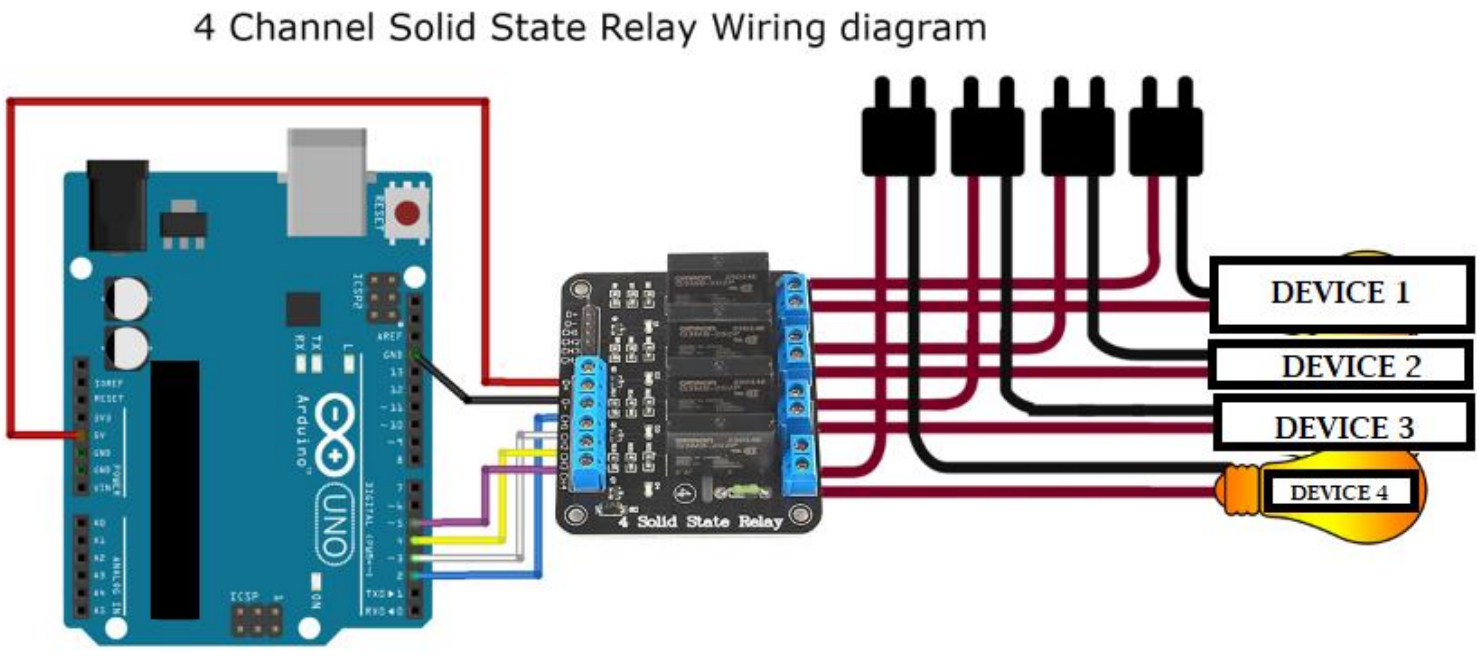
Abstract:

This project proposes the development of a voice-controlled university assistance system that uses Arduino. The system will enable users to navigate the campus, access event notifications, and obtain faculty details and campus news using voice commands. The project will leverage Arduino technology, wireless communication, and speech recognition to provide comprehensive support to the users. And is capable of operating connected devices by simple voice commands. If run on a device can help you open a file just by saying its names

INDEX

CIRCUIT - DIAGRAM



INTRODUCTION

With the increasing prevalence of smart homes and IoT devices, there is a growing demand for voice assistants that can automate and simplify various tasks within a home environment. DEXI is a voice assistant system that utilizes NLP techniques to recognize and respond to voice commands. In addition to home automation features, DEXI also includes campus navigation, faculty search, campus updates, and internet search capabilities.

BACKGROUND

Voice assistants have become increasingly popular in recent years, with devices such as Amazon Alexa, Google Home, and Apple Home Pod dominating the market. These systems utilize NLP algorithms to recognize and respond to voice commands, allowing users to control various smart home devices, play music, and perform other tasks. However, most commercially available voice assistants are limited in their functionality and do not provide advanced features such as campus navigation or faculty search.

