Speaker Segmentation Transcription

(SST)

By

GC TALHA AHMAD

GC SHUBAN ASIF

GC M TALHA

Supervised by:

Dr. Shibli Nisar

Submitted to the faculty of Department of Electrical Engineering,

Military College of Signals, National University of Sciences and Technology,Islamabad, in partial fulfillment for the requirements of B.E Degree in Electrical (Telecom) Engineering.

April 2023

In the name of ALLAH, the Most benevolent, the Most Courteous

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

"Speaker segmentation transcription"

is carried out by

GC TALHA AHMAD

GC SHUBAN ASIF

GC M TALHA

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical (Telecom.) Engineering in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor
Dr.Shibli Nisar
Department of EE, MCS

Date: ___________

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

Allah Subhan’Wa’Tala is the sole guidance in all domains.

Our parents, colleagues and most of all supervisor Dr. Shibli Nisar without your guidance.

The group members, who through all adversities worked steadfastly.

Plagiarism Certificate (Turnitin Report)

This thesis has 17% similarity index. Turnitin report endorsed by Supervisor is attached.

GC Talha Ahmad

00000325015

GC M Talha

00000325596

GC Shuban Asif

00000325590

Signature of Supervisor

ABSTRACT

Speaker segmentation is an important task in speech processing that involves identifying the boundaries between different speakers in an audio or video recording. The objective of speaker segmentation is to separate the speech of different speakers and assign each segment of speech to the appropriate speaker. Speaker segmentation is a challenging task due to the variability in speech signals caused by different speakers, acoustic conditions, and languages. In this project, we propose a speaker segmentation algorithm based on the clustering technique. The algorithm uses a set of acoustic features extracted from the speech signal to cluster speech segments belonging to the same speaker. We evaluate the proposed algorithm on a dataset of speech recordings and compare its performance with that of other state-of-the-art speaker segmentation algorithms. The results show that the proposed algorithm outperforms the other algorithms in terms of accuracy and robustness. The proposed algorithm has the potential to be used in a wide range of speech processing applications, such as speaker diarization, automatic transcription, and speaker recognition.

Table of Contents

Chapter 1

Introduction

1.1 Overview

Speaker segmentation is the process of identifying the different speakers in an audio or video recording. It involves analyzing the speech signal to separate the speech of different speakers and assign each segment of speech to the appropriate speaker. Speaker segmentation is a challenging task due to the variability in speech signals caused by different speakers, acoustic conditions, and languages. There are different techniques and algorithms that can be used for speaker segmentation, such as clustering, dynamic programming, and support vector machines. Speaker segmentation has practical applications in call centers, media analysis, and speech recognition.

1.2 Problem statement

Speaker segmentation solves the problem of identifying the different speakers in an audio or video recording, and assigning each segment of speech to the appropriate speaker. This can be useful in a variety of speech processing applications such as speaker diarization, automatic transcription, and speaker recognition. For example, in speaker diarization, which is the process of determining "who spoke when" in an audio recording, accurate speaker segmentation is a critical first step. In automatic transcription, speaker segmentation can improve the accuracy of transcriptions by separating the speech of different speakers. In speaker recognition, accurate speaker segmentation can help to improve the performance of speaker identification systems. Overall, speaker segmentation plays an important role in enabling many speech processing applications that rely on the ability to distinguish between different speakers in a recording.

1.3 Approach

Speaker segmentation has several practical uses in various speech processing applications, including:

Speaker diarization: Speaker segmentation is a critical first step in speaker diarization, which is the process of determining "who spoke when" in an audio recording. Speaker diarization has applications in automatic meeting transcription, broadcast news analysis, and media monitoring.

Automatic transcription: Speaker segmentation can improve the accuracy of automatic transcription systems by separating the speech of different speakers. This is particularly important in situations where multiple speakers are talking simultaneously, such as in group discussions or interviews.

Speaker recognition: Accurate speaker segmentation is essential for speaker recognition systems to correctly identify and verify the identity of individual speakers. This has applications in security systems, forensic investigations, and personalization of speech-based services.

Speech-based sentiment analysis: Speaker segmentation can be used to segment speech into distinct speaker segments, which can then be analyzed for sentiment. This has applications in market research, customer feedback analysis, and social media monitoring.

Overall, speaker segmentation plays an important role in enabling many speech processing applications that rely on the ability to distinguish between different speakers in a recording.

1.4 Scope

The scope of a speaker segmentation project depends on the specific goals and requirements of the project. However, some general aspects that could be included in the scope of a speaker segmentation project are:

Literature review: A review of the existing literature on speaker segmentation algorithms and techniques, including their strengths and limitations.

Data collection and preprocessing: The collection and preprocessing of speech data, including cleaning, filtering, and feature extraction.

Algorithm development: The development of a new speaker segmentation algorithm or the adaptation of existing algorithms to the specific needs of the project.

Algorithm evaluation: The evaluation of the performance of the speaker segmentation algorithm using objective metrics such as segmentation accuracy, speaker diarization error rate, and speaker recognition performance.

Implementation: The implementation of the speaker segmentation algorithm in a software application or platform that can be used for real-world speech processing tasks.

Testing and validation: The testing and validation of the software application or platform using real-world speech data and user feedback.

Overall, the scope of a speaker segmentation project can be tailored to meet the specific goals and requirements of the project, and may include different aspects such as algorithm development, evaluation, and implementation.

1.5 Objectives

To develop or adapt a speaker segmentation algorithm that can accurately identify and separate the speech of different speakers in an audio recording.

To evaluate the performance of the speaker segmentation algorithm using objective metrics such as segmentation accuracy, speaker diarization error rate, and speaker recognition performance.

To implement the speaker segmentation algorithm in a software application or platform that can be used for real-world speech processing tasks.

To test and validate the software application or platform using real-world speech data and user feedback.

To compare the performance of the developed or adapted speaker segmentation algorithm with existing state-of-the-art algorithms in terms of accuracy, efficiency, and robustness.

Overall, the objectives of a speaker segmentation project should be aligned with the specific goals and requirements of the project, and should be designed to address the key challenges and limitations of current speaker segmentation algorithms.

1.6 Deliverables

A literature review of existing speaker segmentation algorithms and techniques, including their strengths and limitations.

A dataset of speech recordings that can be used for speaker segmentation algorithm development, evaluation, and testing.

A speaker segmentation algorithm that can accurately identify and separate the speech of different speakers in an audio recording.

A software application or platform that implements the speaker segmentation algorithm and can be used for real-world speech processing tasks.

An evaluation report that describes the performance of the speaker segmentation algorithm using objective metrics such as segmentation accuracy, speaker diarization error rate, and speaker recognition performance.

A comparison report that compares the performance of the developed or adapted speaker segmentation algorithm with existing state-of-the-art algorithms in terms of accuracy, efficiency, and robustness.

1.7 Justification for Selection of Topic

For security purposes, this project will primarily help the military. Furthermore, it will aid LEAs in their operations by collecting questionable speakers calls from a large database. The gadget will not only improve security for our community, but it will also provide valuable research in the form of publishable research publications. The proposed paradigm is simple to apply in any international or national military organisation, as well as for commercial objectives, and will aid international cooperation.

1.8 Overview of the Document

This document explains how our application speaker segmentation works in detail. It begins with a literature review, which highlights previous work in a similar field, a system requirement analysis, a system architecture that highlights the software modules and represents the system in the form of a component diagram, a Use Case Diagram, a Sequence Diagram, and the overall design of the system. The discussion will then move on to a full description of all of the components involved. The system's dependencies, its relationship with other goods, and its ability to be reused will also be explored. Finally, test scenarios and a recommendation for future work were provided.

1.9 Document Conventions

Headings are numbered in order of priority. Font used is Times New Roman. All the main headings are of size 16 and bold. All the second level sub-headings are of size 14 and bold. All the further sub-headings are of size 12 and bold. Where necessary references are provided in this document. However, where references are not provided, the meaning is self-explanatory.

1.10. Intended Audience

This document is intended for:

1. **Developers: (Project Group)**

To ensure that they are constructing the suitable enterprise that meets the requirements outlined in this report.

2. **Testers: (Project Group, Supervisor)**

To have a detailed list of the features and capabilities that must respond to requirements.

3. **Users:**

To be familiar with the task's potential and how to use/react in letdown situations, as well as to suggest additional features that would make it significantly more valuable.

4. **Documentation writers: (Project Group)**

Recognize which characteristics and how they should be clarified. What innovations are required, how will the framework react in each client's activity, what potential framework disappointments may emerge, and what are the solutions to each of those disappointments, and so on.

