

2021

Evaluating the Performance of Using Speaker Diarization for Speech Separation of In-Person Role-Play Dialogues

Raveendra Medaramitta
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Repository Citation

Medaramitta, Raveendra, "Evaluating the Performance of Using Speaker Diarization for Speech Separation of In-Person Role-Play Dialogues" (2021). *Browse all Theses and Dissertations*. 2555.
https://corescholar.libraries.wright.edu/etd_all/2555

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

EVALUATING THE PERFORMANCE OF USING SPEAKER
DIARIZATION FOR SPEECH SEPARATION OF IN-PERSON
ROLE-PLAY DIALOGUES

A Thesis submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Engineering

by

RAVEENDRA MEDARAMITTA

B.Tech., Jawaharlal Nehru Technological University, India, 2015

M.S., Wright State University, U.S.A, 2018

2021

Wright State University

WRIGHT STATE UNIVERSITY
GRADUATE SCHOOL

12/08/2021

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Raveendra Medaramitta ENTITLED Evaluating the Performance of Using Speaker Diarization for Speech Separation of In-Person Role-Play Dialogues BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Computer Engineering.

Yong Pei, Ph.D.
Thesis Director

Michael L. Raymer, Ph.D.
Chair, Department of Computer Science
and Engineering

Committee on Final Examination:

Yong Pei, Ph.D. (Advisor)

Paul J. Hershberger, Ph.D. (Co-Advisor)

Jack S. Jean, Ph.D.

Barry Milligan, Ph.D.
Vice Provost for Academic Affairs
Dean of the Graduate School

ABSTRACT

Medaramitta, Raveendra M.S.C.E., Department of Computer Science and Engineering, Wright State University, 2021. Evaluate the Performance of Using Speaker Diarization for Speech Separation of In-Person Role-Play Dialogues.

Development of professional communication skills, such as motivational interviewing, often requires experiential learning through expert instructor-guided role-plays between the trainee and a standard patient/actor. Due to the growing demand for such skills in practices, e.g., for health care providers in the management of mental health challenges, chronic conditions, substance misuse disorders, etc., there is an urgent need to improve the efficacy and scalability of such role-play based experiential learning, which are often bottlenecked by the time-consuming performance assessment process. WSU is developing ReadMI (Real-time Assessment of Dialogue in Motivational Interviewing) to address this challenge, a mobile AI solution aiming to provide automated performance assessment based on ASR and NLP. The main goal of this thesis research is to investigate current commercially available speaker diarization capabilities and evaluate their performance in separating the speeches between the trainee and the standard patient/actor in an in-person role-play training environment where the crosstalk could interfere with the operation and performance of ReadMI.

Specifically, this thesis research has: 1.) identified the major commercially-available speaker diarization systems, such as those from Google, Amazon, IBM, and Rev.ai; 2.) designed and implemented corresponding evaluation systems that integrate these commercially available cloud services for operating in the in-person role-play training environments; and, 3.) completed

an experimental study that evaluated and compared the performance of the speaker diarization services from Google and Amazon. The main finding of this thesis is that the current speaker diarization capabilities alone are not able to provide sufficient performance for our particular use case when integrating them into ReadMI for operating in in-person role-play training environments. But this thesis research potentially provides a clear baseline reference to future developers for integrating future speaker diarization capabilities into similar applications.

TABLE OF CONTENTS

CHAPTER 1	1
1.1. Motivational Interviewing:.....	1
1.2. Role-Plays in Motivational Interviewing Skill Development:.....	1
1.3. Real-time Assessment of Dialogue in Motivational Interviewing:	2
1.3.1. Problem Statement:	3
1.4. Motivation:	4
CHAPTER 2	7
2.1. Speaker Diarization	7
2.2. Survey of commercially available ASR service with Speaker diarization capabilities: ..	7
2.2.1. Google Speech-to-text:	7
2.2.2. Amazon transcribe:	8
2.2.3. Rev.ai:	9
2.2.4. IBM Watson Speech to Text:.....	9
CHAPTER 3	11
3.1. Block Diagram Representing Speaker Diarization Evaluation set-up:.....	11
3.2. Google Speech-to-text Speaker Diarization:.....	12

3.3. Amazon Transcribe Speaker Diarization:.....	18
CHAPTER 4	23
4.1. Examine effects of Crosstalk in ReadMI:	23
4.2. Accuracy of transcripts with speaker diarization feature enabled for Google and Amazon ASR in real-time streaming mode:.....	25
4.3. Speaker diarization accuracy measure for Google Cloud Speech API and Amazon Transcribe in offline batch text transcription mode:	26
4.4. Gender based roleplay accuracies:	27
4.5. Discussion:	30
CHAPTER 5	31
5.1. Conclusion:.....	31
5.2. Future work:	32
REFERENCES	33

LIST OF FIGURES

Figure 1: ReadMI crosstalk diagram.....	3
Figure 2: Inaccurate transcription due to crosstalk	4
Figure 3: Evaluation set up for speaker diarization	11
Figure 4: Snippet showing google speech-to-text speaker diarization configuration.....	13
Figure 5: Snippet showing recognitionConfig parameters.	14
Figure 6: parameter configuration for speaker diarization in batch transcription	15
Figure 7: Test environment for google cloud speech-to-text speaker diarization transcripts.....	16
Figure 8: Google cloud storage bucket with audio files for transcription	17
Figure 9: Screenshot showing Google ASR speaker diarization batch transcription	18
Figure 10: Snippet showing Amazon transcribe request with speaker diarization feature.....	19
Figure 11: User interface of Amazon streaming speaker diarization	20
Figure 12: Amazon S3 bucket with audio files for transcription.....	21
Figure 13: Amazon management console’s amazon transcribe interface.....	22
Figure 14: Test environment for processing .json file for final transcripts with speaker labelling	22
Figure 15: Percentage of utterances effected with cross talk.....	24
Figure 16: Real-time streaming speaker diarization amazon vs google	25
Figure 17: Accuracy percentages of batch transcription in Google and Amazon	26
Figure 18: Gender specific roleplays for streaming speaker diarization	28
Figure 19: Gender specific speaker diarization for batch transcription	29

LIST OF TABLES

Table 1: Real-time vs batch speaker diarization accuracy percentages for google cloud speech

ASR and Amazon Transcribe 27

Acknowledgement

I cannot express enough thanks to my committee members for their continuous support and encouragement throughout my thesis: Dr. Yong Pie, my advisor, for believing in me even in my hard times and making me confident in the process through your support and guidance, it wouldn't have been possible going through all the hurdles without your help till now. I can't even express on how to show my gratitude to you.

I would like to thank Dr. Paul J. Hershberger, my co-advisor, for guiding me through the entire period of my research related to ReadMI. I would also like to thank Dr. Jack S. Jean for enlightening and building my interests and foundations in Embedded Systems.

My completion of the project would not have been possible, thank you for the support given by my thesis mentor Ashutosh Sivakumar who helped me achieve this feat through his healthy discussions that enlightened me in the process to complete my project and my master's on time.

Finally, I would like to thank my younger brother who stood beside me and understood my situation in this process. Thank you.

CHAPTER 1

1.1. Motivational Interviewing:

Motivational Interviewing [1][2] is a client-focused behavioral therapy technique that aims to motivate a healthy behavior change through clinician-client shared decision making. Motivational Interviewing generally involves a verbal interaction in counseling sessions between the client and the clinician, which aims to reduce patient ambivalence to change progressively. William Miller and Stephen Rollnick introduced it to primarily help people overcome alcohol addiction [3].

1.2. Role-Plays in Motivational Interviewing Skill Development:

Motivational Interviewing technique education is adopted into the medical sciences curriculum as “Role-Play” sessions between medical students and a “Simulated Patient,” usually a well-trained actor who simulates patient profiles selected according to demographic variables like age, sex, and profession. Roleplays allow medical students to develop the requisite skills to counsel patients [4].

According to current practice, MI roleplay evaluation workflow is a manual process. It involves manual transcription of audio files, behavioral coding [5], and MI-consistent metrics evaluation [6]. This process is cost-intensive both in terms of workforce and time [5] [6]. The delayed feedback due to the manual evaluation minimizes the effective transference of MI knowledge to MI students [5] [6].

1.3. Real-time Assessment of Dialogue in Motivational Interviewing:

ReadMI (Real-time Assessment of Dialogue in Motivational Interviewing), a Mobile Cloud Computing based solution effectively addressing the shortcoming, stands out as the technological solution that automatically transcribes the clinician-client conversations with 95% accuracy [6] and uses Natural Language Processing based algorithm that assigns MI-consistent behavioral codes to the clinician-client dialogue and provides feedback instantaneously (< 1 second).

In an in-person role-play training session, the typical setup of ReadMI consists of the MI dialogue participants speaking into their respective microphones, each connected to a separate tablet computer. The use of two sets of tablet/microphones is intended to separate the participant's speech via a simple hardware approach, although it incurs additional cost. This setup, however, might still suffer from the problem of audio "Crosstalk" [7], where the voice signal of one person could be captured by the opposite microphone resulting in noisy speech transcriptions consisting of texts from both speakers interspersed together. The crosstalk might be particularly severe when only a built-in microphone or low-cost commodity microphones are used. The problem could be further worsened due to the speakers' unequal voice volume and intonation. In theory, this problem of "audio crosstalk" could be mitigated by applying a combination of computational linguistics and Machine Learning based technique called "Speaker Diarization" [8]. Moreover, if Speaker Diarization can produce sufficiently accurate speech separation, only one tablet/microphone will be needed for a training session instead of the current two sets setup.

Advanced speaker diarization algorithms can be integrated into applications by software developers through API (Application Programming Interfaces). The Cloud based implementations of these APIs are provided by technology companies like Google, IBM, Amazon, etc., where

speaker detection and diarization algorithms are combined with speech-to-text algorithms. Individual speakers are identified through speaker tags, and their respective speeches are transcribed in real-time or in batches from pre-recorded audio files.

1.3.1. Problem Statement:

ReadMI uses microphones for each participant in the clinician-client dialogue for recording voice signals and transmits them via Bluetooth to the corresponding receivers connected to their respective mobile devices. Typically, participants are seated close to each other to resemble the clinic encountering approximately 6 feet apart. This proximity, however, may allow the audio feed from one participant to be captured by the microphone of the opposite participant, resulting in erroneous speech-to-text transcriptions across participants and affecting the accuracy of the subsequent analysis.

Typically, crosstalk is defined as interference in one channel from another. In this case, we interpret this definition of crosstalk as appearances of noisy erroneous utterances in the final textual representation of each participant’s speech [9].

