Project Name:
Minutes Of Meeting (MOM) Recorder
(Speech to Text Convertor)

Project Initiators & Partners

- Braa Mohamed Salim Kakah (*ai.kaizenuae@gmail.com*)
- Mohamed Ahmed Al Nuaimi (*mohamedalnuaimi.1978@gmail.com*)

Table of Contents

Introduction

This project aims to create an application that allows the user to convert any conversation speech to organized text. Such application will assist to record meeting minutes during the discussion is going on and happen without missing any points of the discussion during the meeting.

The application will help in record all participants discussion in the meeting and recognize the source individually. At the end, the application will convert the recorded speech to organized text that can be presented in any publishable format.

To achieve the goals of the application objectives to convert speech to text from different voices, the following overall steps were considered:

1. Application should record the discussion during a meeting and convert it into sequence minutes.
2. The application shall recognize different voice sources and assigned them accordingly to user name as it happened live (ex. Person 1, Person 2, Person 3… etc.)
3. Organize the converted "Text" to a provided template of Minutes Of Meeting (MOM) in MS Word format ".docx or .doc".
    i. Phase I will focus only to use English as language of the meeting.
    ii. Phase II possibility to add additional languages (United Nation formal Languages).
    iii. Phase III introduce interruption of different languages during the discussion.

Accordingly, a list of actions and tasks were defined to ensure achieving the objectives as required and resolve any challenges that face the different programming phases. Mentioned list or tasks were summarized as following

points:

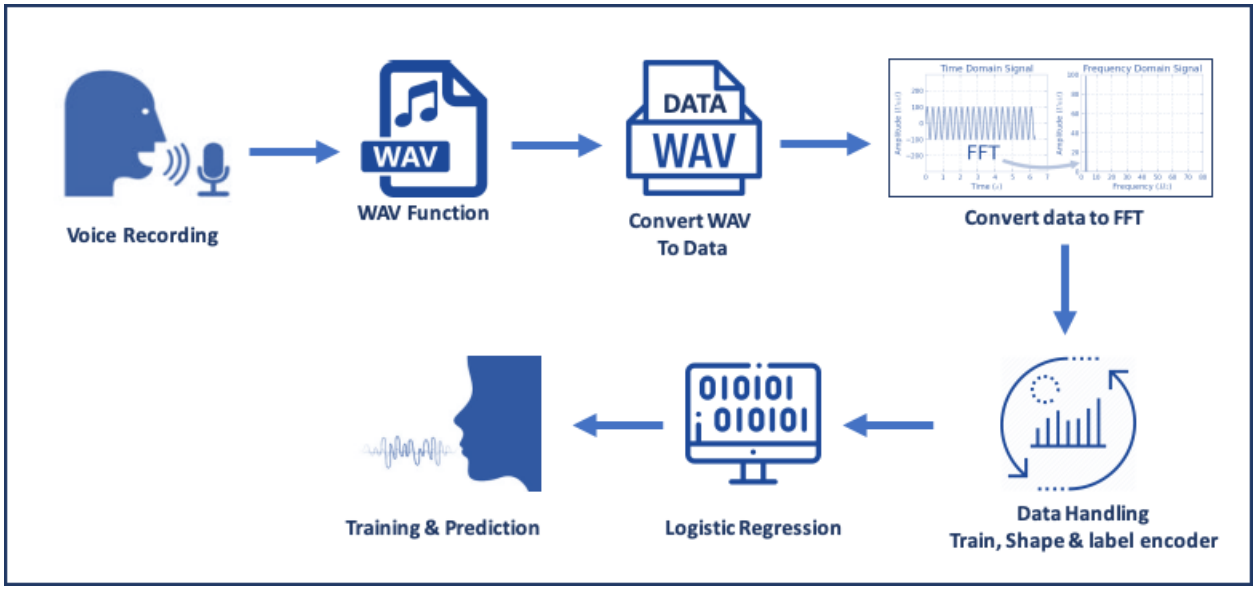
1. Search of any similar application that ever published to gain the lessons learned.
2. List of required Python's libraries that required to code or program.
3. Record "Test" & "Train" voice in WAV format for any public article.
4. Train the application to recognize different voice sources and assigned them accordingly to user name as it happened live (ex. Person 1, Person 2, Person 3… etc.).
5. Converted text should be in accurate sequence.
6. Store converted "Text" to a provided template of Minutes Of Meeting (MOM) in MS Word format ".docx or .doc".
7. Using Languages:
    i. Phase I will focus only to use English as language of the meeting.
    ii. Phase II possibility to add additional languages (United Nation Formal Languages).
    iii. Phase III introduce recording of interruption of different languages during the meeting.

That application will be suitable and useful to be used in wide range of business domain which will assist to accelerate publishing the Minutes of Meeting (MOM) accurate without missing any discussed points and in short time. Not Only this, but it will help also converting any dialogue during investigation in police stations or trial sessions or even in human resources sessions.