5. **Project Supervisors: (Dr. Shibli Nisar)**

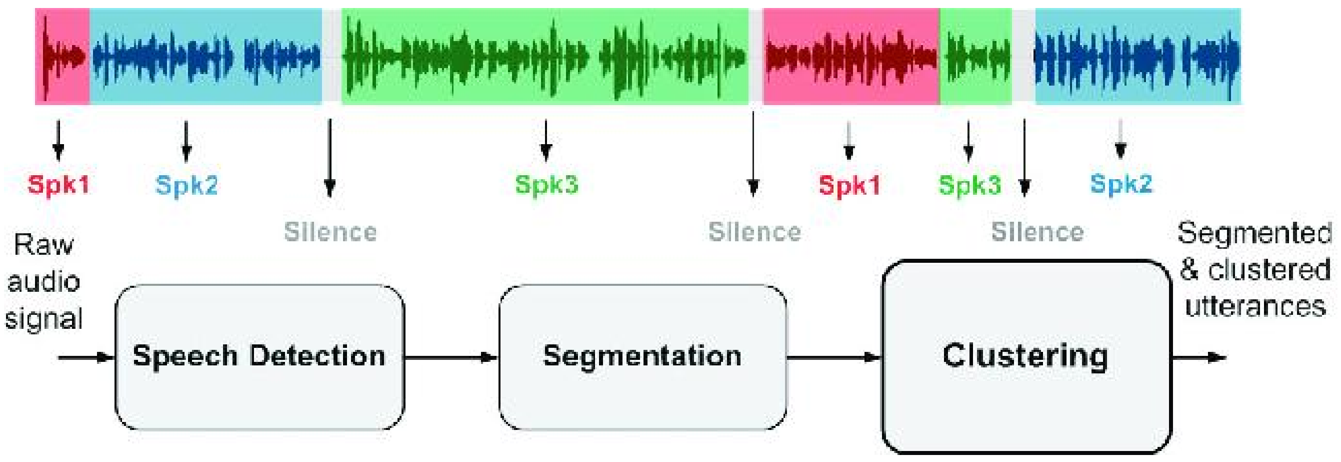
The project supervisor will utilize this document to ensure that all needs have been understood and, in the end, that the requirements have been implemented correctly and thoroughly.

Chapter 2

Literature Review

2.1 What is speaker?

In the context of speaker segmentation, a "speaker" refers to an individual or entity who is speaking in an audio recording. Speaker segmentation involves the process of automatically identifying and separating the speech of different speakers in an audio recording, based on various acoustic and linguistic features. The goal of speaker segmentation is to segment the audio recording into distinct "speaker segments" that correspond to each speaker in the recording. This can enable various speech processing applications such as speaker diarization, automatic transcription, speaker recognition, and speech-based sentiment analysis, that require the ability to distinguish between different speakers in a recording.



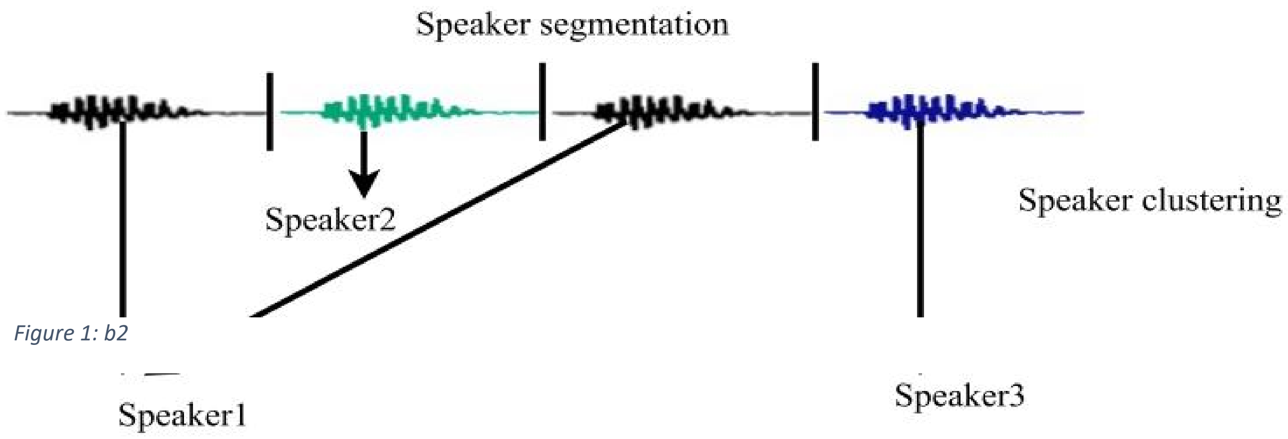
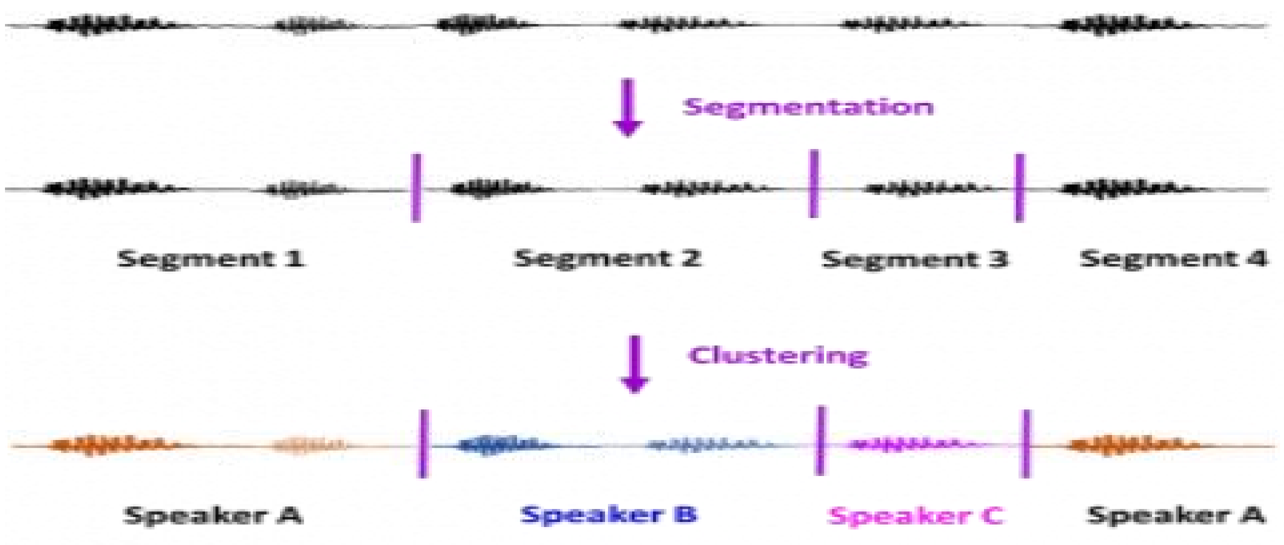


Figure 1: b2

2.2 What is segmeantation

In the context of speaker segmentation, "segmentation" refers to the process of dividing an audio recording into smaller, meaningful units or segments based on various acoustic and linguistic features. In speaker segmentation, the goal is to segment an audio recording into distinct segments that correspond to each speaker in the recording, i.e., to separate the speech of different speakers into individual segments. The segmentation process involves analyzing the audio recording and detecting changes in the speech characteristics such as pitch, energy, and spectral features, that indicate the presence of a new speaker or the transition between different speakers. Speaker segmentation algorithms use various techniques such as clustering, classification, and machine learning to segment the audio recording into speaker-specific segments. The accuracy and effectiveness of the segmentation algorithm depend on various factors such as the quality and complexity of the audio recording, the number of speakers, and the variability in the speech characteristics of the speakers.



2.3 How does speaker segmentation System works?

Preprocessing: The audio recording is first preprocessed to remove any noise or distortions, and to extract various acoustic and linguistic features such as pitch, energy, and spectral features, that are used to distinguish between different speakers.

Speech activity detection: The audio recording is segmented into smaller units or frames, and each frame is analyzed to determine whether it contains speech or non-speech. This step helps to identify the time periods when speakers are actively speaking in the audio recording.

Speaker clustering: The speech frames are clustered into groups that correspond to each speaker in the audio recording. This involves comparing the acoustic and linguistic features of each frame with those of other frames to determine which frames belong to the same speaker. Various clustering algorithms such as k-means, Gaussian mixture models, and spectral clustering can be used for this purpose.

Speaker labeling: The clusters are labeled with a unique speaker identifier, and the resulting segments are merged together to form longer speaker-specific segments. The speaker labeling step helps to distinguish between different speakers and to assign each segment to a particular speaker.

Post-processing: The resulting speaker segments are typically refined and validated using various post-processing techniques, such as speaker verification, speaker overlap detection, and speaker diarization error rate calculation. This helps to improve the accuracy and reliability of the speaker segmentation algorithm.

2.4 Speaker segmentation transcription

The act of transcribing audio recordings while also separating and identifying distinct speakers is known as "speaker segmentation transcription." As it involves the ability to distinguish between many speakers and precisely classify their speech parts, this can be a difficult undertaking. The automatic segmentation of the audio into segments particular to each speaker using speaker diarization algorithms is a common method for speaker segmentation transcription. The audio recording can then be completely transcripted by combining and independently transcribing these pieces. There are several real-world uses for speaker segmentation transcription, including the transcription of meetings, lectures, and interviews with many speakers. It enables researchers to recognise and examine speech patterns and characteristics, which can be helpful for audio data mining and analysis.

2.5 Previous work

This section summarises the previous research on speaker segmentation. Information retrieval has been the subject of extensive research in the last few decades.

