PROJECT: SPEECH TO TEXT IN A MEETING SETTING WITH PERSON IDENTIFICATION

Project Scope:

1. Record the meeting discussion voice
2. Convert it to text
3. Recognise the speaking person's voice (identify the speaker)
4. Assuming speakers talk in turn; if not, the assistant shall prompt for speakers to take turns
5. When the recognition confidence is low, prompt the speaker to repeat or confirm what was recorded

---

Data (Record Audio)

Process Flow:

Convert to text

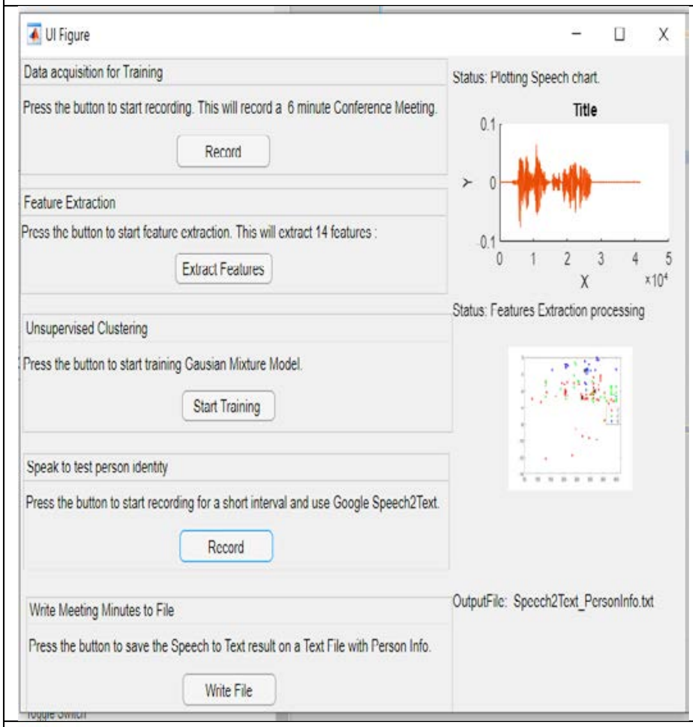
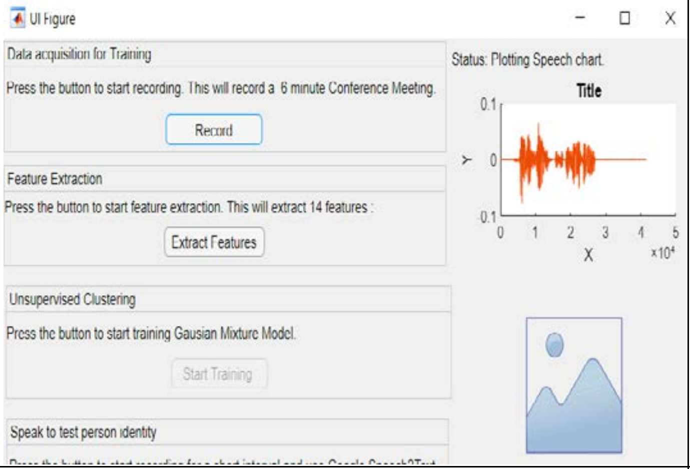
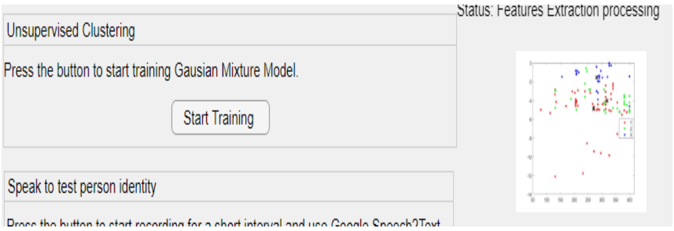
- 6 minutes Clustering
    - Time sampling
    - Data Sampling Feature Extraction
    - Unsupervised Trained Model
- start Listening
- Identify Cluster number (e.g. cluster 3)
- IDENTIFY OVERLAP
    - Identified Clusters more than 1 – in a given time
    - Identification score low
        - Ask to repeat