It was a challenge to complete the project in such limited period of time; however, at this moment this application need to be continue developed and completed with described phases to be useful for business domain.

Project Flow Chat

the following flow chart diagram illustrates the project workflow to achieve the final product of the coding as required.



Coding Phase Explanation

In order to train the code, the following five paragraphs were recorded in WAV format and accordingly the recorded voice from two developers; Braa and Mohamed; were pass to the code:

1. “A massive coral reef garden the size of 60 football stadiums is to be created in the UAE - and is set to drive up eco tourism while safeguarding vital marine life.”

2. “Work is under way on the Fujairah Cultured Coral Reef Gardens, the largest project of its kind in the country, which will include the cultivation of 1.5 million coral reef colonies over the next five years and will span 300,000 square metres, helping to provide a boost to food security efforts.”

3. “The ambitious project was launched by the Ministry of Climate Change and Environment, in partnership with Fujairah Municipality, Dibba Fujairah Municipality and Fujairah Adventures and under the patronage of Sheikh

Mohammed bin Hamad bin Mohammed Al Sharqi, Crown Prince of Fujairah, on Saturday."

4. "The preservation of the country's biodiversity is also a prime area of focus for the UAE leadership today as outlined in the UAE Vision 2021. The Ministry of Climate Change and Environment has put in place multiple programs that aim to carry out the country's national strategy for biodiversity in collaboration with other local environmental authorities in the country."

5. " Ministry is keen on promoting the project as an ecotourism destination. It is also anticipated to encourage the spirit of volunteerism and community work as the cultivation of the coral reefs will depend heavily on the volunteering efforts of the youth."

   Reference: https://www.thenational.ae/uae/environment/major-coral-reef-garden-project-launched-in-fujairah-1.848464

Those recorded voices were saved and labels as following:

- BTEST1.wav
- BTEST2wav
- BTEST3.wav
- BTEST4.wav
- BTEST5.wav
- MTEST1.wav
- MTEST2wav
- MTEST3.wav
- MTEST4.wav
- MTEST5.wav

Where "B" stands for Person 1 (Braa), and "M" stands for Person 2 (Mohamed).

In the following detail description of application code:

1) Training sound files were saved in AI_project folder, and tested files were saved in Test folder. Mkdir command was used to create a new folder and mv command used to move sound files to the relative correct folder.

```
#mkdir AI_project Test
```

```
#mv BTEST1.wav BTEST2.wav BTEST3.wav BTEST4.wav BTEST5.wav AI_project
```

2) Importing the required libraries. Firstly, Numpy library to handle and manipulate data. Then wave library to deal with sound files and convert them to data, then os library (operation system) to handle files and folders. Matplotlib library to show graphs and plot figures. Finally Wave file command to read the sound file and return the wave representation in time domain and the sample rate of the wave. Fft command to do Fast Fourier Transform and fftfreq to return Discrete Fourier Transform.

```
import numpy as np
import wave
import os
import matplotlib.pyplot as plt
from scipy.io import wavfile
from scipy.fftpack import fft,fftfreq
```

3) This code shows the loaddata function, as it reads each sound file and returns its Fourier Transform. This method was selected as that the sound print of each one is fixed and can be differentiated by training a model on Fourier Transform.

```
def loaddata (path):                          # load the data and extrat frequencies
                                              #of each sound file
  samplerate, data = wavfile.read(path)       # Return the sample rate
  samplerate                                  #(in samples/sec) and data from
  data.shape                                  #a WAV file
  samples = data.shape[0]
  samples
  datafft = fft(data)   # returns Fast Fourier Transform of the data
  fftabs = abs(datafft)  # Take absolute values
  freqs = fftfreq(samples,1/samplerate) #Return the Discrete Fourier Transform
                                        #sample frequencies.
  Frequency = []
  for i in range(len(fftabs)):
    Frequency.append(data[i][0])  # This For loop to extract the first
                                  #value of data list

  return Frequency
```

4) Creating empty lists for traindata and labels. The training files are listed in AI_project file. The listdir command will fetch all the files in this folder and save them in the variable (files). As shown in the output, there is unwanted file (checkpoint) and should be avoided in reading and appending to the training data.

```
traindata = []       # an empty list to append later
trainlb = []         # an empty list to append later
folder = 'AI_project'           # sound files are listed here
files = os.listdir(folder)    # Read files in Sounds folder
files
```

```
['.ipynb_checkpoints',
 'BTEST4.wav',
 'MTEST1.wav',
 'BTEST3.wav',
 'BTEST1.wav',
 'MTEST2.wav',
 'BTEST2.wav',
 'MTEST4.wav',
 'MTEST3.wav']
```