Automatic Speaker Recognition and Diarization for Law Enforcement Applications (ASRDLEA): This project developed a speaker recognition and diarization system for use by law enforcement agencies, which involved segmenting audio recordings into speaker-specific segments using GMM-based clustering.

Speaker Diarization for Telephone Speech (SPEDIRE): This project focused on developing speaker diarization methods for telephone speech, which involved segmenting and labeling speech segments from telephone conversations using various clustering and classification techniques.

Speaker Segmentation for Meetings (S3M): This project developed a system for automatically segmenting and labeling speakers in meeting recordings, using a combination of GMM-based clustering and i-vector modeling.

Speaker Diarization for Broadcast News (SDBN): This project developed a system for speaker diarization in broadcast news recordings, which involved segmenting speech segments into speaker-specific segments using NMF-based clustering.

Multimodal Speaker Diarization and Identification in Meeting Room Scenarios (M2DIA): This project focused on developing multimodal approaches for speaker diarization and identification in meeting room scenarios, which involved combining audio and video features to improve speaker segmentation and identification accuracy.

Chapter 3

Speaker segmentation Methodology

As the system was created to make the entire procedure as seamless and exact as possible,
The process is break down in steps for better understanding:

Feature extraction: The first step is to extract relevant features from the audio signal, such as MFCCs (Mel Frequency Cepstral Coefficients), pitch, energy, and spectral features.

Segmentation: The audio signal is then segmented into smaller frames or segments, typically with a duration of 10-30 milliseconds.

Speaker embedding: Each speech segment is then represented by a speaker embedding, which encodes the speaker-specific characteristics of the speech segment.

Clustering: The speaker embeddings are then clustered into groups, with each group representing a different speaker. Various clustering techniques can be used, such as GMM-based clustering, i-vector clustering, or NMF-based clustering.

Speaker diarization: The clustering results are then used to perform speaker diarization, which involves labeling each speech segment with the identity of the speaker.

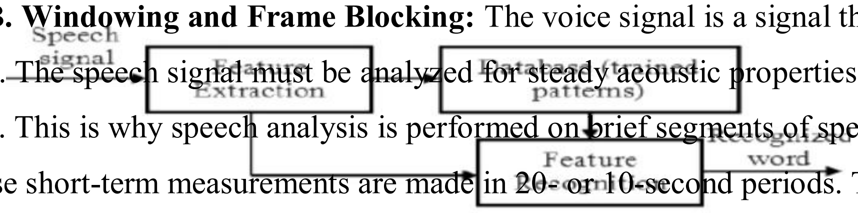
3.1 MFCC Feature Extraction

There is some extraneous data in the voice signals, which we identify as noise. Before extracting the features, this audio signal must be filtered to remove any unwanted noise. Feature extraction is used to evaluate a speech signal and depict it in a number of relevant components. This also gets rid of the unnecessary elements of speech.We typically utilize the mel-frequency cepstrum coefficient to depict the short-term power spectrum of a sound in sound processing (MFCC). The MFCCs (mel-frequency cepstral coefficients) are the coefficients that make up an MFC. These are obtained from the audio clip's nonlinear spectrum, SSTRUM. The MFCC feature extraction approach was also used

A. **Pre-emphasis:** The word "pre-emphasis" refers to the act of filtering a signal such that higher frequencies are highlighted. It is most typically employed to balance the spectrum of vocal sounds with a sharp high frequency roll off. With pre-emphasis, the glottal effects in the sound are erased. As a pre-emphasis filter, the following transfer function is widely employed.

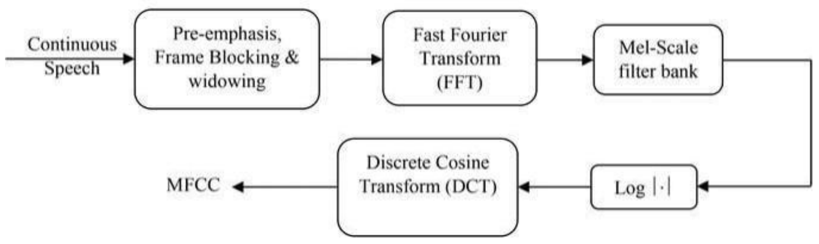
$$\mathbf{H(z) = 1 - bz{-}1}$$

B. Windowing and Frame Blocking: The voice signal is a signal that changes slowly over time. The speech signal must be analyzed for steady acoustic properties over a short period of time. This is why speech analysis is performed on brief segments of speech that are static. These short-term measurements are made in 20- or 10-second periods. This allows for the tracking of individual speech sounds with temporal features, and this window is large enough to provide adequate spectral resolution. During the Fourier transform, hamming windows are typically utilized to improve the harmonics and minimize the edge effect.



C. **Fourier transform:** By applying the Fourier transform to the windowed frames, the magnitude domain (time to frequency) is converted.

D. Mel spectrum: After applying a Mel filter bank to a Fourier converted signal, the Mel spectrum is the following stage. This is a unit of measurement based on the frequency perceived by the human ear. Below 1 kHz, the Mel scale has linear frequency spacing, and above 1 kHz, it has logarithmic frequency spacing.


Continuous Speech
Pre-emphasis, Frame Blocking & widowing
Fast Fourier Transform (FFT)
Mel-Scale filter bank
Log |·|
Discrete Cosine Transform (DCT)
MFCC

3.2 Machine Learning Algorithms

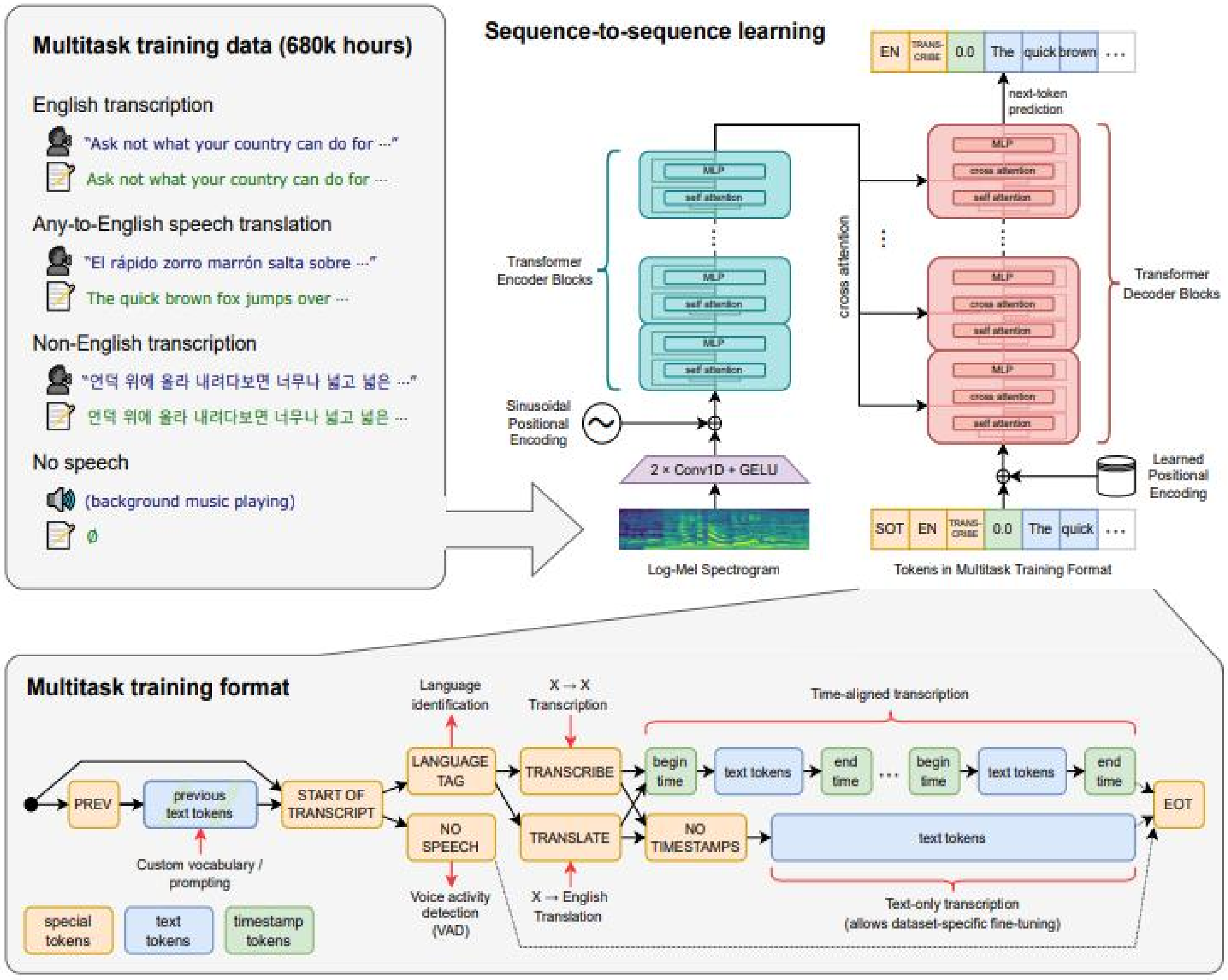
We chose SVC as our core algorithm for our system.