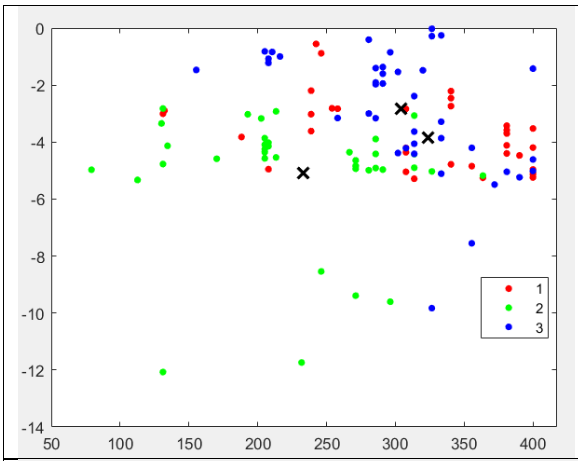
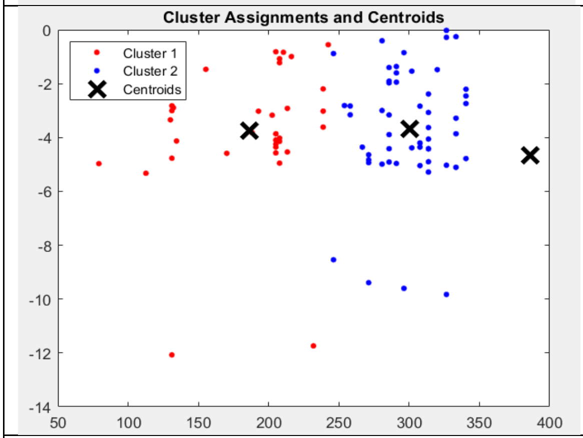
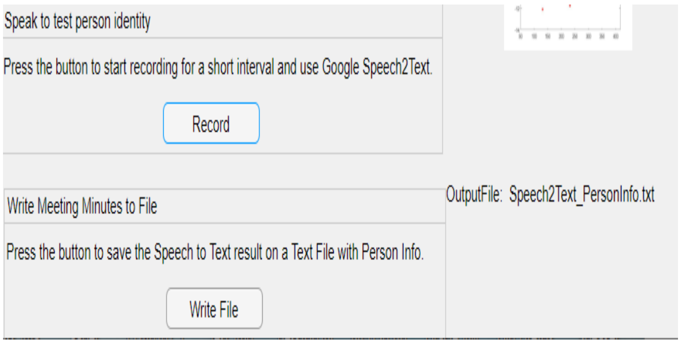
Algorithm I

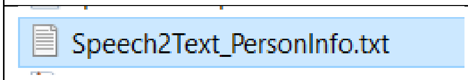
1. Record Audio
2. Extract Features
3. Train an unsupervised model
4. Record Audio
    a. Extract features
    b. Predict speaker
    c. Convert speech to text
        i. For a poor detection, repeat speech
            1. Repeat 4.
        ii. For a good detection, write to file [Person Name, Text]
            1. Repeat 4.

<u>The Speech to Text and Person Identification App:</u>

This is the app interface

Interface	Function	Challenges
 UI Figure Data acquisition for Training Press the button to start recording. This will record a 6 minute Conference Meeting. Feature Extraction Press the button to start feature extraction. This will extract 14 features : Unsupervised Clustering Press the button to start training Gausian Mixture Model. Speak to test person identity Press the button to start recording for a short interval and use Google Speech2Text. Write Meeting Minutes to File Press the button to save the Speech to Text result on a Text File with Person Info. Status: Plotting Speech chart. Status: Features Extraction processing OutputFile: Speech2Text_PersonInfo.txt Record Extract Features Start Training Record Write File Title 0.1 0 -0.1 0 1 2 3 4 5 X x10⁴	This is the interface of the App. It has following sub-units: **1.Data Acquisition** for Training the network in the start. **2. Feature extraction** from the initial Data **3.** Unsupervised **Clustering** for detection likelihood regions of similarities among data. **4.** Audio record for short intervals and **continuous /real time prediction** with the trained model. **5.** Using Google API to **convert speech** intervals **to text**.	Package to .exe issue Matlab compiler was missing.
 UI Figure Data acquisition for Training Press the button to start recording. This will record a 6 minute Conference Meeting. Feature Extraction Press the button to start feature extraction. This will extract 14 features : Unsupervised Clustering Press the button to start training Gausian Mixture Model. Speak to test person identity Status: Plotting Speech chart. Record Extract Features Start Training Title 0.1 0 -0.1 0 1 2 3 4 5 X x10⁴	1. Record audio for at least 6 minutes in the start of the meeting. Figure shows speech signal of 5 seconds (for illustration).	At the earlier period in the meeting, If people can introduce themselves, 'Hi, I am ABC', this info can be used for annotation purposes. Because the data is not annotated by the speaker's name, it is not possible to identify each person - name, for instance.
 Status: Features Extraction processing Unsupervised Clustering Press the button to start training Gausian Mixture Model. Speak to test person identity 	3. Once the features are extracted, a near neighbour model can be trained. The features extracted are Pitch, log Energy and MFCC coefficients). They are 14 features in total.	Lack of access to dataset in this short time limited the research. Sample data was acquired from a YouTube channel and the video was based on an office meeting setup. The speech signal was manually annotated for testing the system's efficiency.