5) To avoid fetching unwanted files, an IF loop was used to check the files before reading. This For loop will read the directory of each sound file and send it to the loaddata function, then FFT data will be returned and saved in data variable. Then The data file will be cropped to make sure that every file will have the same length.

```
for filename in files:
    # a for loop to check files before reading,mayby there is a checkpoint file
    # appears.
    if filename[0]=='B' or filename[0]=='M':

      # send each sound file to loadfile function
      data = loaddata (folder+"//" +filename)
      # crop the sound files to min length of them which is 747859
      traindata.append(data[0:747859])
      # append Labels to trainlb either B for Braa and M for Mohammed
      trainlb.append(filename[0])
      print(filename)
```

```
BTEST4.wav
MTEST1.wav
BTEST3.wav
BTEST1.wav
MTEST2.wav
BTEST2.wav
MTEST4.wav
MTEST3.wav
```

6) This command will convert the lists to numpy arrays and do reshaping of the train label to fit in one column with multiple rows.

```
# convert traindata to a numpy array
traindata= np.array(traindata)
# reshape the lable list to one column and multible rows
trainlb = np.array(trainlb).reshape(-1,1)
```

7) Checking that the train and label lists are in the correct shape before training the model.

```
#check the type of lable list before entering the model
print(type(trainlb))
# check the shape of lable list before entering the model
print(trainlb.shape)
# check the type of train data before entering the model
print(type(traindata))
# check the shape of train data before entering the model
print(traindata.shape)
```

```
<class 'numpy.ndarray'>
(8, 1)
<class 'numpy.ndarray'>
(8, 747859)
```

8) The Label column should be encoded to zero and one before do training, then reshape again to one column and multiple rows.

```
# use Label Encoder for B and M before training
from sklearn.preprocessing import LabelEncoder
# create the LabelEncoder object
le1 = LabelEncoder()
# apply the LabelEncoder for train label data
trainlb = le1.fit_transform(trainlb).reshape(-1,1)
```

9) Using Logistic regression model to train the model and tune the parameters to find results with good prediction.

```
# Use logisticRegression model for the case
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(C=4.0,max_iter=1000000, verbose=1,penalty='l1')
```

10)Train the model with train data and labels

```
# Train the data
model.fit(traindata,trainlb)
```

11)Testing phase: This command will read the Test file and return FFT of the tested file. Then do cropping of the file and reshape with one row and multiple columns. Then pass to the prediction command.

```
# load a test file
testdata = loaddata ('Test/BTEST5.wav')
```

```
# crop the test file and send it to load data function
testdata = testdata[0:747859]
# reshape the data
testdata = np.array(testdata).reshape(1,-1)
print(type(testdata))
print(testdata.shape)
```

```
<class 'numpy.ndarray'>
(1, 747859)
```

```
# Prediction
Prediction = model.predict(testdata)
```

12) Finally, inverting the predicted label to its original label.

```
# Do inverse transform to find the predicted person
Person = le1.inverse_transform(Prediction)
if Person == 'B':
  print ("Braa Mohamad Salim Kakah")
if Person == 'M':
    print ("Mohammed AlNuaimi")
```

```
Braa Mohamad Salim Kakah
```

Difficulties and Challenges

Although the idea of the project is simple in principle; it was bit challenge in phasing and design the workflow nevertheless selecting the optimum of Python libraries and algorithm to code.
One of the main challenges that we face in this project was working in voice recognition in Time Domain; however, that been improved after applying FFT to work in Frequency Domain. Such approach enhances the outcome and give better result as it shown in the code.
Moreover, project schedule was too tight to complete the project as required and expected. In other word, the project time was challenge to deliver the project and align with the supervisor.
One of the interesting challenge or difficulty was searching for problem solving and applying new approach that was not covered in the classroom. However, as it was explained in the classroom the web pages full of articles and support blogs.
Also to come over the issue of searching for the data, the team create a solution that suite the issue by recording their voice as “Test” dataset to use in testing and learning the code. That was better solution fit to train the application to recognize the voice of two persons as start up for this application.

Future Advancements

Currently the application was coded as supervised speech recognition due to tight schedule; hopefully it will be followed to test the code in non-supervised.

Moreover apply more than international languages and become a commercial product.

That application will be suitable and useful to be used in wide range of business domain which will assist to accelerate publishing the Minutes of Meeting (MOM) accurate without missing any discussed points and in short due time. Not Only this, but it will help also converting any dialogue during investigation into police stations or trial sessions or even in human resources sessions, for example.

Resources and Links

Web pages:

- Spoken Speaker Identification based on Gaussian Mixture Models : Python Implementation (https://appliedmachinelearning.blog/2017/11/14/spoken-

speaker-identification-based-on-gaussian-mixture-models-python-implementation/)
- Loading WAV Files and Showing Frequency Response (https://pythondsp.rob-elder.com/loading-wav-files-and-showing-frequency-response/)