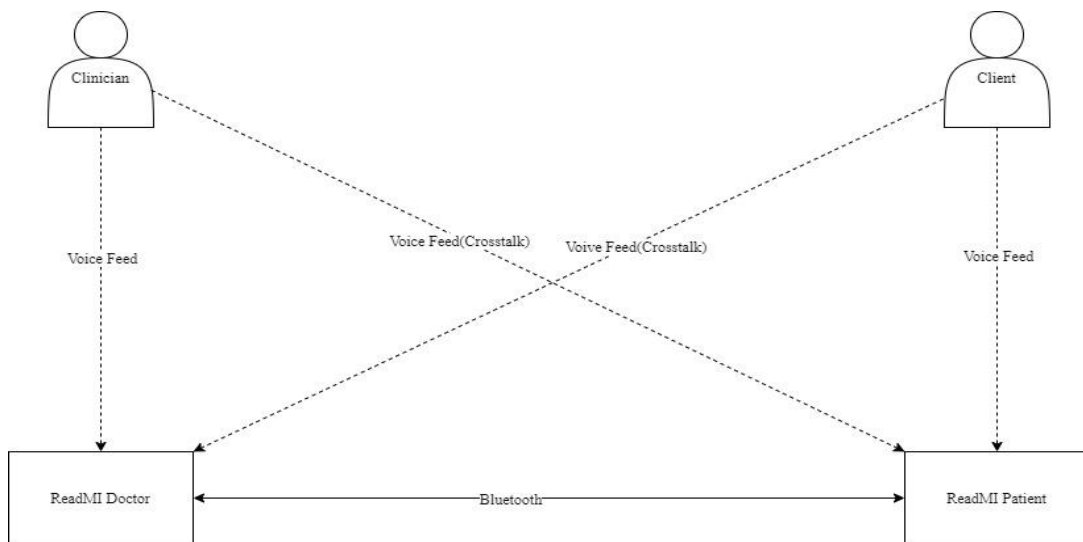


Figure 1: ReadMI crosstalk block diagram

Figure 2 represents inaccurate transcription due to crosstalk where we can see the transcripts on the left side of the picture under the headline labeled 'SPEECH TO TEXT,' with one's speech in the grey background and the others in green. The right-side part of the picture labeled as 'INTERVIEW ANALYSIS' represents the analytics that ReadMI gives as feedback to the roleplay session doctor. In the above screenshot, the utterances affected by cross talk are marked with red bubbles.

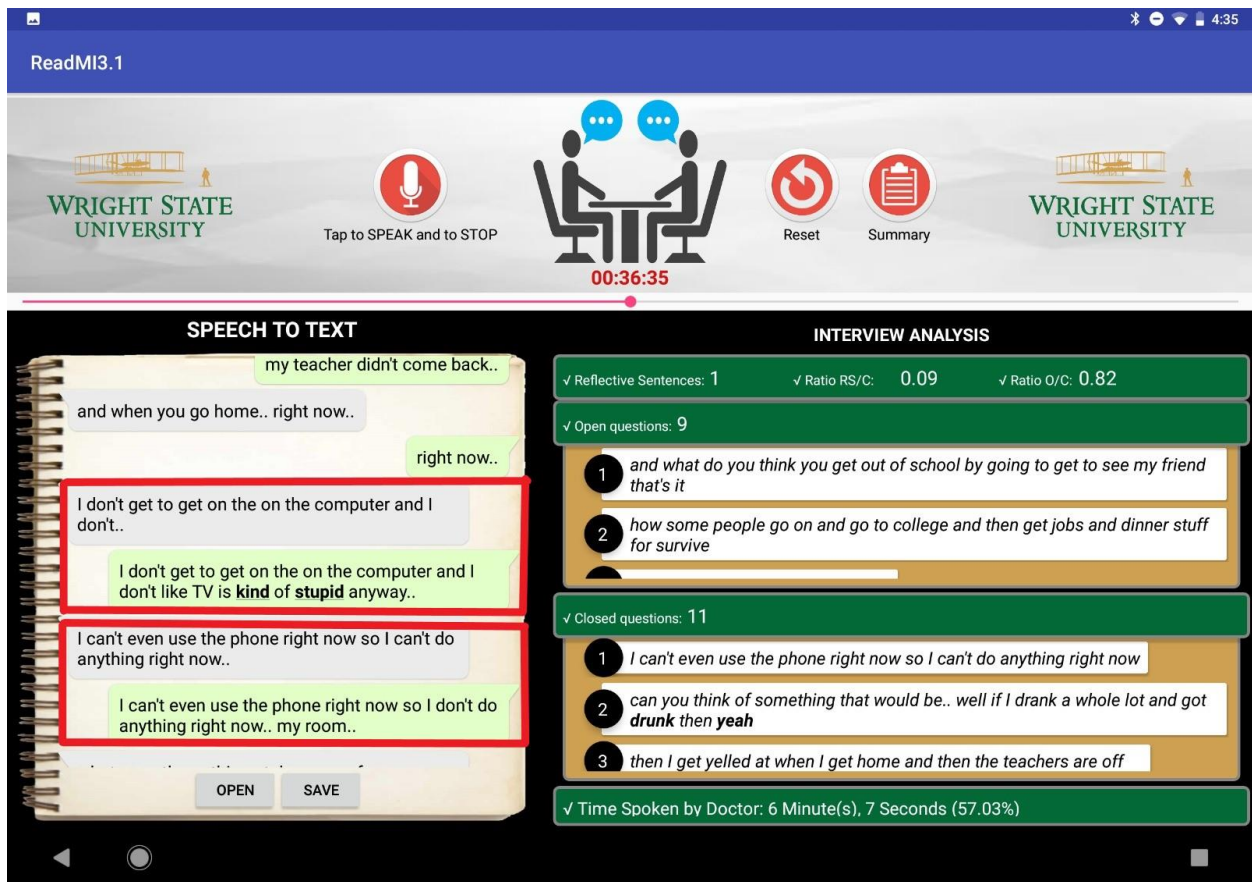


Figure 2: Inaccurate transcription due to crosstalk

1.4. Motivation:

One of the enabling technologies that facilitate the timely delivery of MI feedback in ReadMI is the Automatic Speech Recognition (ASR) algorithm. ASRs are used in applications as

APIs (Application Programming Interfaces). Companies like Google, IBM, Amazon, Rev.ai, IBM Speech-to-text have deployed ASR-based APIs as cloud-based services that enable real-time speech-to-text applications like ReadMI. These APIs provide services like real-time and recorded audio transcriptions and automatic speaker detection and separation also referred to as “Speaker Diarization” [8]. Speaker diarization allows applications to recognize multiple speakers and transcribe their respective speeches automatically. Speaker Diarization allows for a dialogue between multiple participants to be transmitted on a single audio channel. At the receiving end, the deep neural network-based speaker diarization equipped application will automatically tag the individual utterances with “speaker tag” and separate the utterances based on speakers resulting in a significant reduction in crosstalk, resources like separate recording devices per audio channel per speaker, and improvement in user experience.

The efficacy and accuracy of automatic speech recognition have matured to the extent that its utility has become ubiquitous, examples include AI assistants in smartphones, cars, refrigerators, smart televisions, and text editors’ applications like Microsoft word. Speaker diarization would incorporate increased autonomy in dialogue analysis applications where the speakers are automatically detected, and audio transcribed without the explicit assignment of separate recording devices for each participant reducing audio crosstalk, equipment costs and improve user experience. On the other hand, there is limited literature evaluating the effectiveness of commercially available speaker diarization services. In particular, we did not find any surveys available comparing this feature in supporting role-play sessions between different Speech Recognition Systems.

In this thesis, 1) We examined the crosstalk in the existing ReadMI set up to establish a baseline in speaker separation; 2) We demonstrate the use of speaker diarization APIs in

overcoming crosstalk; 3) We evaluate both real-time streaming and batch diarization capabilities of Google Cloud Speech to Text API and Amazon Transcribe API to evaluate speaker separation accuracy through speaker tag assignment; and, 4) We compare the diarization accuracies of the two APIs to inform user decision to choose the best possible diarization API for dialogue analysis applications.

Hence, this research presents a data-driven study of diarization accuracies provided by the most popular ASR APIs: Google Cloud Speech [10] and Amazon Transcribe [11]. The accuracy of these APIs in labeling the distinguished speakers in the audio feed is evaluated.

CHAPTER 2

Literature survey

2.1. Speaker Diarization

Speaker diarization [8] is a process in which the speaker's voice is automatically recognized and labeled. It improves our understanding of automatic speech recognition, which is used in downstream applications such as analytics for call-center transcription. Diarization is done through mapping. It is mapping a speech segment, such as a word, into a space representing the speaker's specific characteristics that cluster the segments together. Mapping should be done by identifying different speakers mapped according to different positions in the space, irrespective of their conversation. Neural networks are recently being used for mapping, improving diarization accuracy.

Speaker diarization can be used in Telemedicine, Conference calls, Podcast hosting, hiring platforms, Video casting, Broadcast media [12] [13] for analyzing and deriving semantic value. For example, customers in the Application Tracking System depend heavily on phone and video calls to recruit their applicants. Speaker Diarization helps separate the recruiter's and applicant's transcripts without listening to their audio and video feed.

2.2. Survey of commercially available ASR service with Speaker diarization capabilities:

2.2.1. Google Speech-to-text:

Google speech to text is a cloud-based automatic transcription API from Google that supports 289 language variants. With options for adding custom vocabulary, the ASR supports real-time streaming speech to text along with punctuations in its beta version. Transcription service

for recorded audio supports FLAC, WAV, OGG, AMR, MP3, where MP3 support is restricted to beta implementation. Custom speech transcription models can be added to the google speech-to-text API for specific tasks. Currently, the API supports phone calls, video, voice commands. Developers can utilize the Google Cloud Speech ASR API in java, python, node.js, ruby, go, PHP, and c# languages [14].

Speaker diarization is one of the features of API. This feature in its beta form supports real-time and batch transcriptions, allowing users to either upload audio files from the local computer or cloud storage. Audio files are restricted to one minute when uploaded from the local computer. Google Cloud Speech ASR can transcribe up to 4 hours of audio and video files. The Speaker diarization feature of Google cloud speech ASR supports 37 languages, and the default number of speakers for the Google ASR speaker diarization is 6.