PROBLEM DEFINITION

The project aims to address the need for an efficient and easy-to-use interface for controlling smart home devices and accessing campus-related information. Users often find it challenging to navigate through the campus or search for faculty details. Additionally, controlling multiple devices at home can be tedious and time-consuming. DEXI aims to simplify these tasks by providing an all-in-one solution for home automation and campus navigation.

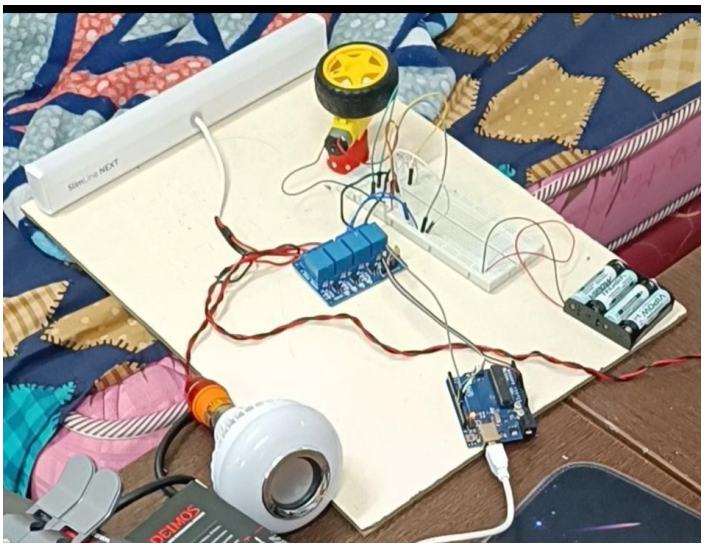
OBJECTIVES

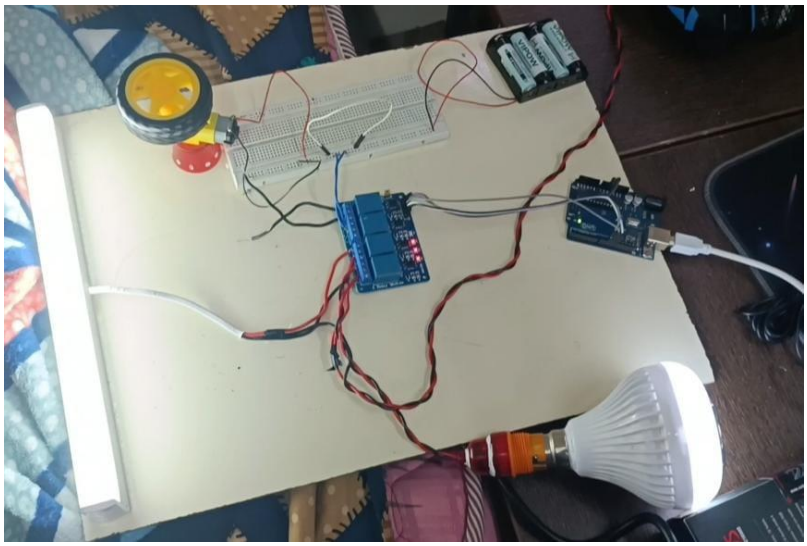
The primary objectives of the project are as follows:

- To develop a digital voice assistant for home automation and campus navigation.
- To provide an efficient and easy-to-use interface for controlling smart home devices.
- To enable users to search for faculty details, navigate through the campus, get campus updates, and search for information on the internet using their voice commands.
- To integrate various home automation devices and campus navigation systems to create a comprehensive digital voice assistant.

METHODOLOGY

The DEXI voice assistant with home automation using Arduino was developed using a combination of hardware and software. The hardware components included an Arduino board, a microphone, a speaker, and several sensors such as temperature and motion sensors. The software component was developed using Python and involved the use of various libraries such as the speech recognition library, pyaudio, and the Arduino library.The procedure involved building the hardware components and wiring them to the Arduino board. Next, the software was developed and uploaded to the board. The voice assistant was trained to recognize certain voice commands using the speech recognition library. For example, the assistant could recognize commands such as "find faculty details" or "navigate to the library."The home automation component involved controlling various devices such as lights and fans using the Arduino board and sensors. For instance, if the temperature sensor detected a high temperature, the fan would automatically turn on..