3.2.1. Support Vector Classifier

Support Vector Classifier (SVC) is a type of machine learning algorithm used for classification tasks. It is a variant of Support Vector Machines (SVMs) that works by dividing the data into two classes and finding the hyperplane that separates them with the largest margin. The SVC algorithm tries to maximize the margin between the decision boundary and the closest data points from each class, so as to minimize the risk of misclassification. The hyperplane is chosen such that it maximizes the margin, and the data points that are closest to the hyperplane are called support vectors. SVC can also handle non-linearly separable data by using a kernel trick to map the data into a higher-dimensional feature space, where it may become linearly separable. The performance of the SVC algorithm depends on the choice of kernel function and the regularization parameter, which can be tuned using cross-validation techniques. SVC is a powerful and widely-used classification algorithm that has been successfully applied to a wide range of applications, including image recognition, text classification, and bioinformatics.

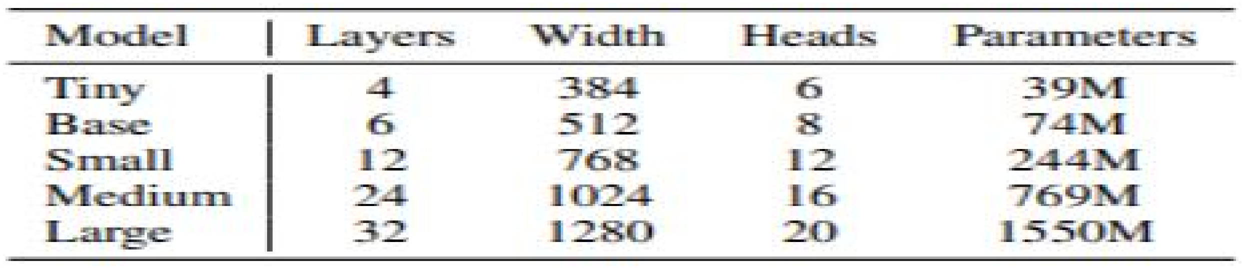
3.2.2. Model

The model used in the study is an encoder-decoder Transformer, which has been validated to scale reliably for speech recognition. All audio is re-sampled to 16,000 Hz and converted to a log-magnitude Mel spectrogram representation. Sinusoidal position embeddings are added to the output of the stem, followed by the encoder Transformer blocks, which use pre-activation residual blocks. The decoder uses learned position embeddings and tied input-output token representations. Feature normalization is globally scaled to be between -1 and 1 with approximately zero mean across the pre-training dataset. [1] The model architecture is summarized in Figure 1.


Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ···"
Ask not what your country can do for ···
Any-to-English speech translation
"El rápido zorro marrón salta sobre ···"
The quick brown fox jumps over ···
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···"
언덕 위에 올라 내려다보면 너무나 넓고 넓은 ···
No speech
(background music playing)
∅
Sequence-to-sequence learning
EN TRANSCRIBE 0.0 The quick brown ...
next-token prediction
MLP
self attention
cross attention
Transformer Encoder Blocks
Transformer Decoder Blocks
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT EN TRANSCRIBE 0.0 The quick ...
Tokens in Multitask Training Format
Multitask training format
Language identification
X → X Transcription
Time-aligned transcription
PREV
previous text tokens
START OF TRANSCRIPT
LANGUAGE TAG
NO SPEECH
TRANSCRIBE
TRANSLATE
begin time
text tokens
end time
NO TIMESTAMPS
EOT
Custom vocabulary / prompting
Voice activity detection (VAD)
X → English Translation
Text-only transcription (allows dataset-specific fine-tuning)
special tokens
text tokens
timestamp tokens

3.2.3. Training Model

The paper describes the training details of the Whisper model family, which is designed to study the scaling properties of large-scale supervised pre-training for speech recognition. Various models of different sizes are trained with data parallelism across accelerators using FP16 with dynamic loss scaling and activation checkpointing. The models are trained with AdamW and gradient norm clipping with a linear learning rate decay to zero after a warmup. Batch size, number of updates, and training hyperparameters are given. To avoid incorrect guesses for the names of speakers, the models are fine-tuned on the subset of transcripts that do not include speaker annotations. [1]

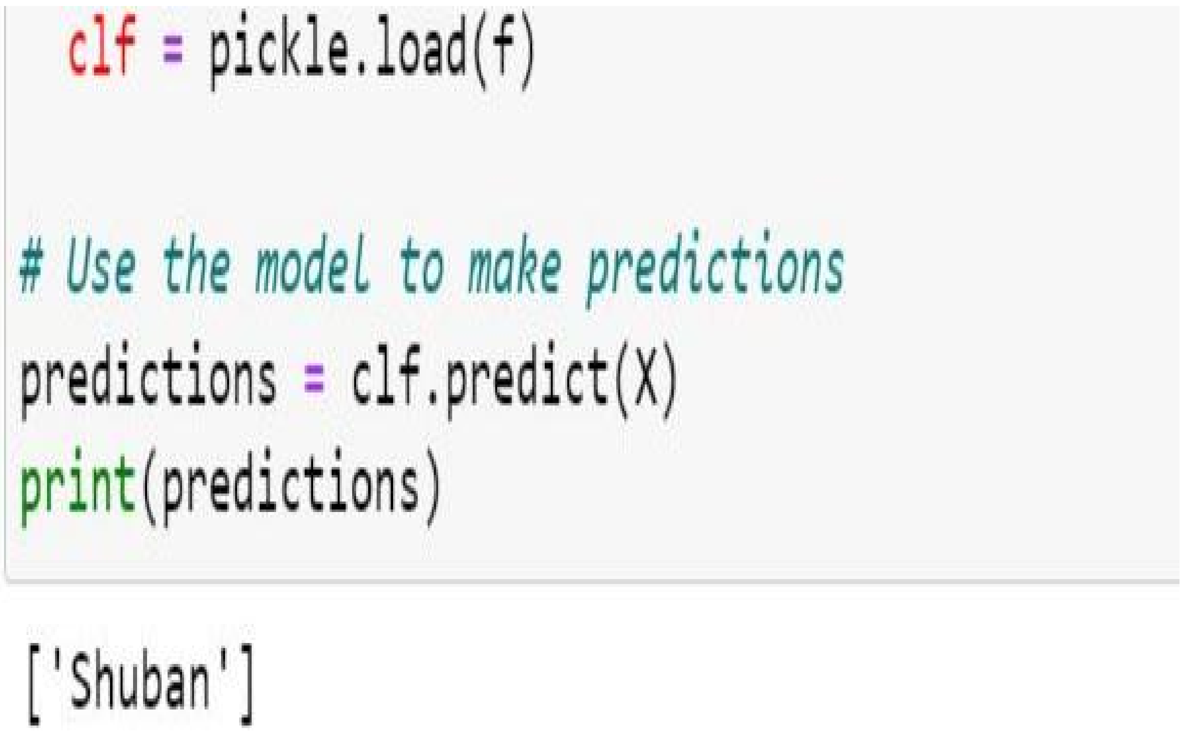


Model	Layers	Width	Heads	Parameters
Tiny	4	384	6	39M
Base	6	512	8	74M
Small	12	768	12	244M
Medium	24	1024	16	769M
Large	32	1280	20	1550M

3.2.3. Speaker Change Detection

The approach for speaker change detection involves using the error between the pitch measurement and prediction, with a threshold to determine a change in speaker. A Kalman filter is used to track the pitch of each speaker, with a new filter initialized when a change in speaker is detected. The closest Kalman pitch track to the current measurement is selected, and if the difference between the measurement and the last pitch value of the closest track is below a threshold, the previous filter is continued, otherwise a new filter is generated. This approach assumes that different speakers have different mean pitches, and the Kalman filter tracks correspond to different speakers in the audio recording. [2]