2.2.2. Amazon transcribe:

Amazon transcribe is available in 37 languages. It supports features like speech-to-text, custom vocabulary filtering, and auto-punctuation for streaming and recorded audio (WAV, MP3, FLAC, and MP4 media formats). Software Development Kits [15] are available in java, python, node.js, ruby, go, PHP, and .NET languages [16] for developers.

Speaker diarization, all 37 languages are supported with certain limitations, i.e., batch transcriptions for media files of duration up to 4 hours. Amazon Transcribes speaker diarization can label up to 2-5 different speakers in real-time and batch transcriptions, with batch transcriptions limited to 4 hours for all 37 languages. However, the speaker diarization feature for

real-time transcription is only limited to the “English-US” language. The accuracy of diarization decreases with an increase in the number of speakers.

2.2.3. Rev.ai:

Rev.ai is available in 31 languages [17]. It supports features like Automatic Speaker Recognition, Punctuation & Capitalization, Speaker Diarization, and Time Stamp Generation for recorded media (all major audio and video types like WAV, MP3, OGG, e,t,c.)[18]. The best results are obtained with file formats such as FLAC, ALAC, MP3 with a bit rate of 192kbps or above [18]. The recorded media files should be of size (<5 GB) [18], or else they should be compressed to match the size requirements. Software Development Kits [15] are available in java, python, node.js languages [19] for developers.

Rev.ai speaker diarization is available as a default feature for recorded media files transcription and can label all the speakers in the media file.

2.2.4. IBM Watson Speech to Text:

IBM Watson Speech to Text is available in 13 language variants. It supports features like pre-trained speech models, Model training, Fine-tuning, low latency transcription, audio diagnostics before transcribing, interim transcription, smart formatting, speaker diarization, and word spotting and filtering for both streaming and recorded audio (All popular formats like FLAC, MP3, OGG, etc.). For the best transcription of recorded files, mp3, MPEG, WAV, FLAC, opus file formats [20] are recommended. Software Development Kits [15] are available in Android, Go, Java, Node.js, Python, Ruby, .NET, Swift, Unity [21].

The Speaker diarization feature of IBM Speech to text is available for telephony models for now. The speaker diarization performance will be poor if the audio feed contains more than six speakers [22].

CHAPTER 3

System design and implementation:

The primary purpose of this research is to evaluate the accuracy of speaker diarization systems in both real-time streaming and batch transcription conditions for the commercially available Speaker diarization APIs. Amazon Transcribe and Google speech-to-text[23] APIs transcribe the previously recorded 11 Motivational Interviewing roleplay sessions.

This section presents a comprehensive description of the test-bench used for speaker-diarization accuracy evaluation.

3.1. Block Diagram Representing Speaker Diarization Evaluation set-up:

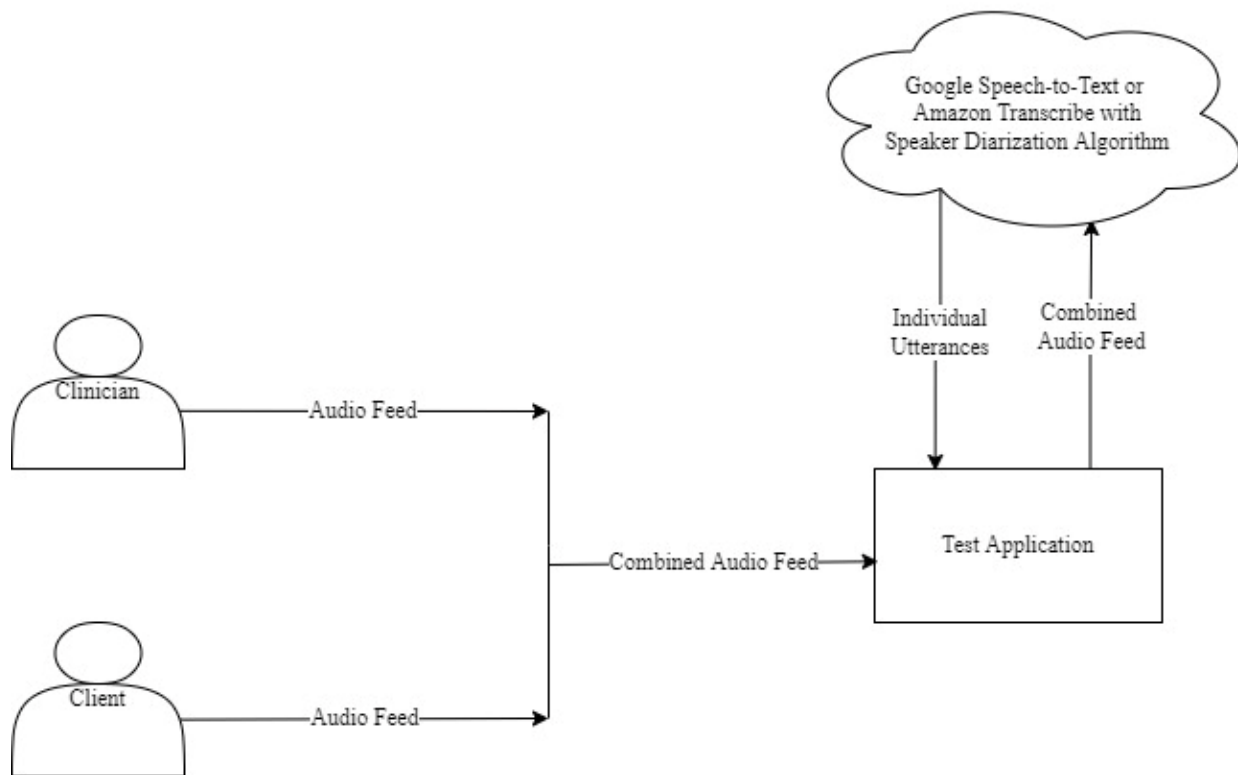


Figure 3: Evaluation set up for speaker diarization

Figure 3 represents the block diagram for the evaluation setup for speaker diarization analysis with MI-based roleplay data. According to the block diagram, a test application with a speaker diarization client is installed on the participating clinician and client's mobile device (PC, tablet, or mobile phone). The device captures combined audio feed from both the clinician and the client. The combined audio feed is then sent to the Google Cloud Speech ASR or Amazon Transcribe server for transcription and diarization. The test application receives the transcribed and speaker tagged utterances from the server is received by the test application where the speaker tagged utterances are displayed in Figure [7].

3.2. Google Speech-to-text Speaker Diarization:

Step 1: Initial setup

Google “speech to text” can be accessed by rest API or speech-to-text console. For the sake of convenience and future purpose use, we used the REST APIs for these experiments. The sequence of steps for accessing google speech to text is mentioned below.

- 1) Create an account and sign into the Google Cloud console.
- 2) Go to the manager resources page and select create a project by entering the project’s name and by linking billing details to the project.
- 3) In the search products and resources bar, search and select Google speech-to-text API from the list of available resources.
- 4) A new service account is created with basic IAM roles, and it is linked to google speech to text API.
- 5) We now must create and download the JSON key for the service account.

- 6) In our local machine, we must add a new environmental variable `GOOGLE_APPLICATION_CREDENTIALS` and assign the path of the JSON file to this variable.

Finally, we are all set to access google speech-to-text from our machine.

Step 2: API configuration for Google Cloud Speech ASR

a) Online Speaker Diarization configuration:

To enable speaker diarization, we must set the 'EnableSpeakerDiarization' to 'true' in the 'SpeakerDiarizationConfig' parameters. We can also set the minimum and the maximum number of speakers in the 'SpeakerDiarizationConfig' according to the number of speakers in our audio feed to get better diarization results in online speaker diarization, as shown below:

```
SpeakerDiarizationConfig speakerDiarizationConfig =  
SpeakerDiarizationConfig.newBuilder()  
.setEnableSpeakerDiarization(true)  
.setMinSpeakerCount(2)  
.setMaxSpeakerCount(2)  
.build();
```

Figure 4: Snippet showing google speech-to-text speaker diarization configuration

Figure 4 represents the 'speakerDiarizationConfig' parameters in which 'EnableSpeakerDiarization' is set to 'true,' and 'MinSpeakerCount' and 'MaxSpeakerCount' parameters are set to '2' to get better results as our roleplay sessions have only two speakers.

'recognitionConfig' provides information for the recognizer for processing the request. We must specify the necessary feature from the available google speech-to-text features into the 'recognitionConfig.' As our primary focus is on the speaker diarization, 'recognitionConfig' is enabled with the speaker diarization feature as shown below.

```
RecognitionConfig recognitionConfig =
RecognitionConfig.newBuilder()
.setEncoding(RecognitionConfig.AudioEncoding.LINEAR16)
.setLanguageCode("en-US")
.setSampleRateHertz(16000)
.setDiarizationConfig(speakerDiarizationConfig)
.build();
```

Figure 5: Snippet showing recognitionConfig parameters.

Figure 5 represents the 'recognitionConfig' parameters in which the language is enabled to 'English-US' and Sampling Rate Hertz is set to '16000' along with speaker diarization configuration.

b) Batch speaker diarization configuration:

In batch transcription, we must assign the 'enable_speaker_diarization' parameter to 'true' to enable the speaker diarization feature of google cloud speech-to-text API. The audio files used for our experiments are from the medical student's roleplay sessions. These files have only two speakers. So the minimum and the maximum number of speakers in the audio files are restricted to 2 using the 'min_speaker_count' and 'max_speaker_count' parameters. 'config' provides information for the recognizer to process the request. Language is enabled to 'English-US,' and Sampling Rate Hertz is set to '8000' along with speaker diarization configuration in 'config' as shown in Figure 6.


```
diarization_config = speech.SpeakerDiarizationConfig(
    enable_speaker_diarization=True,
    min_speaker_count=2,
    max_speaker_count=2,
)

config = speech.RecognitionConfig(
    encoding=speech.RecognitionConfig.AudioEncoding.LINEAR16,
    sample_rate_hertz=8000,
    language_code="en-US",
    diarization_config=diarization_config,
)
```

Figure 6: parameter configuration for speaker diarization in batch transcription

Step 3: Testbench environments:

a) Online speaker diarization:

The combined audio feed is sent as a request to the google cloud speech-to-text API every 60 seconds. Google cloud speech-to-text will assign speaker labels to every word for the request and return the results. For each request till the end of the session, google cloud speech-to-text will send the results back to the testbench environment. The final utterances, and the speaker labels are generated based on the last response received from google cloud speech-to-text. Figure 7 represents the testbench environment for getting the speaker diarization results for a session. The final utterances with speaker assignment can also be seen in the terminal highlighted with a red bubble.

