Shri Vile Parle Kelvani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

Approved by AICTE and Affiliated to the University of Mumbai

Department of Electronics & Telecommunication Engineering

OST MINI PROJECT REPORT

ON

VOICE BASED SERVICES

GROUP MEMBERS:

Atulya Kumar 60002160052

Crispin Lobo 60002160054

PROFESSOR INCHARGE:

PROFESSOR RANJUSHREE PAL

AIM:

To create speech controlled applications for checking mail, setting reminders and mobile services.

Software Used:

Python 3.7.1

Theory:

Basically this system is aimed at creating voice controlled applications. The user is greeted by the main menu which contains the list of all the available services. The prompt "Please speak now" indicates that the system is ready to receive the audio commands. The program derives the keywords from the input audio command and compares it with the pre-set database. The system is enable to provide three types of distinct services based on the keywords detected.

1. Checking E-mails
2. Setting Reminders/Appointments
3. Mobile services

1. Checking E-mails:

The system acts on the derived keyword 'email'. This leads to list of further option of what to check in one's E-mail. After the "Please speak now" prompt the user is to speak the select option to check the respective folders of the E-mail account. The options are:

(i) Inbox: Extract the keyword 'inbox' and keeps it for further action

(ii) Unread: Extract the keyword 'unread' and keeps it for further action

(iii) Sent: Extract the keyword 'sent' and keeps it for further action

In this prototype system the actual email account and services are not linked. However the keywords are derived to trigger these applications. So as to establish a fully operational system the E-mail account and services can be linked.

This application allow E-mail services to be accessed easily and enhance user experience.

2. Setting Reminders/Appointments:

The system acts on the derived keyword ‘reminders’. The system leads the user to a sequence of questions to acquire the details of the reminder to be set. Each question is preceded by the “Please speak now” prompt.

(i) Title of the reminder: The first question is “What is the title of the reminder? ex. Pay bill”. This is used to set the topic of the reminder or appointment.
(ii) Time: The second question is “At what time do you want to be reminded? ex. 10pm”. This used to set the time of the reminder.
(iii) Venue: “What is the place of the reminder? ex. at Home” This used to set the venue if any.

Thus this simple sequential system is used to set reminder in quick and comprehensive manner.

3. Mobile services

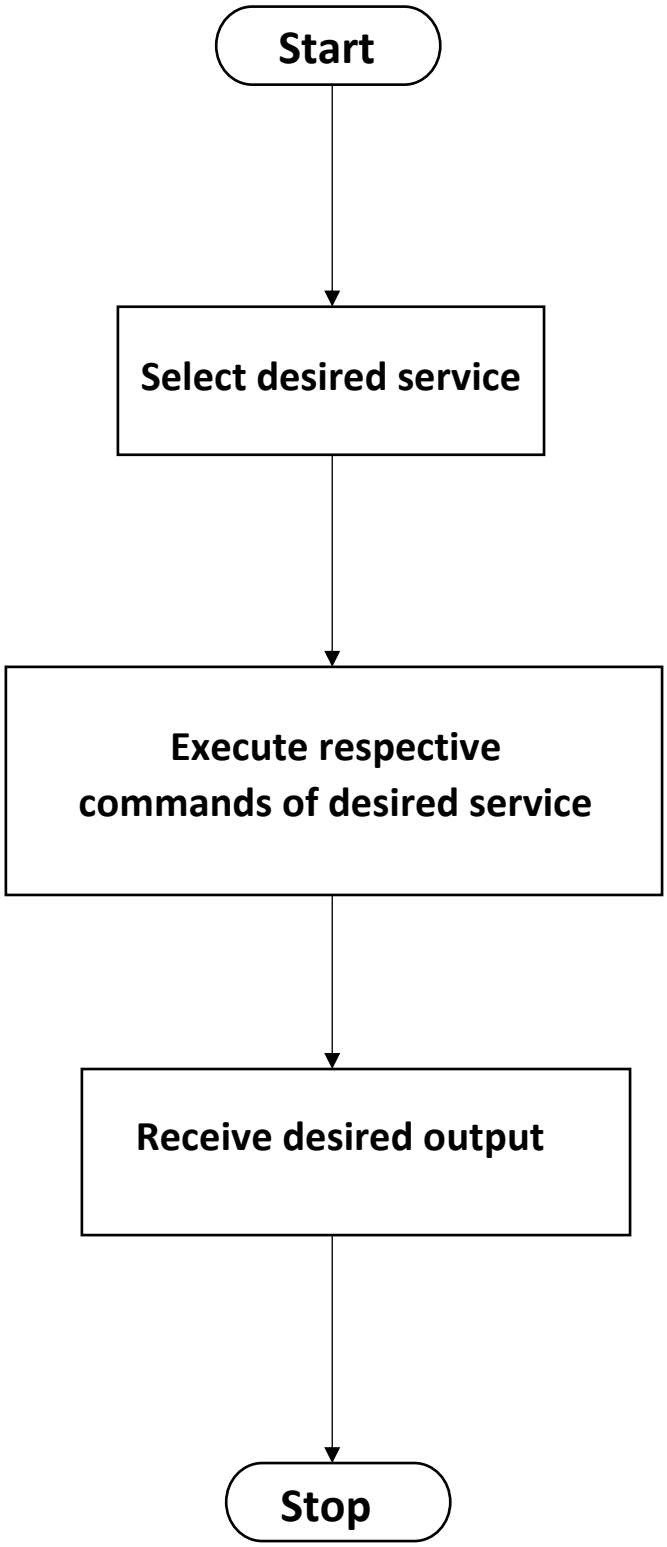
The system acts on the derived keyword ‘mobile bill’. The system displays the mobile bill for the current month. It further gives all available offers so as to choose a new plan.