Speaker Identification



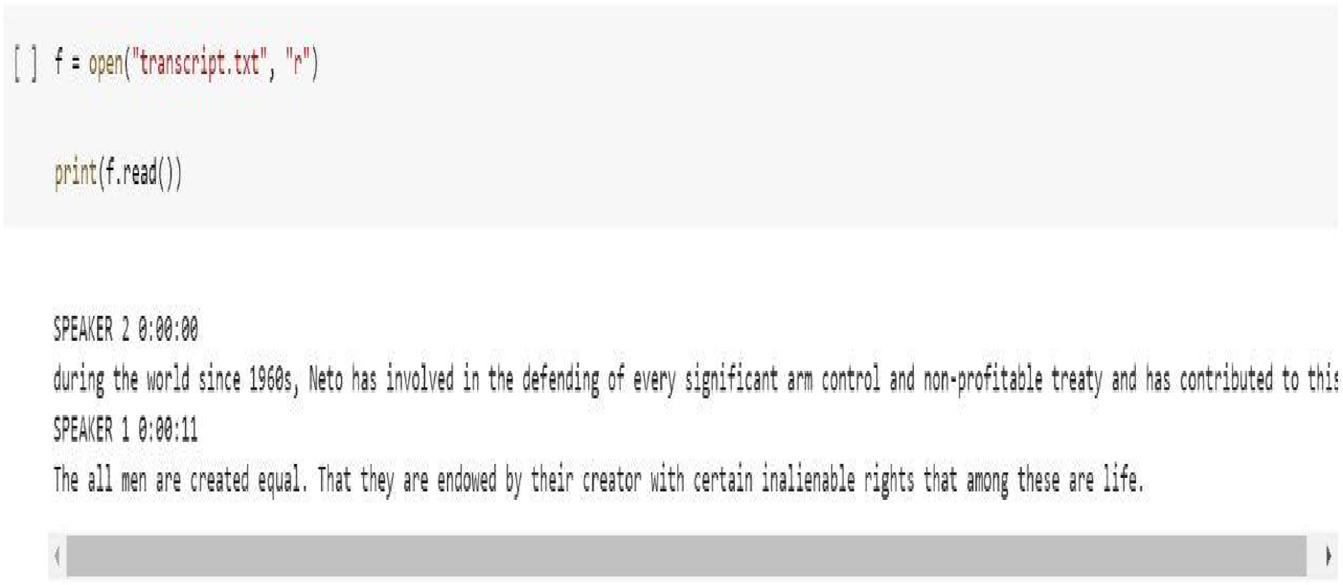
```
  clf = pickle.load(f)


# Use the model to make predictions
predictions = clf.predict(X)
print(predictions)
```

```
['Shuban']
```

Figure 3.2.1(2)

Speaker Segmentation Transcription



```
[ ] f = open("transcript.txt", "r")

    print(f.read())
```

```
SPEAKER 2 0:00:00
during the world since 1960s, Neto has involved in the defending of every significant arm control and non-profitable treaty and has contributed to this
SPEAKER 1 0:00:11
The all men are created equal. That they are endowed by their creator with certain inalienable rights that among these are life.
```

Model coding explained:

Training Speaker Identification

```
# Important Reading csv file to use
import csv

# Open the CSV file for reading
with open("audio_data.csv", "r") as f:
  reader = csv.reader(f)

  # Skip the header row
  next(reader)

  # Extract the file names and labels from the rows
  audio_files = []
  speaker_labels = []
  for row in reader:
    audio_files.append(row[0])
    speaker_labels.append(row[1])

# Import imporatnt libraries

#!pip install librosa
import numpy as np
import librosa
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
import os

# Load audio files and extract MFCCs and pitch
X = []    #Features
```

```
y = []    #Labels
# Open the CSV file
with open('audio_data.csv', 'r') as file:
  # Create a CSV reader object
  reader = csv.reader(file)

  # Skip the first row (header)
  next(reader)

  # Iterate through the rows
  for row in reader:
    file = row[0]
    label = row[1]
    #print(f'Processing file {file} with label {label}')
    audio, sr = librosa.load(file)
    freqs, times, mags = librosa.reassigned_spectrogram(audio, sr=sr,
                                    fill_nan=True)
    freqs = freqs[mags > np.median(mags)]
    mfccs = librosa.feature.mfcc(audio, sr)
    pitch = librosa.pitch_tuning(freqs)
    X.append(np.concatenate((mfccs, pitch),axis = None))
    #X.append(mfccs)
    y.append(label)




# Split data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.2)
print("X-TRAIN",X_train)
print("X-TEST",X_test)
print("Y-TRAIN",y_train)
print("Y-TEST",y_test)
```

```
#y_train.append('unknown')
# Train a support vector machine classifier on the training data
clf = SVC()
clf.fit(X_train, y_train)

# Evaluate the model on the test data
accuracy = clf.score(X_test, y_test)
print(f"Test accuracy: {accuracy:.2f}")



import pickle

# Save the model to a file
with open("new_model.pkl", "wb") as f:
  pickle.dump(clf, f)

# Load the model from a file
with open("new_model.pkl", "rb") as f:
  clf = pickle.load(f)

# Use the model to make predictions
predictions = clf.predict(X_test)
predictions
```

The Support Vector Classifier (SVC) technique is used to implement Speaker Segmentation in the given code. The method reads an audio dataset from a CSV file and uses the Librosa package to extract the Mel-frequency cepstral coefficients (MFCCs) and pitch data from the audio files. The train_test_split function from scikit-learn is then used to divide the data into training and testing sets. The SVC algorithm is then trained on the training set, and the correctness of the algorithm is assessed on the test set. The console prints the accuracy rating. Finally, a file containing the trained model is stored and then loaded to make predictions on the test set.

Code 3.2.1(4)

Prediction Speaker Identification

```
import numpy as np
import librosa
#from sklearn.externals import joblib
import pickle
# Load the audio file and extract features
audio, sr = librosa.load("Test/Shuban_Test1.wav")
mfccs = librosa.feature.mfcc(y=audio, sr=sr)
freqs, times, mags = librosa.reassigned_spectrogram(audio, sr=sr,
                                fill_nan=True)
freqs = freqs[mags > np.median(mags)]
pitch= librosa.pitch_tuning(freqs)
X = np.concatenate((mfccs, pitch),axis = None)

# Preprocess the input data
X=X.reshape(1, -1)
#X = preprocess(X)

# Load the model from a file
with open("new_model.pkl", "rb") as f:
  clf = pickle.load(f)

# Use the model to make predictions
predictions = clf.predict(X)
print(predictions)
```

This code loads an audio file using Librosa, which is a Python library for analyzing and processing audio signals. The audio file is then processed to extract features such as MFCCs and pitch using the Librosa functions "librosa.feature.mfcc" and "librosa.pitch_tuning". The extracted features are concatenated using NumPy's "np.concatenate" function and reshaped using the NumPy function "reshape". The code then loads a pre-trained SVC model from a file using the "pickle.load" function and uses it to make predictions on the input data. The predicted label is then printed to the console using the "print" function.

Code 3.2.1(5)

Prediction speaker segmentation and Transcription

```
# upload audio file
from google.colab import files
uploaded = files.upload()
path = next(iter(uploaded))

num_speakers = 2 #@param {type:"integer"}

language = 'English' #@param ['any', 'English']

model_size = 'tiny' #@param ['tiny', 'base', 'small', 'medium', 'large']



model_name = model_size
if language == 'English' and model_size != 'large':
  model_name += '.en'
```

```
!pip install -q git+https://github.com/openai/whisper.git > /dev/null
!pip install -q git+https://github.com/pyannote/pyannote-audio > /dev/null

import whisper
import datetime

import subprocess

import torch
import pyannote.audio
from pyannote.audio.pipelines.speaker_verification import PretrainedSpeakerEmbedding
embedding_model = PretrainedSpeakerEmbedding(
    "speechbrain/spkrec-ecapa-voxceleb",
    device=torch.device("cuda"))

from pyannote.audio import Audio
from pyannote.core import Segment

import wave
import contextlib

from sklearn.cluster import AgglomerativeClustering
import numpy as np

if path[-3:] != 'wav':
  subprocess.call(['ffmpeg', '-i', path, 'audio.wav', '-y'])
  path = 'audio.wav'

result = model.transcribe(path)
segments = result["segments"]

with contextlib.closing(wave.open(path,'r')) as f:
  frames = f.getnframes()
  rate = f.getframerate()
  duration = frames / float(rate)



audio = Audio()

def segment_embedding(segment):
  start = segment["start"]
  # Whisper overshoots the end timestamp in the last segment
  end = min(duration, segment["end"])
  clip = Segment(start, end)
  waveform, sample_rate = audio.crop(path, clip)
  return embedding_model(waveform[None])




embeddings = np.zeros(shape=(len(segments), 192))
for i, segment in enumerate(segments):
  embeddings[i] = segment_embedding(segment)

embeddings = np.nan_to_num(embeddings)

clustering = AgglomerativeClustering(num_speakers).fit(embeddings)
labels = clustering.labels_
for i in range(len(segments)):
  segments[i]["speaker"] = 'SPEAKER ' + str(labels[i] + 1)

def time(secs):
  return datetime.timedelta(seconds=round(secs))

f = open("transcript.txt", "w")

for (i, segment) in enumerate(segments):
  if i == 0 or segments[i - 1]["speaker"] != segment["speaker"]:
    f.write("\n" + segment["speaker"] + ' ' + str(time(segment["start"])) + '\n')
  f.write(segment["text"][1:] + ' ')
f.close()



f = open("transcript.txt", "r")

print(f.read())
```

The above code is a Python script that performs speaker diarization on an audio file uploaded by the user.

First, it imports the necessary libraries including Whisper, Pyannote-audio, and Numpy. Then it prompts the user to input the number of speakers and the language used in the audio file.

It then uses Pyannote-audio to extract speaker embeddings from the audio file. These embeddings are then clustered using Agglomerative Clustering to group the embeddings into different speakers based on similarity.

The script then writes the transcribed text into a file called 'transcript.txt', which includes the assigned speaker label, their start time, and their spoken words.

Finally, the script reads and prints the 'transcript.txt' file to the console, displaying the transcribed text with speaker labels and start times.