<table>
<tr>
<td> </td>
<td>Result of Gausian mixture Model with 3 cluster centres. With a little data of 6 minutes, the system could be trained into 3 clusters.
The clustering is distinguishing the regions vaguely.</td>
<td>Lack of samples for training resulted in weak clustering results. Larger dataset and a better segmentation technique for speech signals can improve the results.</td>
</tr>
<tr>
<td> </td>
<td>Result of KMEANS clustering. Data points are shown in color, centroids marked in cross.</td>
<td></td>
</tr>
<tr>
<td>Speak to test person identity
Press the button to start recording for a short interval and use Google Speech2Text.

OutputFile: Speech2Text_PersonInfo.txt
Write Meeting Minutes to File
Press the button to save the Speech to Text result on a Text File with Person Info.
 </td>
<td>4. Once the model is trained, the system is ready to be integrated to predict the person in terms of which cluster the voice of the person belongs to. As the system is partly trained, the more it gets exposed to similar voice patterns, the better it will become at recognition and distinguishable clustering, as the neural weights are already trained to a curtained point.</td>
<td>To Identify an individual speaker, their names have to be mentioned as data markers. Supervised machine learnings usually perform better for the similar reason.
A time series component could help synchronise speech identification at a better real-time pace.</td>
</tr>
<tr>
<td>Speech2Text_PersonInfo.txt</td>
<td>While the speech is continuously send to Google API for converting to Text, the output of the API shows two fields: text and confidence level. If the confidence level is low, the person can be asked to repeat themselves.

This implementation is done with Automated Meeting Minutes scenario in mind.</td>
<td>GoogleAPI stopped working for some reason. This is what is content of the text File:
<table>
<tr><th>Person Number</th><th>status</th><th>message</th></tr>
<tr><td>2</td><td>NoResult</td><td>Speech API did not return any Results. Try changing API options</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>
</td>
</tr>
</table>

Observations:

GoogleAPI has a huge resource of recognizable speech. It can be further explored for better applications and problem solutions.

The duration of speech when the Speech-to-Text function is called is not robust. It is fixed to give the GoogleAPI time to match the current phrase. However, in real-time, the duration of phrases vary, and can't be limited to a short sentence, such as in a conference. Sampling rate of the signals for people who speak different dialects of a language could also vary.

A network design for speech identification must be robust and not limited to a fixed number of people in a meeting, for instance. There is a lot of room for classification techniques for robust applications, such as , for a meeting which grows in participants without prior notice.

Overlapping speech will result in poor detection score and can be corrected. This can begets the machine to be well trained for identification.

Running the Code in Matlab:

Follow the sequence for running different stages of the flow:

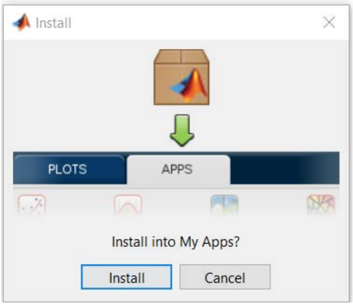
1. file8_speech2textWithGoogle
2. file8_1_speech2feats.m
3. file8_som.m
4. file9_test2retrainAndIdentify.m
5. file10_write2file.m

To Install the package in Matlab:

Run

app1.mlapp

Followed by



Install into My Apps?

============ End of Document ===============