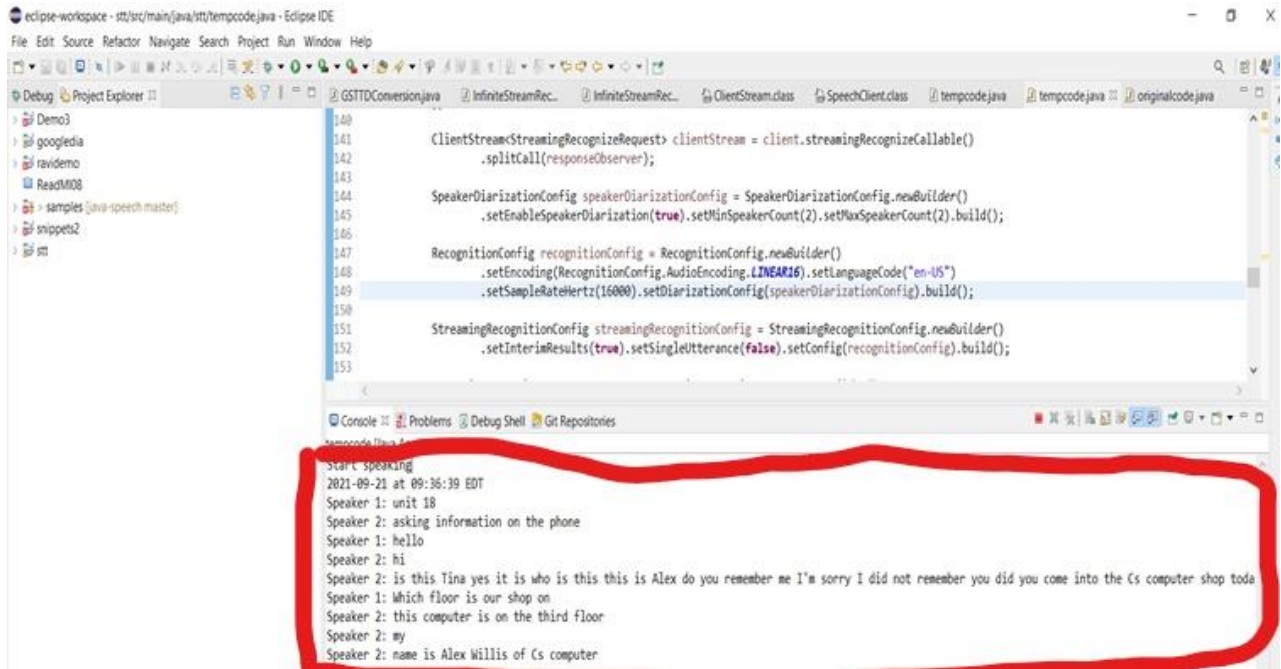


Figure 7: Test environment for google cloud speech-to-text speaker diarization transcripts

b) Batch speaker diarization:

Google ASR batch transcription can be achieved by uploading a file from the local machine or the cloud. As Google ASR supports audio files within only one minute when uploaded from the local machine, we must use Google Cloud Platform buckets to store the audio files. A storage bucket with the necessary permission for accessing the files in the bucket has been created. The audio files are uploaded manually into the Google cloud storage bucket from the local machine to avoid latency issues. Figure 8 represents the files stored in the google cloud platform bucket.

The screenshot shows the Google Cloud Platform interface for a bucket named 'bucket'. The bucket is located in 'us (multiple regions in United States)' and has a 'Standard' storage class. It is subject to object ACLs and has no retention policy. The bucket is currently empty of folders but contains several audio files. The files are listed in a table with columns for Name, Size, Type, Created, Storage class, Last modified, Public access, Version history, Encryption, and Retention expiration.

Name	Size	Type	Created	Storage class	Last modified	Public access	Version history	Encryption	Retention expiration
10.wav	91.4 MB	audio/wav	Sep 12, 20...	Standard	Sep 12, 20...	Public to internet	Copy URL	Google-managed key	-
11.wav	74.4 MB	audio/wav	Sep 12, 20...	Standard	Sep 12, 20...	Public to internet	Copy URL	Google-managed key	-
2.wav	80.8 MB	audio/wav	Sep 12, 20...	Standard	Sep 12, 20...	Public to internet	Copy URL	Google-managed key	-
4.wav	88.5 MB	audio/wav	Sep 12, 20...	Standard	Sep 12, 20...	Public to internet	Copy URL	Google-managed key	-
5.wav	77.4 MB	audio/wav	Sep 12, 20...	Standard	Sep 12, 20...	Public to internet	Copy URL	Google-managed key	-
6.wav	95.1 MB	audio/wav	Sep 12, 20...	Standard	Sep 12, 20...	Public to internet	Copy URL	Google-managed key	-
7.wav	105.8 MB	audio/wav	Sep 12, 20...	Standard	Sep 12, 20...	Public to internet	Copy URL	Google-managed key	-
8.wav	72.3 MB	audio/wav	Sep 12, 20...	Standard	Sep 12, 20...	Public to internet	Copy URL	Google-managed key	-
9.wav	69.8 MB	audio/wav	Sep 12, 20...	Standard	Sep 12, 20...	Public to internet	Copy URL	Google-managed key	-
test.wav	92.6 MB	audio/wav	Sep 7, 202...	Standard	Sep 7, 202...	Public to internet	Copy URL	Google-managed key	-
wave-male-11th.wav	15.1 MB	audio/wav	Oct 6, 202...	Standard	Oct 6, 202...	Public to internet	Copy URL	Google-managed key	-

Figure 8: Google cloud storage bucket with audio files for transcription

We must provide a link for the specific audio file in the google cloud storage bucket from the testbench environment setup to get the diarization transcripts. Figure 9 represents the testbench environment set up for google cloud speech speaker diarization with batch transcription. The diarization results with individual speaker labels are highlighted in the red bubble, as shown below.

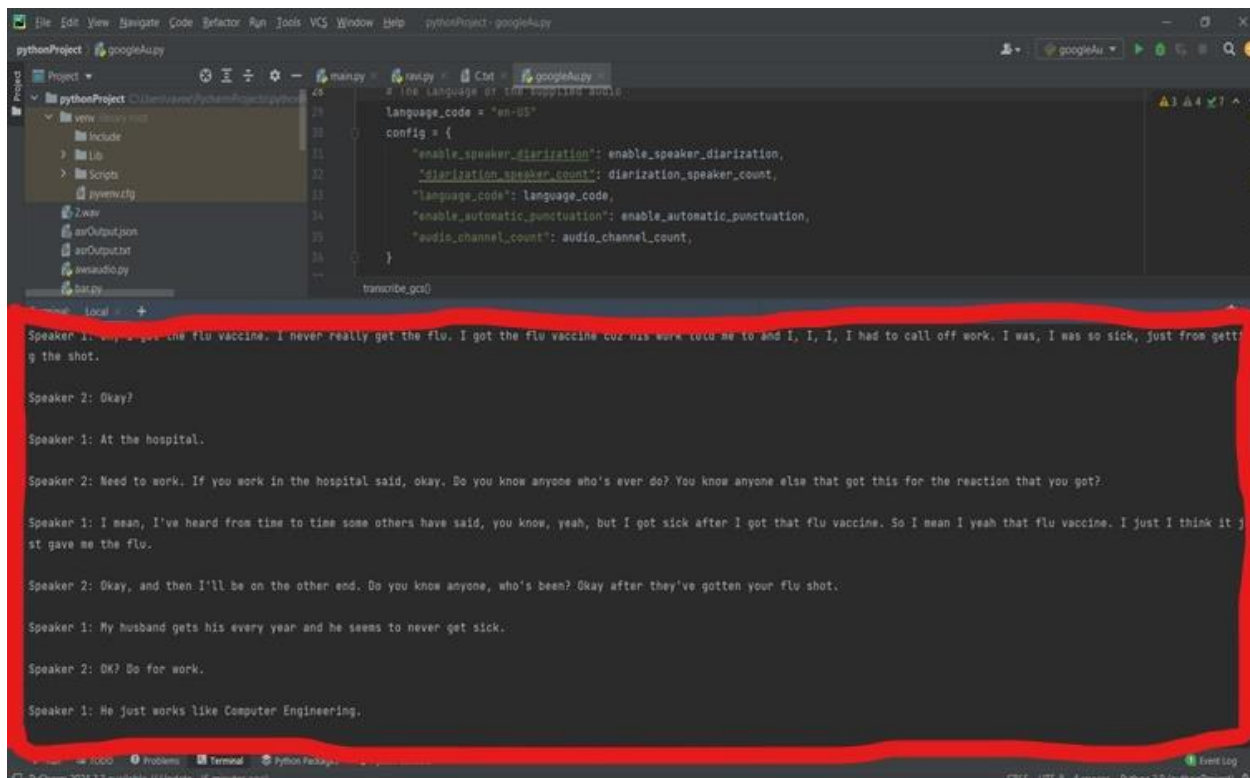


Figure 9: Screenshot showing Google ASR speaker diarization batch transcription

3.3. Amazon Transcribe Speaker Diarization:

Step 1: Initial setup

Amazon Transcribe can be accessed by rest API or speech-to-text console. For the sake of convenience and future purpose use, we used the REST APIs for these experiments. The sequence of steps for accessing Amazon Transcribe is mentioned below.

- 1) We must create an account for the amazon management console.
- 2) We have to create a key-pair, and the private key file is downloaded and stored in the '. Aws' folder in our local machine to access the EC2 instance from our local machine, and the public key is stored in our instance on the cloud.

- 3) We must go to the IAM console after logging into the amazon management console, and a role must be created, and EC2 is chosen for the use cases.
- 4) Under permissions, we must set the policy name as “AmazonTranscribeFullAccess.” The default trust relationship policy for the role must be edited.

We are all set to access google speech-to-text from our machine.

Step 2: API configuration for Amazon Transcribe

The below code snippet represents Streaming Transcription Request to Amazon transcribe with media encoding format as ‘PCM.’ Language is set to ‘English-US’. It supports only one language for streaming speaker diarization. The Sampling rate is set to 16000Hz, and speaker diarization is enabled.

```
StartStreamTranscriptionRequest request =
    StartStreamTranscriptionRequest.builder()
        .languageCode(LanguageCode.EN_US.toString())
        .mediaEncoding(MediaEncoding.PCM)
        .mediaSampleRateHertz(16_000)
        .showSpeakerLabel(true)
        .build();
```

Figure 10: Snippet showing Amazon transcribe request with speaker diarization feature

Step 3: Testbench environments:

a) Online speaker diarization:

The below code snippet shows the user interface for the amazon transcribe real-time streaming transcription, an open-source code obtained from the GitHub repository of amazon. Once all the requests are sent, the transcription with the final utterances is saved to the folder in the local machine.