Code 3.2.1(6)

Chapter 4:

Software Requirement Engineering

4.1 Introduction

This chapter summarises the complete Software Requirement Specification document, including the system's purpose, scope, functional, and non-functional needs. The purpose of this article is to provide a full overview of the Speaker Segmentation and transcription project, which intends to construct a suspicious audio recognition and transcription device. This paper contains the Speaker Segmentation specific requirements.

4.1.1 Purpose

The goal of this chapter is to provide a full description of the software that does Suspicious audio identification and transcription in order to prevent terrorist activity. This is the foundation for all software development guidelines. It will also help clients guarantee that all needs and specifications are met.

4.2 Overall Description

4.2.1 Product Functions

The following are the key properties of Speaker Segmentation in this domain:

The system will identify different users in a single audio clip.
It will also make a transcription of different members in that audio (only in English).
It will help in different law and militry releated cases.

4.3 User Classes and Characteristics

The following section describes the types of users of Speaker Segmentation and transcription.

4.3.1 Security Agencies

Once the prototype is in their hands, security agencies will have access to the system and will be able to enhance the dataset to meet their needs and resources.

4.3.2 Corporate sector

Using relevant datasets, many firms will be able to keep track of their conversations while protecting their privacy.

4.4 Operating Environment

4.4.1 Hardware

Speaker Segmentation will have the following Hardware specifications:

A well updated personal computer: For processing sounds as a standalone device
High quality professional speaker audio: In order to record the audio.

4.4.2 Software

Speaker segmentation transcription will have the following Software specifications:

Python IDE
Anaconda

4.5 Design and implementation Constraints

It will be capable of handling only one audio at a time.

The dataset will be confined to a few selected words and trained on those words, with the option to increase the dataset as needed.

We'll have to meet those requirements because the system will require a lot of processing power.

4.6 User Documentation

The users will be given a user handbook with instructions on how to run Speaker Segmentation as well as the limits that must be considered. Users will also have access to a project report outlining the software's features, capabilities, and procedures

4.7 Assumptions and Dependencies

Constant electricity supply

Users must be aware of the dataset's limits and that the system will need to be trained on a different dataset if they want it to be adaptable to their company.

It will be necessary to consider the system's accuracy.

4.8 External Interface Requirements

4.8.1 User Interfaces

Front-end software: currently we only worked on developing the code we didn't work on UI front end.

Back-end software: Python

4.8.2 Hardware Interfaces

Windows Operating System(For training the machine)

4.9 Communication Interfaces

Our system uses Python in the background to scan (for suspicious terms) every audio file delivered or listened to in real time.

Req 2: The system must be able to distinguish between suspicious and non-suspicious speakers.

5. Notify the user if there are any dubious audio.

Priority and Description

If it detects any questionable terms in the audio file or call, Speaker Segmentation will notify the user instantly.

It is a high priority since the user must be notified of questionable audio in order to take action to stop it.

Sequences of Stimulus/Response

1. Audio files will be uploaded by the user.

2. The machine will compare the audio words to the dataset's .

3. If suspicious speakers are detected, the user will receive a warning on the system.

Requirements for Function

Req 1: The user should be able to receive appropriate alerts when questionable terms are entered.

6. Update the dataset/list of suspicious words

Priority and Description

1. More suspicious words will be added to the dataset by the user.

2. After that, the user will train the system to recognise new suspicious terms.

3. The user will put the system to the test by listening to the audio files for new suspicious terms.

Requirements for Function

Req 1: Users should be able to add new suspicious words to the dataset with ease.

Req 2: The user should be able to test the system for new suspicious speakers.

4.10 Non-functional Requirements

4.10.1 Requirements for Performance

The software's response time must not be very long, and the application must run on operating systems with high processing power.
The system's accuracy should be sufficient to make it a trusted system.

4.10.2 Requirements for Safety

Any user's information/audio must be handled carefully by the application.
Users must register with accurate information so that if an error arises, the service can assist him as much as possible.
User credentials and personal information are not to be shared with other users.

4.10.3 Requirements for Security

Only authorized users have access to the system, which prevents unauthorized alterations.
Unauthorized persons shall not have access to the system.

4.10.4 Attributes of Software Quality

Availability: If the user meets the software requirements, he can access the software whenever he wants.
Maintainability: New suspicious terms should be added to the dataset to keep it up to date.
Reusability: The system's components must be written in such a way that they can be reused.

Chapter 5

Analysis

Speaker segmentation and transcription are important tasks in speech processing and natural language processing. Speaker segmentation refers to the task of dividing an audio signal into segments corresponding to different speakers. This is often done in order to identify and label the speakers in a conversation, and can be useful for a range of applications such as speech-to-text transcription, speaker diarization, and sentiment analysis.

One common approach to speaker segmentation is to use clustering algorithms to group together speech segments that are likely to belong to the same speaker. In the code example provided, the AgglomerativeClustering algorithm is used to cluster embeddings of speech segments that are generated using a pre-trained speaker embedding model (the "speechbrain/spkrec-ecapa-voxceleb" model). The embeddings are generated by first segmenting the audio signal using the Whisper voice activity detection algorithm, and then extracting speech segments from the audio signal and applying the embedding model to each segment.

The resulting speaker labels are then applied to the text transcription of the audio signal. In the code example, this is done by iterating over the segments and adding a "speaker" field to each segment, which is set to a label indicating the speaker identified by the clustering algorithm. The resulting transcription includes these speaker labels, along with the text for each segment.

In addition to the clustering algorithm used for speaker segmentation, the example code also uses a Support Vector Classification (SVC) algorithm for speaker verification. The SVC algorithm is used to train a model to distinguish between different speakers based on their voice characteristics, and can be used to identify whether two audio segments are spoken by the same speaker or not. In the code example, a pre-trained speaker embedding model is used to extract features from audio segments, which are then fed into an SVC model for classification. The resulting model can be used for tasks such as speaker verification and speaker identification.

Overall, speaker segmentation and transcription are important tasks in speech processing and natural language processing, and can be used for a range of applications such as speech-to-text transcription, speaker diarization, and sentiment analysis. The example code provided demonstrates one approach to performing these tasks using clustering and classification algorithms, and highlights the importance of pre-processing audio signals and using pre-trained models for feature extraction and classification.

Chapter 6

Future Work

Speaker segmentation can be further developed to automate speech recognition on a big scale. To reach as many individuals as possible, we will endeavour to provide more data to our project. The technology could attain greater accuracy with more data. Furthermore, we must cover all available datasets of our desired folks in order to rely on it as an accurate gadget. This database will be shared with researchers who desire to work on any such system for the greater good. We've developed a system application so far, but we'll strive to convert it to a standalone device so that security agencies may deploy it quickly.

6.1 Increasing Accuracy

In our current system, we have implemented the SVC (support vector theorems) algorithm. However, as we move forward, we plan to integrate more advanced and sophisticated algorithms that can enhance the system's performance and accuracy. By exploring and implementing state-of-the-art algorithms, we aim to improve the overall efficiency of our system, leading to better results and outcomes.

6.2 Enhancing Dataset

Our current data set includes 150 audio files and 3 identified speakers. However, in order to improve the accuracy and efficiency of our system, we need to increase the size of our data set. By adding more audio files with a larger variety of speakers and speech patterns, we can improve the performance of our system and ensure that it can accurately identify a wider range of speakers.

Expanding our data set will allow us to better train our system to recognize patterns and variations in speech that may be unique to different speakers. This can help to reduce errors and false positives, leading to more accurate speaker identification. By continuously increasing the size and diversity of our data set, we can ensure that our system stays up-to-date and capable of accurately identifying speakers in a wide range of scenarios.

6.3 Standalone Device

Converting the existing system into a web application would make it more user-friendly and accessible to a wider audience. The current system requires knowledge of coding and programming to operate, which can be a barrier for many users. A web application would simplify the process by providing a user interface that is easy to navigate, and would not require any coding skills. This would make the system more accessible to individuals and organizations who may not have the technical expertise to operate the current system.

Furthermore, a web application would offer greater flexibility and scalability compared to the current code-based system. With a web application, users could access the system from anywhere with an internet connection, making it more convenient to use. Additionally, a web application would be easier to maintain and update, as changes could be made to the application without requiring users to download and install new software. This would enable the system to evolve and improve over time, ensuring that it remains effective and useful for its users.

Conclusion

Speaker segmentation and transition detection are essential tasks in many audio processing applications, including speech recognition, speaker diarization, and speaker verification. With the growing demand for automated audio analysis, developing accurate and efficient methods for speaker segmentation and transition detection has become increasingly important. In this regard, various algorithms have been proposed in the literature, ranging from classical statistical methods to more recent deep learning-based approaches. These methods have shown promising results in various scenarios and are continuously being improved to achieve better performance.