Plan A - Airtel's Rs. 399 postpaid recharge plan
Plan B - Airtel's Rs. 499 postpaid recharge plan
Plan C - Airtel's Rs. 649 postpaid recharge plan
Plan D - Airtel's Rs. 799 postpaid recharge plan
Plan E - Airtel's Rs. 1,199 postpaid recharge plan

After the “Please speak now” prompt the user can chose the required plan.

This systems can be expanded to a large to incorporate various features to enhance various services.

Flowchart


Start
Select desired service
Execute respective commands of desired service
Receive desired output
Stop

CODE :

1. **Main.py**

```
from speechtotext import *

if __name__ == "__main__":

    # create recognizer and mic instances
    recognizer = sr.Recognizer()
    microphone = sr.Microphone()

    initial_keywords=None
    mail_keywords=None
    reminder_title_keywords=None
    reminder_time_keywords=None
    reminder_place_keywords=None
    plan_keywords=None

    print('Please state which facility you want to use\n1.\tCheck email\n2.\tSet
Reminder\n3.\tView Mobile Bill\n4..\tExit')
    while initial_keywords is None:
        initial_keywords=start_recording(recognizer,microphone)

    if 'email' in initial_keywords:
        print('\nSelect Email Folder\n1.\tInbox\n2.\tUnread\n3.\tSent')
        while mail_keywords is None:
            mail_keywords=start_recording(recognizer,microphone)
            if mail_keywords!=None and 'inbox' in mail_keywords:
                print('View inbox')
            elif mail_keywords!=None and 'unread' in mail_keywords:
                print('View unread emails')
            elif mail_keywords!=None and 'sent' in mail_keywords:
                print('View sent messages')
            else:
                print('Program did not understand your instruction. Please try again\n')


    elif 'reminder' in initial_keywords:
        print('\nWhat is the title of the reminder ? ex. Pay bill')
        while reminder_title_keywords is None:
            reminder_title_keywords=start_recording(recognizer,microphone)
            title=' '.join(reminder_title_keywords)
            print('\nAt what time do you want to be reminded ? ex. 10pm')
            while reminder_time_keywords is None:
                reminder_time_keywords=start_recording(recognizer,microphone)
                time=' '.join(reminder_time_keywords)
                print('\nWhat is the place of the reminder ? ex. at Home')
                while reminder_place_keywords is None:
                    reminder_place_keywords=start_recording(recognizer,microphone)
```

```
            place=' '.join(reminder_place_keywords)

        print('\nOkay! We will remind you to `{}` at `{}` `{}`'.format(title,time,place))


    elif 'mobile' in initial_keywords:
        print('''\nBill :
              One time charges              2,000.00
              Monthly charges                850.00
              Usage charges Call charges       0.00
              Value added services             0.00
              Mobile internet usage         29,473.95
              Roaming                        0.00
              Discounts                   29,473.95
              Last bill period late fee       100.00
              Taxes                        413.25
              This month's charges          3,363.25''')

        print('''\nList of Available Plans :

Plan A - Airtel's Rs. 399 postpaid recharge plan
Plan B - Airtel's Rs. 499 postpaid recharge plan
Plan C - Airtel's Rs. 649 postpaid recharge plan
Plan D - Airtel's Rs. 799 postpaid recharge plan
Plan E - Airtel's Rs. 1,199 postpaid recharge plan''')

        print('To select or change plan, say `Plan A/B/C/D/E ')
        while plan_keywords is None:
            plan_keywords=start_recording(recognizer,microphone)

        print('Your new postpaid plan is {}'.format(' '.join(plan_keywords)))


    elif 'exit' in initial_keywords:
        print('Bye !')

    else:
        initial_keywords=None
        print('Program did not understand your instruction. Please try again\n')
```