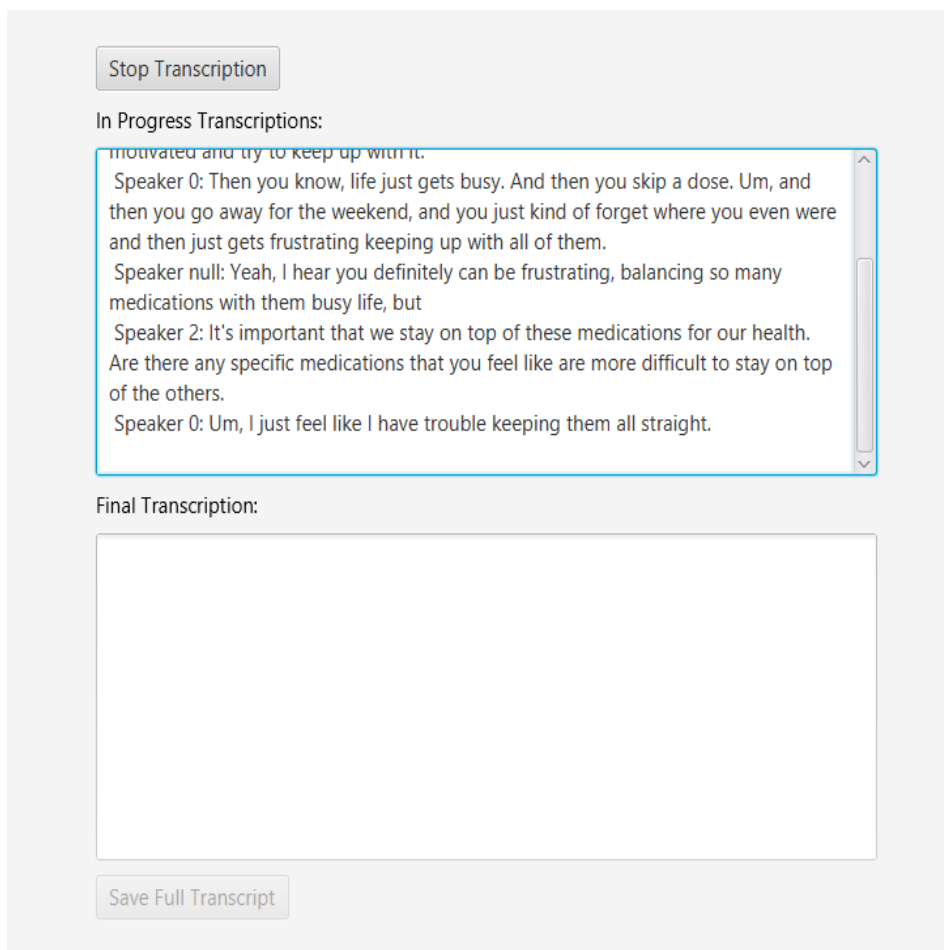
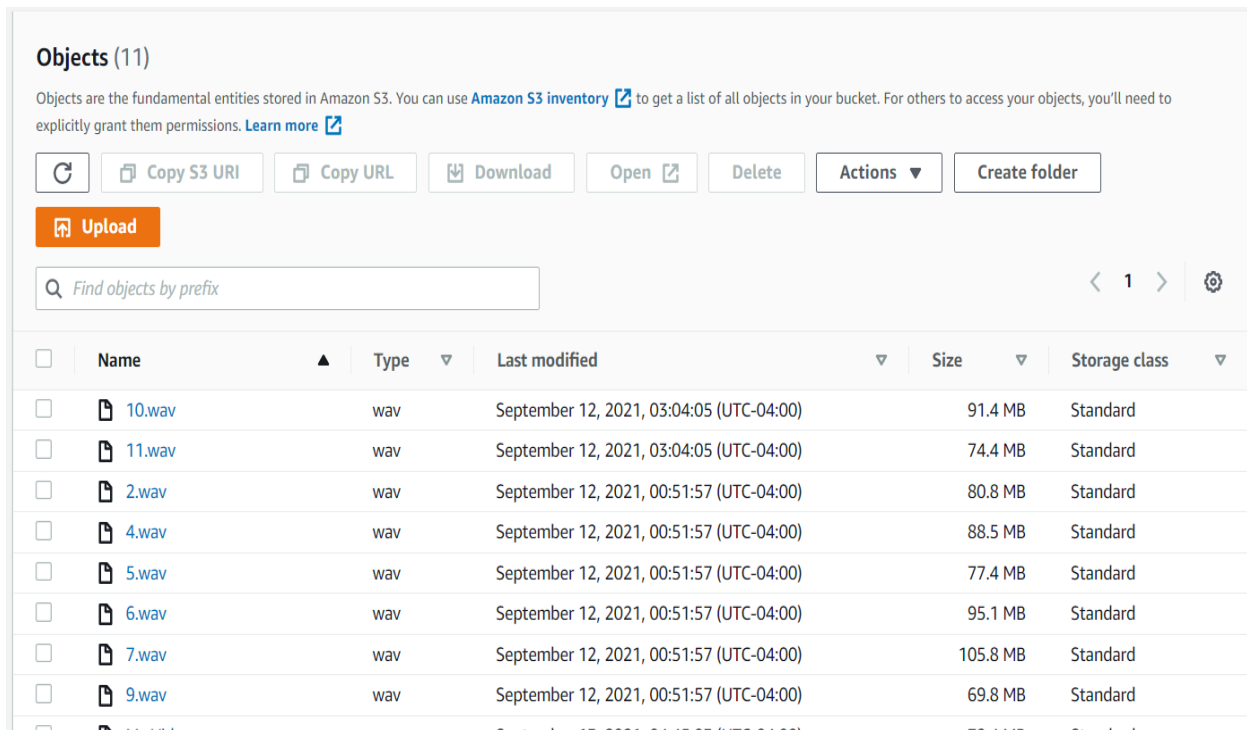


Figure 11: User interface of Amazon streaming speaker diarization

b) Batch speaker diarization:

For the batch speaker diarization, the audio files are uploaded to the s3 bucket [24] in the cloud, as shown in figure 12. Amazon management console provided an interface for getting the amazon transcribe diarization results as a JSON file, as shown in figure 13. We have to provide the URL link of the specific audio file stored in the S3 bucket and the number of speakers in the audio file to get accurate results. The JSON file obtained from the Amazon Transcribe management console is processed in our local machine to get the final transcription results with each speaker utterances labeled as shown in figure 14.



Objects (11)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#)

[Upload](#)

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	10.wav	wav	September 12, 2021, 03:04:05 (UTC-04:00)	91.4 MB	Standard
<input type="checkbox"/>	11.wav	wav	September 12, 2021, 03:04:05 (UTC-04:00)	74.4 MB	Standard
<input type="checkbox"/>	2.wav	wav	September 12, 2021, 00:51:57 (UTC-04:00)	80.8 MB	Standard
<input type="checkbox"/>	4.wav	wav	September 12, 2021, 00:51:57 (UTC-04:00)	88.5 MB	Standard
<input type="checkbox"/>	5.wav	wav	September 12, 2021, 00:51:57 (UTC-04:00)	77.4 MB	Standard
<input type="checkbox"/>	6.wav	wav	September 12, 2021, 00:51:57 (UTC-04:00)	95.1 MB	Standard
<input type="checkbox"/>	7.wav	wav	September 12, 2021, 00:51:57 (UTC-04:00)	105.8 MB	Standard
<input type="checkbox"/>	9.wav	wav	September 12, 2021, 00:51:57 (UTC-04:00)	69.8 MB	Standard