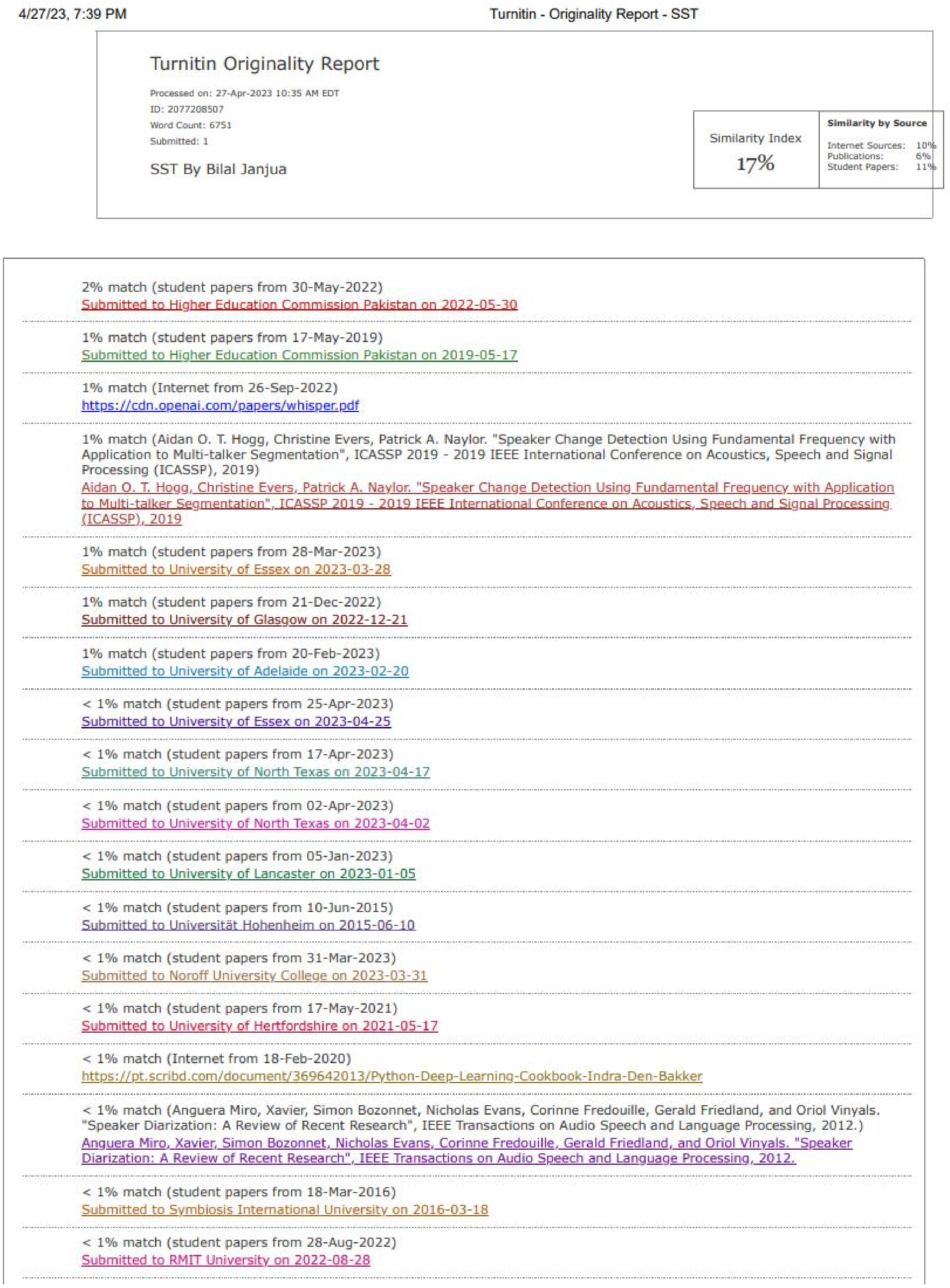
In conclusion, speaker segmentation and transition detection are critical components of many audio processing systems, and their accuracy and efficiency are crucial for the success of such systems. While various algorithms have been proposed, the choice of the most suitable method depends on the specific requirements of the application. With the continuous advancements in machine learning and signal processing techniques, we can expect to see further improvements in speaker segmentation and transition detection, leading to more robust and reliable audio analysis systems.

References

[1] A. R. *. 1. J. W. K. *. 1. T. X. 1. G. B. 1. C. M. 1, "Robust Speech Recognition via Large-Scale Weak Supervision," *arXiv:2212.04356v1 ,* 2022.

[2] C. E. Aidan O. T. Hogg, "Speaker Change Detection using Fundamental Frequency with Application to Multi-talker Segmentation," *research gate,* 2019.

TURN IT IN ORIGINALITY REPORT



4/27/23, 7:39 PM Turnitin - Originality Report - SST

Turnitin Originality Report

Processed on: 27-Apr-2023 10:35 AM EDT
ID: 2077208507
Word Count: 6751
Submitted: 1

SST By Bilal Janjua

Similarity Index	Similarity by Source	
17%	Internet Sources:	10%
	Publications:	6%
	Student Papers:	11%

2% match (student papers from 30-May-2022)
Submitted to Higher Education Commission Pakistan on 2022-05-30

1% match (student papers from 17-May-2019)
Submitted to Higher Education Commission Pakistan on 2019-05-17

1% match (Internet from 26-Sep-2022)
https://cdn.openai.com/papers/whisper.pdf

1% match (Aidan O. T. Hogg, Christine Evers, Patrick A. Naylor. "Speaker Change Detection Using Fundamental Frequency with Application to Multi-talker Segmentation", ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019)
Aidan O. T. Hogg, Christine Evers, Patrick A. Naylor. "Speaker Change Detection Using Fundamental Frequency with Application to Multi-talker Segmentation", ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019

1% match (student papers from 28-Mar-2023)
Submitted to University of Essex on 2023-03-28

1% match (student papers from 21-Dec-2022)
Submitted to University of Glasgow on 2022-12-21

1% match (student papers from 20-Feb-2023)
Submitted to University of Adelaide on 2023-02-20

< 1% match (student papers from 25-Apr-2023)
Submitted to University of Essex on 2023-04-25

< 1% match (student papers from 17-Apr-2023)
Submitted to University of North Texas on 2023-04-17

< 1% match (student papers from 02-Apr-2023)
Submitted to University of North Texas on 2023-04-02

< 1% match (student papers from 05-Jan-2023)
Submitted to University of Lancaster on 2023-01-05

< 1% match (student papers from 10-Jun-2015)
Submitted to Universität Hohenheim on 2015-06-10

< 1% match (student papers from 31-Mar-2023)
Submitted to Noroff University College on 2023-03-31

< 1% match (student papers from 17-May-2021)
Submitted to University of Hertfordshire on 2021-05-17

< 1% match (Internet from 18-Feb-2020)
https://pt.scribd.com/document/369642013/Python-Deep-Learning-Cookbook-Indra-Den-Bakker

< 1% match (Anguera Miro, Xavier, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. "Speaker Diarization: A Review of Recent Research", IEEE Transactions on Audio Speech and Language Processing, 2012.)
Anguera Miro, Xavier, Simon Bozonnet, Nicholas Evans, Corinne Fredouille, Gerald Friedland, and Oriol Vinyals. "Speaker Diarization: A Review of Recent Research", IEEE Transactions on Audio Speech and Language Processing, 2012.

< 1% match (student papers from 18-Mar-2016)
Submitted to Symbiosis International University on 2016-03-18

