and L2 norms, with the cosine similarity metric performing the best across languages with an 80-20 split. Our threshold optimized model has reduced the error by at least 2 percentage points over the existing state-of-the-art approach [4.4]. This methodology is a generic formulation that can be used with custom metrics or combined with kernel techniques such as Gaussian or radial basis kernels.

Although supervised techniques give the least DER, they are costly in terms of money and time for detailed annotation. Our approach is best suited when the aim is to be better than unsupervised approaches and minimize the cost of learning. The current approach took an average of 3 hours of training time for each language and metric combination, with Kendall's tau taking the longest time of 5-6 hours. As next steps, we plan to explore other optimization techniques to further reduce training time

ACKNOWLEDGEMENTS

I am grateful to my mother Subbalakshmi Devulapally for inspiring me to do research. Also, a special thanks to my wife Bhavani Paruchuri for continually supporting me and standing by me through this journey.

REFERENCES

[1] M. Senoussaoui, P. Kenny, T. Stafylakis and P. Dumouchel, (2014), "A Study of the Cosine Distance-Based Mean Shift for Telephone Speech Diarization," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 1, pp. 217-227

[2] G. Schwarz, (1978), “Estimating the dimension of a model,” Ann. Statist. 6, 461-464

[3] S. S. Chen and P. Gopalakrishnan, (1998), “Clustering via the bayesian information criterion with applications in speech recognition,” in ICASSP’98, vol. 2, Seattle, USA, pp. 645–648

[4] F. Valente,(2005), “VariationalBayesianmethods foraudio indexing,” Ph.D. dissertation, Eurecom

[5] P. Kenny, D. Reynolds and F. Castaldo,(2010), “Diarization of Telephone Conversations using Factor Analysis,” Selected Topics in Signal Processing, IEEE Journal of, vol.4, no.6, pp.1059-1070

[6] F. Landini et al.,(2020), "But System for the Second Dihard Speech Diarization Challenge," ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6529-6533

[7] https://www.openslr.org/resources.php

[8] http://kaldi-asr.org/models/m7

[9] Khoma, Volodymyr, et al., (2023), "Development of Supervised Speaker Diarization System Based on the PyAnnote Audio ProcessingLibrary

[10] Canovas, O., & Garcia, F. J., (2023). Analysis of Classroom Interaction Using Speaker Diarization and Discourse Features from Audio Recordings. In Learning in the Age of Digital and Green Transition: Proceedings of the 25th International Conference on Interactive Collaborative Learning (ICL2022), Volume 2 (pp. 67-74). Cham: Springer International Publishing

[11] Li, T., Shao, G., Zuo, W., & Huang, S. (2017). Genetic algorithm for building optimization: State-of-the-art survey. In Proceedings of the 9th international conference on machine learning and computing (pp. 205-210)

[12] Hubert, L. J. (1974). Some applications of graph theory to clustering. Psychometrika, 39(3), 283-309.

AUTHORS

Mr. Jagat Chaitanya Prabhala, is doing Ph.D from applied sciences department of National Institute Of Technology, Goa, India. He also has 15 years industry experience applying data science and AI for real business usecases. Currently he is working as data science leader at Sixt Research & Development Pvt Ltd.

Dr. Venkatnareshbabu K is working as Assistant Professor at computer science department of National Institute Of Technology, Goa, India.He has published multiple papers in very reputed journals in the area of computer vision and AI. He is a co-guide of Mr. Jagat Chaitanya

Dr. Ragoju Ravi is working as Associate Professor at applied sciences department of National Institute Of Technology, Goa, India. He has published multiple papers in reputed journals in thearea of applied mathematics including flued mechanics, heat and mass transfer etc.

Dr. Ravi is guide of Mr. Jagat Chaitanya