Figure 12: Amazon S3 bucket with audio files for transcription

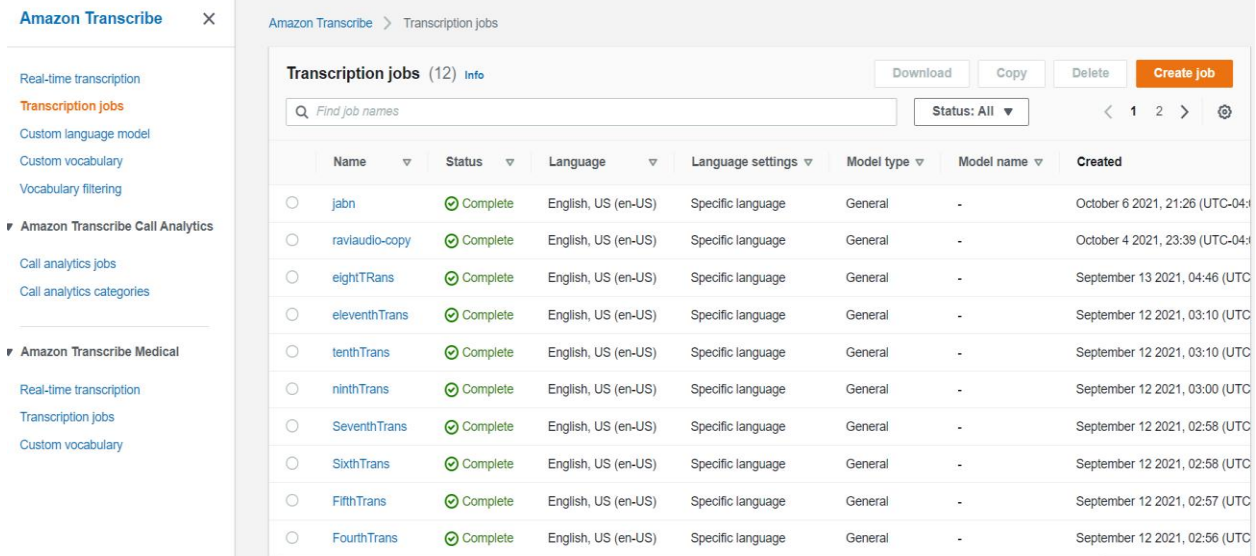


Figure 13: Amazon management console’s amazon transcribe interface

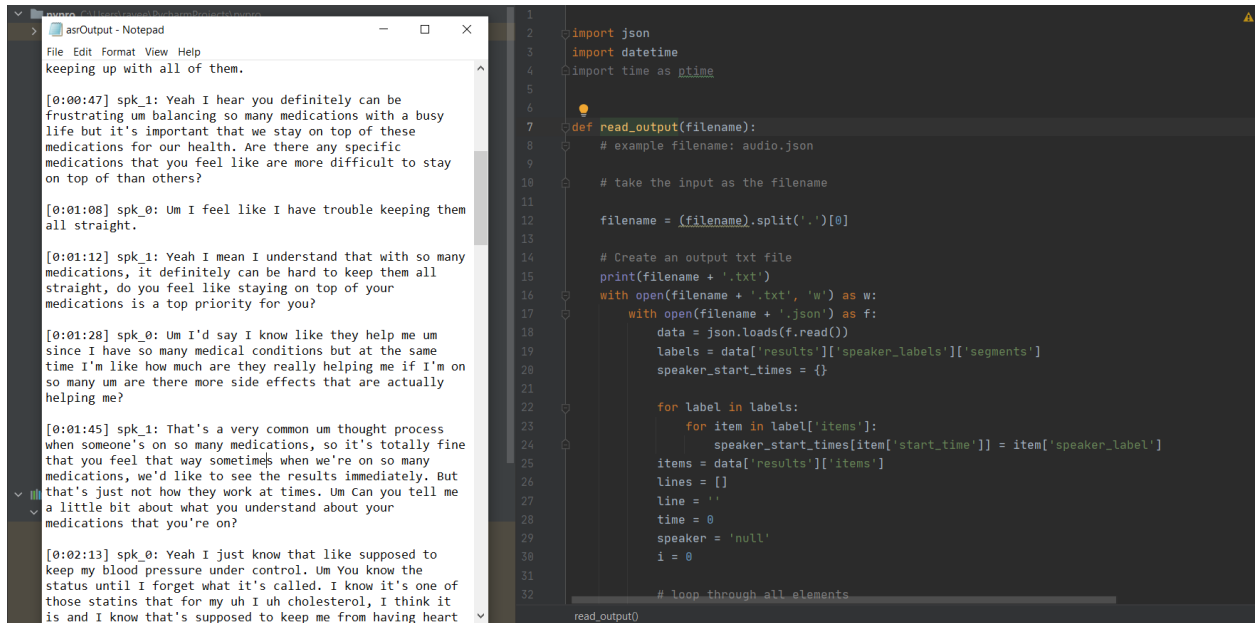


Figure 14: Test environment for processing .json file for final transcripts with speaker labelling

CHAPTER 4

Experimental Results:

This chapter compares the speech diarization capabilities of Amazon Transcribe and Google Cloud Speech ASR and presents the results as bar plots. Clinician-client audio files and corresponding speech-to-text files from ReadMI [6] represent datasets for this comparison study. This significance of using dialogue datasets can be attributed to, 1) Crosstalk errors: The existing ReadMI speech to text datasets are comprised of texts with crosstalk errors which provide a way to establish ground-truth 2) Turn-taking: ReadMI dialogue datasets are structured by equal turns [25] of clinician and client utterances which provides for a rigorous evaluation of speaker detection for each utterance.

We first present the effects of crosstalk in the existing ReadMI setup where each speaker is assigned a separate audio channel for a sample of $N=9$ roleplay sessions consisting of a total of 915 utterances. Subsequently, we present the mean speaker detection accuracy measure results of the speaker diarization feature, applied to mitigate the deleterious effects of the crosstalk from the Google Cloud Speech API and Amazon Transcribe for streaming and recorded audio. Lastly, we compare the mean speaker detection accuracy measure between Google Cloud Speech API and Amazon Transcribe.

4.1. Examine effects of Crosstalk in ReadMI:

In this experiment, 9 MI roleplay sessions (915 utterances), previously transcribed using ReadMI, are utilized. These nine transcripts are manually coded to identify utterances affected by crosstalk. Out of 915 utterances, 294 are affected by a cross-talk with an average accuracy of 67.8% for speaker separation, while 32.2% of utterances are affected by crosstalk. Using two sets

of tablet/microphones, the cross talk may significantly interfere with the speech separation of the dialogues and lead to incorrect MI evaluations.

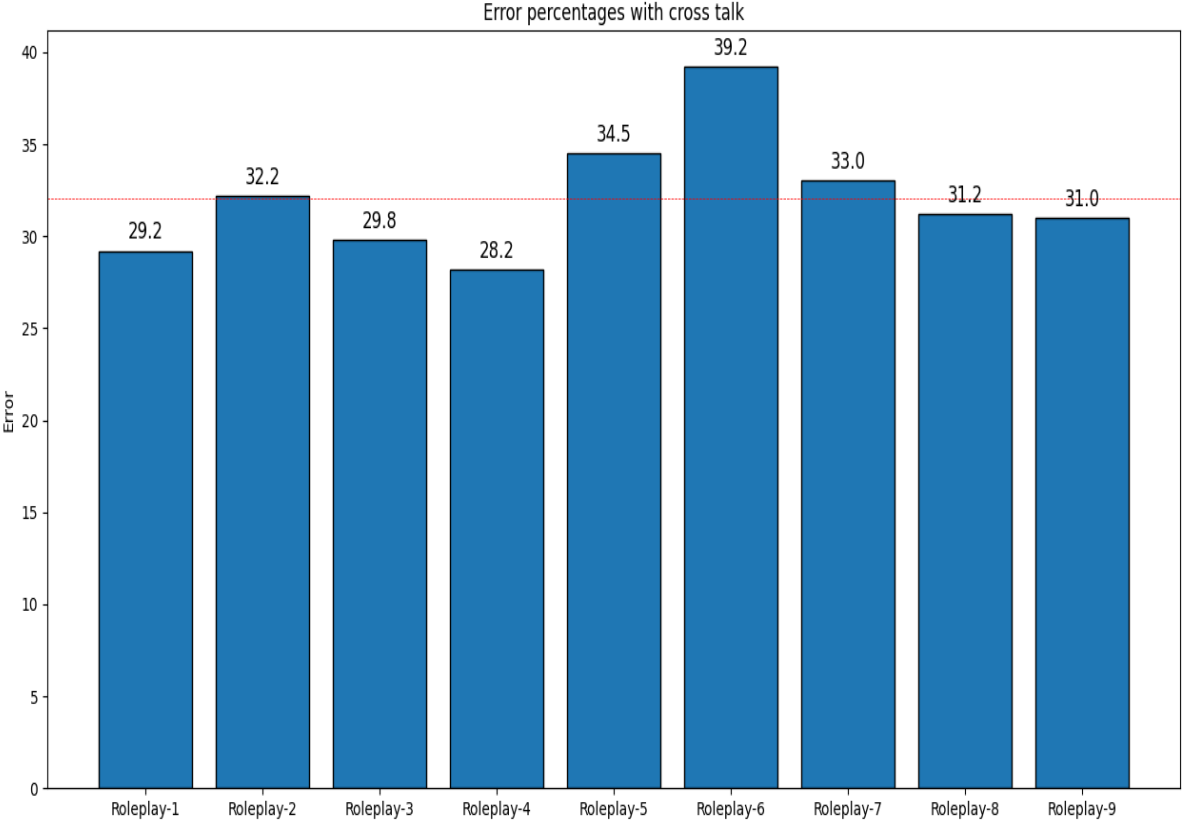


Figure 15: Percentage of utterances effected with cross talk

4.2. Accuracy of transcripts with speaker diarization feature enabled for Google and Amazon ASR in real-time streaming mode:

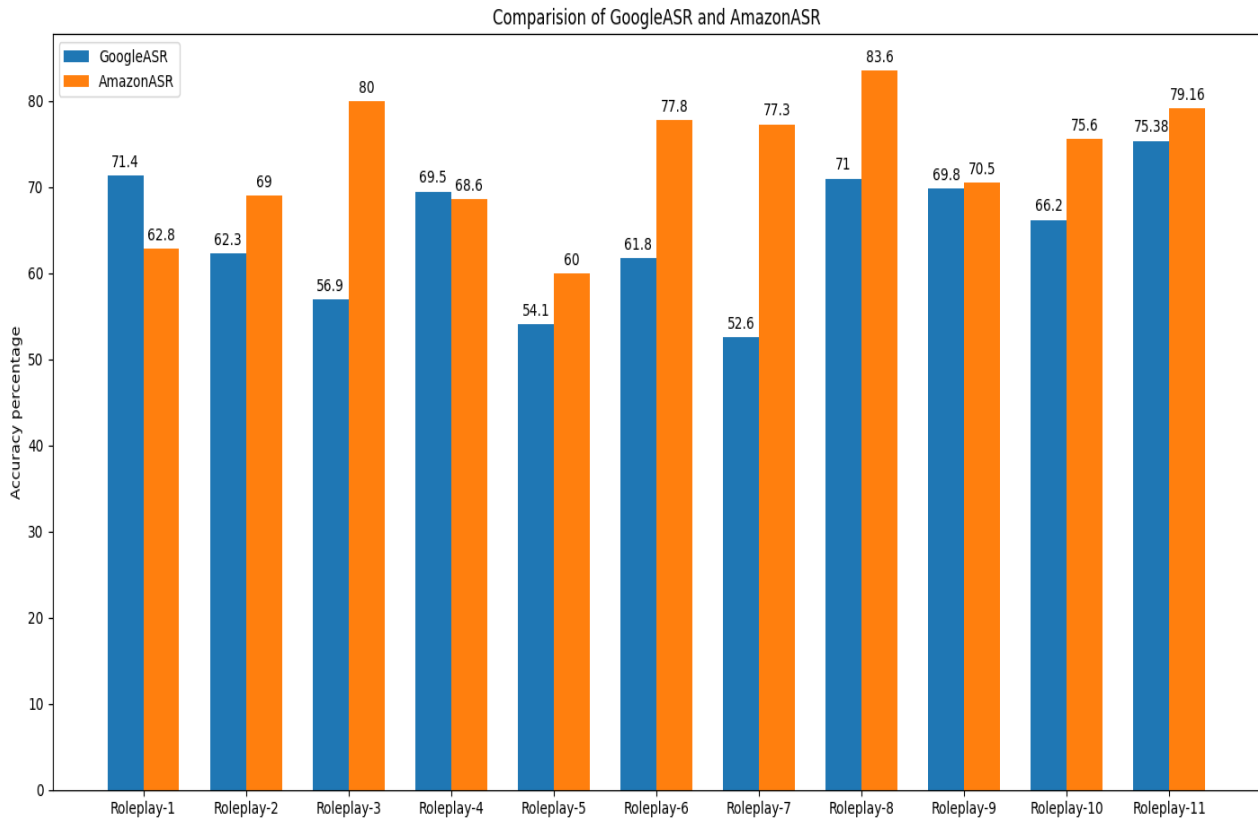


Figure 16: Real-time streaming speaker diarization amazon vs google

In the second step of our analysis, the speaker diarization feature in both the google cloud speech ASR and amazon transcribe are enabled, and the ASR is running in real-time, which is the case during a live training session. N=11 dialogue files were transcribed in real-time with speaker identification. Figure 16. represents the speaker identification accuracy of both Google Cloud Speech API and Amazon Transcribe. Google Cloud Speech API accounts for a mean accuracy of 66.36%. In comparison, a mean accuracy of 71.65% can be attributed to the Amazon transcribe, an improvement of 5.3% over the Google Cloud Speech API.