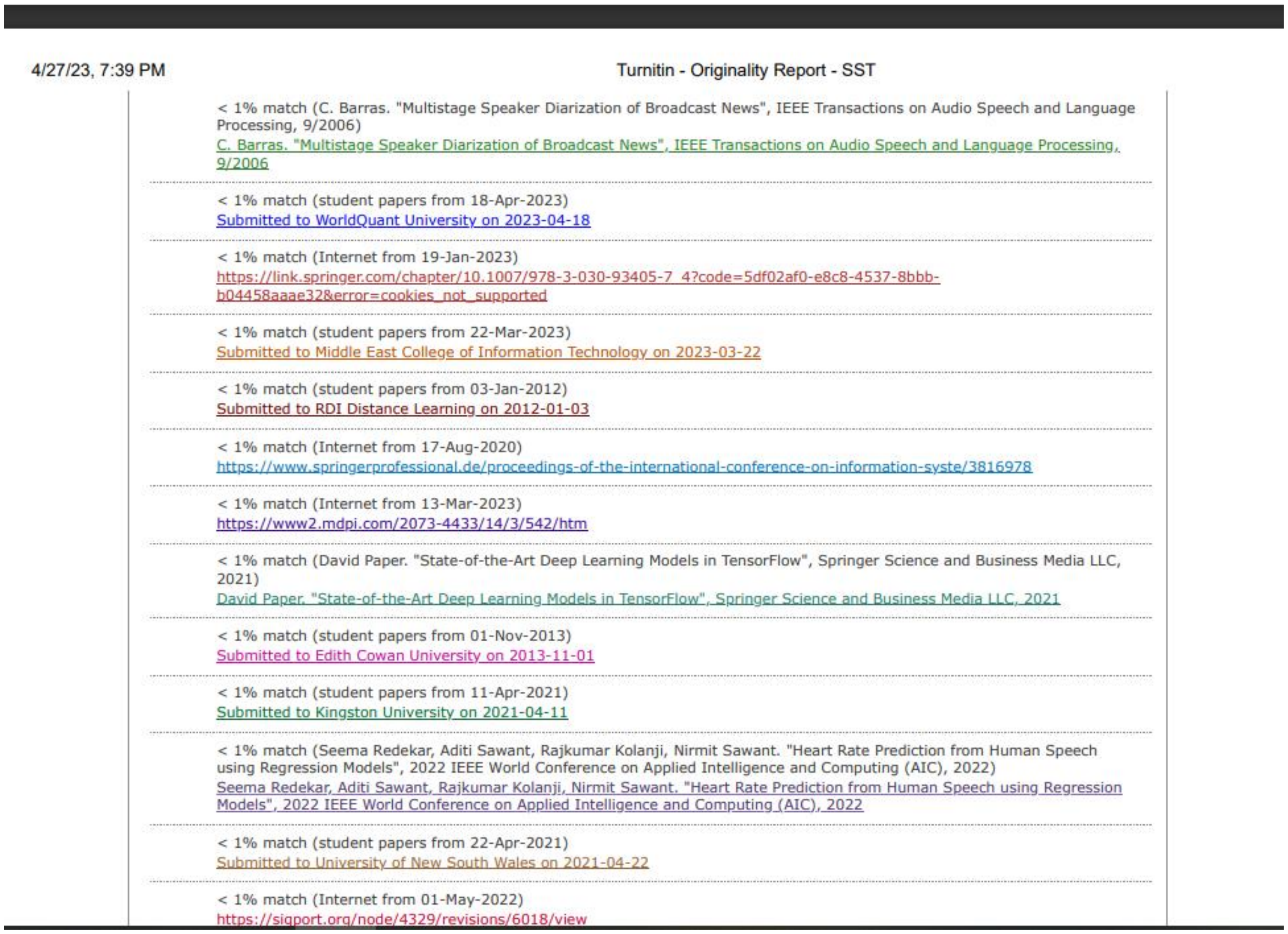
< 1% match (student papers from 28-Aug-2022)
Submitted to RMIT University on 2022-08-28

< 1% match (student papers from 15-Apr-2023)
Submitted to Amrita Vishwa Vidyapeetham on 2023-04-15

< 1% match (student papers from 04-Jan-2022)
Submitted to University of Sussex on 2022-01-04

< 1% match (Internet from 28-Oct-2022)
https://www.coursehero.com/file/138392808/SRS-i200946-i200958pdf/

https://www.turnitin.com/newreport_printview.asp?eq=0&eb=0&esm=3&oid=2077208507&sid=0&n=0&m=2&svr=54&r=43.99410474909442&lang=en_... 1/8


4/27/23, 7:39 PM Turnitin - Originality Report - SST

< 1% match (C. Barras. "Multistage Speaker Diarization of Broadcast News", IEEE Transactions on Audio Speech and Language Processing, 9/2006)
C. Barras. "Multistage Speaker Diarization of Broadcast News", IEEE Transactions on Audio Speech and Language Processing, 9/2006

< 1% match (student papers from 18-Apr-2023)
Submitted to WorldQuant University on 2023-04-18

< 1% match (Internet from 19-Jan-2023)
https://link.springer.com/chapter/10.1007/978-3-030-93405-7_4?code=5df02af0-e8c8-4537-8bbb-b04458aaae32&error=cookies_not_supported

< 1% match (student papers from 22-Mar-2023)
Submitted to Middle East College of Information Technology on 2023-03-22

< 1% match (student papers from 03-Jan-2012)
Submitted to RDI Distance Learning on 2012-01-03

< 1% match (Internet from 17-Aug-2020)
https://www.springerprofessional.de/proceedings-of-the-international-conference-on-information-syste/3816978

< 1% match (Internet from 13-Mar-2023)
https://www2.mdpi.com/2073-4433/14/3/542/htm

< 1% match (David Paper. "State-of-the-Art Deep Learning Models in TensorFlow", Springer Science and Business Media LLC, 2021)
David Paper. "State-of-the-Art Deep Learning Models in TensorFlow", Springer Science and Business Media LLC, 2021

< 1% match (student papers from 01-Nov-2013)
Submitted to Edith Cowan University on 2013-11-01

< 1% match (student papers from 11-Apr-2021)
Submitted to Kingston University on 2021-04-11

< 1% match (Seema Redekar, Aditi Sawant, Rajkumar Kolanji, Nirmit Sawant. "Heart Rate Prediction from Human Speech using Regression Models", 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), 2022)
Seema Redekar, Aditi Sawant, Rajkumar Kolanji, Nirmit Sawant. "Heart Rate Prediction from Human Speech using Regression Models", 2022 IEEE World Conference on Applied Intelligence and Computing (AIC), 2022

< 1% match (student papers from 22-Apr-2021)
Submitted to University of New South Wales on 2021-04-22

< 1% match (Internet from 01-May-2022)
https://sigport.org/node/4329/revisions/6018/view

4/27/23, 7:39 PM Turnitin - Originality Report - SST

https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/5213/1/Cap15.pdf

< 1% match (Ali AlQahtani, Thamraa Alshyab. "Wi-Fi Radio Wave-Based Continuous Zero-Effort Two-Factor Authentication Utilizes Machine Learning", Institute of Electrical and Electronics Engineers (IEEE), 2023)
Ali AlQahtani, Thamraa Alshyab. "Wi-Fi Radio Wave-Based Continuous Zero-Effort Two-Factor Authentication Utilizes Machine Learning", Institute of Electrical and Electronics Engineers (IEEE), 2023

< 1% match (Kotti, M., D. Ververidis, G. Evangelopoulos, I. Panagakis, C. Kotropoulos, P. Maragos, and I. Pitas. "Audio-Assisted Movie Dialogue Detection", IEEE Transactions on Circuits and Systems for Video Technology, 2008.)
Kotti, M., D. Ververidis, G. Evangelopoulos, I. Panagakis, C. Kotropoulos, P. Maragos, and I. Pitas. "Audio-Assisted Movie Dialogue Detection", IEEE Transactions on Circuits and Systems for Video Technology, 2008.

< 1% match (Kyu J. Han. "The SAIL speaker diarization system for analysis of spontaneous meetings", 2008 IEEE 10th Workshop on Multimedia Signal Processing, 10/2008)
Kyu J. Han. "The SAIL speaker diarization system for analysis of spontaneous meetings", 2008 IEEE 10th Workshop on Multimedia Signal Processing, 10/2008

< 1% match (Radhakrishan, Regunathan, Ziyou Xiong, Ajay Divakaran, Bhiksha Raj, Sethuraman Panchanathan, and Tong Zhang. "", Internet Multimedia Management Systems IV, 2003.)
Radhakrishan, Regunathan, Ziyou Xiong, Ajay Divakaran, Bhiksha Raj, Sethuraman Panchanathan, and Tong Zhang. "", Internet Multimedia Management Systems IV, 2003.

< 1% match (Shashank Mohan Jain. "Chapter 5 Tasks Using the Hugging Face Library", Springer Science and Business Media LLC, 2022)
Shashank Mohan Jain. "Chapter 5 Tasks Using the Hugging Face Library", Springer Science and Business Media LLC, 2022

< 1% match (Shen, Han-Ping, Jui-Feng Yeh, and Chung-Hsien Wu. "Speaker Clustering Using Decision Tree-Based Phone Cluster Models With Multi-Space Probability Distributions", IEEE Transactions on Audio Speech and Language Processing, 2011.)
Shen, Han-Ping, Jui-Feng Yeh, and Chung-Hsien Wu. "Speaker Clustering Using Decision Tree-Based Phone Cluster Models With Multi-Space Probability Distributions", IEEE Transactions on Audio Speech and Language Processing, 2011.

< 1% match (Internet from 11-Apr-2023)
https://archive.org/stream/1988influenceoftheislamiclawofwaqfonthedevelopmentofthetrustsinenglandmonicagaudiosi/India%20Waqf%20Admin%20Copy_djvu.txt

< 1% match (Internet from 03-Mar-2023)
https://dokumen.pub/tensorflow-20-computer-vision-cookbook-implement-machine-learning-solutions-to-overcome-various-computer-vision-challenges-183882913x-9781838829131.html

< 1% match (Internet from 08-Dec-2021)
https://ir.lib.uth.gr/xmlui/bitstream/handle/11615/56361/22965.pdf?isAllowed=y&sequence=1

< 1% match (Internet from 15-Jan-2022)
https://ogma.newcastle.edu.au/vital/access/%20/services/Download/uon:36253/ATTACHMENT01

< 1% match (Internet from 28-Sep-2017)
http://prr.hec.gov.pk/Thesis/2259S.pdf

< 1% match (Internet from 06-Aug-2022)
https://vdoc.pub/documents/applied-machine-learning-with-python-40dl389e5nt0

< 1% match (Orhan Gazi Yalçın. "Applied Neural Networks with TensorFlow 2", Springer Science and Business Media LLC, 2021)
Orhan Gazi Yalçın. "Applied Neural Networks with TensorFlow 2", Springer Science and Business Media LLC, 2021