2. **speechtotext.py**

```
import random
import time
import speech_recognition as sr

def recognize_speech_from_mic(recognizer, microphone):
    """Transcribe speech from recorded from `microphone`.

    Returns a dictionary with three keys:
    "success": a boolean indicating whether or not the API request was
               successful
    "error":   `None` if no error occured, otherwise a string containing
               an error message if the API could not be reached or
               speech was unrecognizable
    "transcription": `None` if speech could not be transcribed,
               otherwise a string containing the transcribed text
    """
    # check that recognizer and microphone arguments are appropriate type
    if not isinstance(recognizer, sr.Recognizer):
        raise TypeError("`recognizer` must be `Recognizer` instance")

    if not isinstance(microphone, sr.Microphone):
        raise TypeError("`microphone` must be `Microphone` instance")

    # adjust the recognizer sensitivity to ambient noise and record audio
    # from the microphone
    with microphone as source:
        recognizer.adjust_for_ambient_noise(source)
        audio = recognizer.listen(source)

    print('Audio Recorded')

    # set up the response object
    response = {
        "success": True,
        "error": None,
        "transcription": None
    }

    # try recognizing the speech in the recording
    # if a RequestError or UnknownValueError exception is caught,
    #     update the response object accordingly

    try:
        response["transcription"] = recognizer.recognize_google(audio)
    except sr.RequestError:
        # API was unreachable or unresponsive
        response["success"] = False
```

```
        response["error"] = "API unavailable"
    except sr.UnknownValueError:
        # speech was unintelligible
        response["error"] = "Unable to recognize speech"

    return response

def start_recording(recognizer,microphone):
    try:
        print('\nPlease speak now ...')
        for i in range(1):
            text = recognize_speech_from_mic(recognizer, microphone)

            if text["transcription"]:
                # show the user the transcription
                print("You said: {}".format(text["transcription"]))
                break

            if not text["success"]:
                print("I didn't catch that. What did you say?\n")
                break

            # if there was an error
            if text["error"]:
                print("ERROR: {}".format(text["error"]))
                break


        string=text["transcription"]
        keywords=string.split()
        return keywords
    except AttributeError:
        pass
```

Output :

Main Menu

```
Please state which facility you want to use
1.      Check email
2.      Set Reminder
3.      View Mobile Bill
4..     Exit

Please speak now ...
```

Checking Email Option

```
Please speak now ...
Audio Recorded
You said: check email

Select Email Folder
1.      Inbox
2.      Unread
3.      Sent

Please speak now ...
Audio Recorded
You said: inbox
View inbox
```

Setting Reminder Option

```
Please speak now ...
Audio Recorded
You said: set reminder

What is the title of the reminder ? ex. Pay bill

Please speak now ...
Audio Recorded
You said: show project

At what time do you want to be reminded ? ex. 10pm

Please speak now ...
Audio Recorded
You said: 7 a.m.

What is the place of the reminder ? ex. at Home

Please speak now ...
Audio Recorded
You said: in college

Okay! We will remind you to `show project` at `7 a.m.` `in college`
```

Change Mobile Plan Option

```
Please speak now ...
Audio Recorded
You said: view mobile bill

Bill :
                One time charges                2,000.00
                Monthly charges                   850.00
                Usage charges Call charges          0.00
                Value added services                0.00
                Mobile internet usage          29,473.95
                Roaming                             0.00
                Discounts                      29,473.95
                Last bill period late fee         100.00
                Taxes                             413.25
                This month's charges            3,363.25

List of Available Plans :

Plan A - Airtel's Rs. 399 postpaid recharge plan
Plan B - Airtel's Rs. 499 postpaid recharge plan
Plan C - Airtel's Rs. 649 postpaid recharge plan
Plan D - Airtel's Rs. 799 postpaid recharge plan
Plan E - Airtel's Rs. 1,199 postpaid recharge plan
To select or change plan, say `Plan A/B/C/D/E

Please speak now ...
Audio Recorded
You said: plan c
Your new postpaid plan is plan c
```

Conclusion:

1. Python computer programming language is a very useful utility.
2. We have successfully implemented voice controlled application services
3. We have performed this application of voice to control various services for checking emails, setting reminders and mobile services.

References:

1. J. L. Flanagan, Speech Analysis Synthesis and Perception, New York:Springer-Verlag, 1972.
2. Python Cookbook, Third edition by David Beazley and Brian K. Jones
3. How To Think Like A Computer Scientist: Learning With Python, by Allen Downey, Jeff Elkner and Chris Meyers.