Please note no utterance was affected by crosstalk due to multiple devices anymore, as only one set of tablets/microphones was used to capture speech from both speakers. However, mislabeling of the speaker may lead to incorrect MI evaluations.

4.3. Speaker diarization accuracy measure for Google Cloud Speech API and Amazon

Transcribe in offline batch text transcription mode:

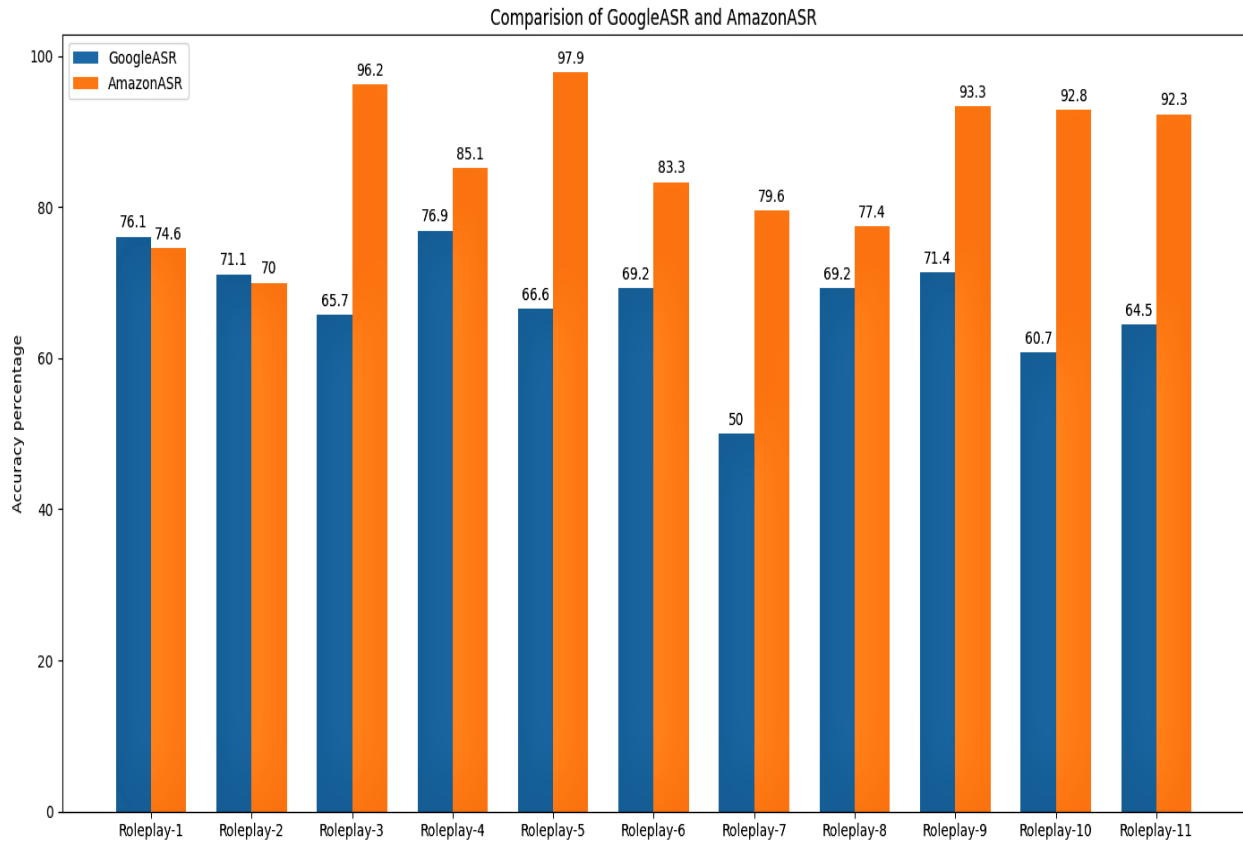


Figure 17: Accuracy percentages of batch transcription in Google and Amazon

Figure 17 represents speaker diarization accuracy percentages of batch text files obtained from uploaded audio files to Google and Amazon cloud platforms. Here the ASR is not done in real-time as the session progresses. Instead, it is completed after the session is concluded and the entire speech is made available to the ASR services. A total of N=11 previously recorded roleplay sessions were considered. A mean accuracy measure of 59.2% can be reported for the Google

Cloud Speech API. On the other hand, Amazon transcribe attributed to a mean accuracy measure of 85.1%, which is 25.9% higher than the mean speech diarization accuracy measure of Google Cloud Speech API.

Speech data API	<i>Online (Real-time)</i> <i>(% diarization accuracy)</i>	<i>Offline (Batch)</i> <i>(% diarization accuracy)</i>
Google Speech-to-text	66.36	59.22
Amazon Transcribe	71.35	85.11

Table 1: Real-time vs batch speaker diarization accuracy percentages for google cloud speech ASR and Amazon Transcribe

4.4. Gender based roleplay accuracies:

There has been a lot of ongoing research going on gender-based voice recognition. Some research proved that male voice recognition is better than female voice recognition [26]. We also extended our research into that area with preliminary results as mentioned below.

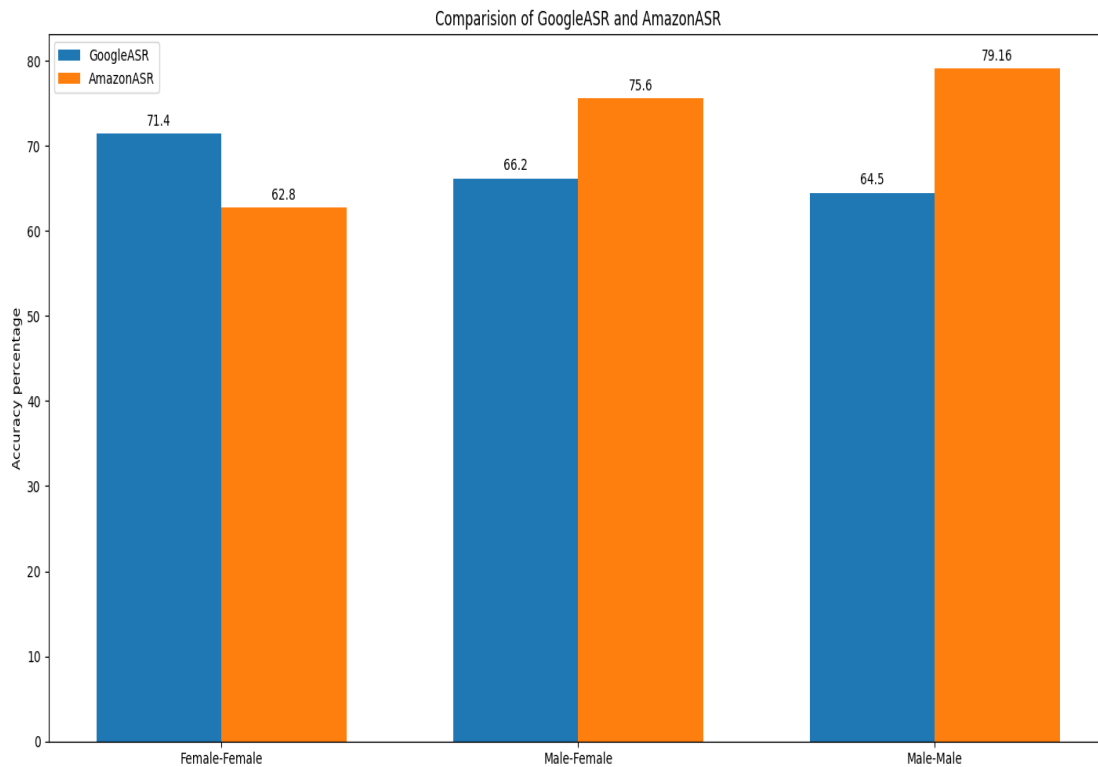


Figure 18: Gender specific roleplays for streaming speaker diarization

The above figure represents three roleplays from the previously experimented real-time streaming speaker diarization 11 roleplays. As we can see from the figure that there is no significant difference in the speaker diarization systems performance in distinguishing each speaker in all three categories: female-female, male-female, and male-male.

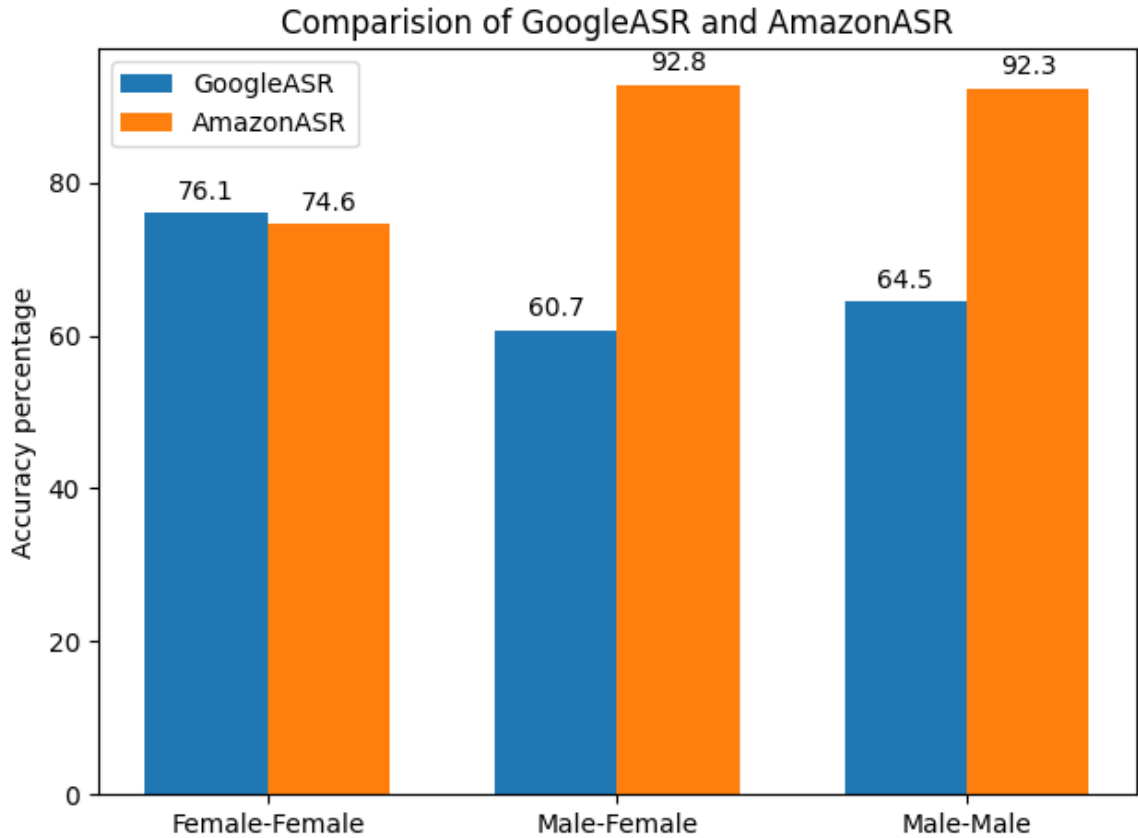


Figure 19: Gender specific speaker diarization for batch transcription

The above figure represents accuracy percentages in identifying different speakers of three roleplay sessions by batch transcription (by uploading audio files). It can be seen that Amazon Transcribe gave much more improved accuracy in all cases: female-female, male-female, and male-male roleplay sessions in batch transcription mode compared to its real-time streaming mode. The performance for male-female and male-male roleplay sessions reaches over 90%. On the other side, Google Cloud Speech didn't show similar improvement in the batch transcription mode compared to its real-time streaming mode.

4.5. Discussion:

As mentioned in section 4.1, the utterances are affected by the issue of cross-talk. The current ReadMI system individually captures and transmits audio on two separate channels to the Google cloud ASR. It has been demonstrated that by using the Speaker diarization feature, this multi-channel audio from both participants can be transmitted on a single channel to the Google Cloud ASR. The ASR separates the individual audio and labels each speaker in the audio feed.

From Table-1, it is demonstrably clear that Amazon Transcribe performs better than Google cloud speech-to-text, indicating a higher level of maturity of Speaker diarization capabilities in Amazon Transcribe. This could have alluded to the “beta” implementation of Google cloud speech ASR. A beta implementation of a software feature is typically characterized by incomplete and/or inaccurate implementation. At the time of experimentation, only the beta version of the Google Cloud Speech ASR was available. Therefore, we may have to wait for the “alpha” version of the Google Cloud Speech ASR for a comprehensive comparison. Amongst the nature of the speech data, i.e., streaming versus batch transcription, Amazon Transcribe shows improved accuracy for batch transcriptions versus real-time streaming transcription (Table 1). This improvement can be attributed to additional information from audio and text sequences embedded in the stored file. The future, present, and past knowledge of the sequence are available for decision-making. On the other hand, in the Google Cloud Speech API context, the online (real-time) streaming diarization outperforms the offline (batch) diarization. This change in result can be attributed to 1) The beta version of the Google Cloud Speech API implementation and its lack of optimization in the batch transcription mode; 2) The need for additional data samples for a more comprehensive examination.

CHAPTER 5

5.1. Conclusion:

This thesis research is focused mainly on avoiding the crosstalk issue, which is an active field of study in speech-to-text for transcribing individual utterances [9][27]. We take advantage of the latest advances of the Automatic Speech Recognition systems in speaker diarization to overcome the issue of crosstalk. As speaker diarization is an emerging technology in Speech-to-Text, unfortunately, I didn't find any surveys available comparing this feature between different Speech Recognition Systems. Thus, this thesis intended to fill this gap and establish the performance baseline of adopting speaker diarization for speech separations for supporting performance assessment of MI skills in role-play training/learning sessions.

This thesis has made contributions in the following aspects:

- It contributed to evaluating the percentage of utterances affected by crosstalk when using multi-channels.
- It provides a brief overview of the available features of the topmost automatic speech recognition systems, along with their speaker diarization capabilities.
- It contributes to the successful set up of the testbench environments for assessing the speaker diarization capabilities from Google Speech-to-Text and Amazon Transcribe in both online and batch transcriptions.
- It contributes to evaluating the accuracy of speaker diarization capabilities of both online and batch transcriptions in the two topmost available speech recognition systems.
- It contributes to evaluating the speaker diarization capability accuracy based on the participant's gender.

In conclusion, the current speaker diarization capabilities of Google and Amazon ASRs alone might not produce sufficiently accurate speech separation for our particular use case in integrating this feature into our ReadMI application. A combination of speaker diarization, microphone array, and/or adaptive microphone control technologies may be necessary to achieve the desired performance. But this thesis research potentially provides a clear baseline reference for future developers to integrate speaker diarization capabilities into similar applications.

5.2. Future work:

Future work might involve the use of diarization on a training set up with two sets of tablet/microphones, which will examine the performance of a combined approach that uses the diarization on top of the hardware-based speech separation. Furthermore, collecting more audio files and evaluating the diarization capabilities on a more significant amount of data is warranted to get more comprehensive evaluation results.

REFERENCES

- 1) Rollnick, S., & Miller, W. R. (1995). What is motivational interviewing? *Behavioural and cognitive Psychotherapy*, 23(4), 325-334.
- 2) Noonan, W. C., & Moyers, T. B. (1997). Motivational interviewing. *Journal of Substance Misuse*, 2(1), 8-16.
- 3) Miller, W. R., & Rollnick, S. (2002). *Motivational interviewing: Preparing people for change*. Book Review.
- 4) Joyner, B., & Young, L. (2006). Teaching medical students using role play: twelve tips for successful role plays. *Medical teacher*, 28(3), 225-229.
- 5) Xiao, B., Can, D., Georgiou, P. G., Atkins, D., & Narayanan, S. S. (2012, December). Analyzing the language of therapist empathy in motivational interview-based psychotherapy. In *Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference* (pp. 1-4). IEEE.
- 6) Vasoya MM, Shivakumar A, Pappu S, Murphy CP, Pei Y, Bricker DA, Wilson JF, Castle A, Hershberger PJ. ReadMI: An Innovative App to Support Training in Motivational Interviewing. *J Grad Med Educ*. 2019 Jun;11(3):344-346. doi: 10.4300/JGME-D-18-00839.1. PMID: 31210875; PMCID: PMC6570458.
- 7) Chris Huff, "Understanding Crosstalk and How To Eliminate It", <https://www.behindthemixer.com/understanding-crosstalk-and-how-eliminate-it/>. Accessed on November 15, 2021.
- 8) X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland and O. Vinyals, "Speaker Diarization: A Review of Recent Research," in *IEEE Transactions on Audio, Speech, and*

Language Processing, vol. 20, no. 2, pp. 356-370, Feb. 2012, doi:
10.1109/TASL.2011.2125954.

- 9) S. N. Wrigley, G. J. Brown, V. Wan and S. Renals, "Speech and crosstalk detection in multichannel audio," in IEEE Transactions on Speech and Audio Processing, vol. 13, no. 1, pp. 84-91, Jan. 2005, doi: 10.1109/TSA.2004.838531.
- 10) Google Cloud Speech-to-Text, <https://cloud.google.com/speech-to-text> . Accessed on November 15, 2021.
- 11) AWS Transcribe Streaming Example Java Application, <https://github.com/aws-samples/aws-transcribe-streaming-example-java> . Accessed on November 15, 2021.
- 12) Hernawan, S. (2012, September). Speaker Diarization: Its Developments, Applications, And Challenges. In proceedings intl conf information system business competitiveness.
- 13) Joe Zaghoul, "Speaker Diarization: Speaker Channels for Mono Channel Files", <https://www.assemblyai.com/blog/speaker-diarization-speaker-labels-for-mono-channel-files/> . Accessed on November 15, 2021.
- 14) Google Cloud Client Libraries, <https://cloud.google.com/speech-to-text/docs/client-libraries> . Accessed on November 15, 2021.
- 15) Willenbring, J. M. (2019). Software Development Kits: A Software Integration Strategy for CSE (No. SAND2019-1978C). Sandia National Lab. (SNL-NM), Albuquerque, NM (United States).
- 16) What programming languages does Amazon Transcribe support?
<https://aws.amazon.com/transcribe/faqs/> . Accessed on November 15, 2021.

- 17) Ajita Mishra, “What Languages Does Rev.ai Support”,
<https://help.rev.ai/en/articles/2620995-what-languages-does-rev-ai-support> . Accessed on November 15, 2021.
- 18) Cayla Walkerson, “Accepted File Types and Sizes for Transcription”
<https://support.rev.com/hc/en-us/articles/360035077992-Accepted-File-Types-and-Sizes-for-Transcription> . Accessed on November 15, 2021.
- 19) Rev.ai API Overview(v1), <https://www.rev.ai/docs/overview#section/SDKs> . Accessed on November 15, 2021.
- 20) Transcribe Audio, <https://speech-to-text-demo.ng.bluemix.net/> . Accessed on November 15, 2021.
- 21) Watson SDKs, <https://cloud.ibm.com/docs/speech-to-text?topic=watson-using-sdks> . Accessed on November 15, 2021.
- 22) Speaker Labels, <https://cloud.ibm.com/docs/speech-to-text?topic=speech-to-text-speaker-labels#speaker-labels-ids> . Accessed on November 15, 2021.
- 23) Detect different speakers in an Audio Recording, <https://cloud.google.com/speech-to-text/docs/multiple-voices> . Accessed on November 15, 2021.
- 24) Amazon S3, <https://aws.amazon.com/s3/> . Accessed on November 15, 2021.
- 25) Stivers, T., Enfield, N. J., Brown, P., Englert, C., Hayashi, M., Heinemann, T., ... & Levinson, S. C. (2009). Universals and cultural variation in turn-taking in conversation. *Proceedings of the National Academy of Sciences*, 106(26), 10587-10592.
- 26) Bajorek, J. P. (2019). Voice recognition still has significant race and gender biases. *Harvard Business Review*.

27) Mic crosstalk on live interview (two microphones), <https://gearspace.com/board/post-production-forum/1201824-mic-crosstalk-live-interview-two-microphones.html> .

Accessed on November 15, 2021.