DISCUSSION

The DEXI voice assistant with home automation using Arduino was successfully developed and tested. The voice assistant was able to recognize several voice commands and perform the corresponding actions such as finding faculty details and navigating through the campus. The home automation component was also successful in controlling various devices based on the sensor readings.One limitation of the system was the accuracy of the speech recognition library, which sometimes led to misinterpretation of voice commands. However, this could be improved with further training of the voice assistant.

CONCLUSION & FUTURE SCOPE :

In conclusion, the DEXI voice assistant with home automation using Arduino is a useful system that can help users navigate through the campus and control various home appliances. Future improvements could include adding more voice commands, improving the accuracy of the speech recognition library, and integrating the system with a mobile application for remote control.

REFERENCE :

- Voice Controlled Home Automation System Using Arduino," International Journal of Innovative Research in Computer and Communication Engineering, vol. 4, no. 6, pp. 10625-10631, 2016.
- "Design and Implementation of a Voice Controlled Home Automation System," International Journal of Computer Applications, vol. 179, no. 7, pp. 35-41, 2018.
- "Wireless Home Automation System Using Arduino," International Journal of Scientific Research in Computer Science and Engineering, vol. 5, no. 2, pp. 1-7, 2017.

CODES IN APPENDIX

4RELAYCONTROL.ino

```
int relay1 = 12;
int relay2 = 11;
int relay3 = 10;
int relay4 = 9;
char input;
void setup() {
  // put your setup code here, to run once:
Serial.begin(9600);
pinMode(relay1,OUTPUT);
pinMode(relay2,OUTPUT);
pinMode(relay3,OUTPUT);
pinMode(relay4,OUTPUT);
Serial.print("press Y to turn ON the LED\n press N to turn OFF the LED");
}

void loop() {
  input = Serial.read();
  if(input == 'Q'){
    digitalWrite(relay1,HIGH);
  }
  if(input == 'W'){
    digitalWrite(relay1,LOW);
  }
    if(input == 'E'){
    digitalWrite(relay2,HIGH);
  }
  if(input == 'R'){
    digitalWrite(relay2,LOW);
  }z
    if(input == 'T'){
    digitalWrite(relay3,HIGH);
  }
  if(input == 'Y'){
    digitalWrite(relay3,LOW);
  }
    if(input == 'U'){
    digitalWrite(relay4,HIGH);
  }
  if(input == 'I'){
    digitalWrite(relay4,LOW);
  }
  if(input == 'A'){
    digitalWrite(relay1,LOW);
    digitalWrite(relay2,LOW);
    digitalWrite(relay3,LOW);
    digitalWrite(relay4,LOW);
  }
    if(input == 'a'){
    digitalWrite(relay1,HIGH);
    digitalWrite(relay2,HIGH);
    digitalWrite(relay3,HIGH);
    digitalWrite(relay4,HIGH);
  }
  // put your main code here, to run repeatedly:

}
```

GRAPHIC USER INTERFACE PYTHON FILE

```
from PyQt5 import QtCore, QtGui, QtWidgets
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("Jarvis 2.0")
        MainWindow.resize(1440, 900)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(0, 0, 1440, 900))
        self.label.setText("")
        self.label.setPixmap(QtGui.QPixmap(
            "C:/Users/krish/Documents/Dexi/Dexi/images/gif.gif"))
        self.label.setScaledContents(True)
        self.label.setObjectName("label")
        self.pushButton = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton.setGeometry(QtCore.QRect(1180, 800, 101, 51))
        self.pushButton.setStyleSheet("background-color: rgb(0, 170, 255);\n"
                                      "font: 75 18pt \"MS Shell Dlg 2\";")
        self.pushButton.setObjectName("pushButton")
        self.pushButton_2 = QtWidgets.QPushButton(self.centralwidget)
        self.pushButton_2.setGeometry(QtCore.QRect(1310, 800, 101, 51))
        self.pushButton_2.setStyleSheet("background-color:rgb(255, 0, 0);\n"
                                        "font: 75 18pt \"MS Shell Dlg 2\";")
        self.pushButton_2.setObjectName("pushButton_2")
        self.textBrowser = QtWidgets.QTextBrowser(self.centralwidget)
        self.textBrowser.setGeometry(QtCore.QRect(640, 30, 291, 61))
        self.textBrowser.setStyleSheet("font: 75 16pt \"MS Shell Dlg 2\";\n"
                                       "background-color:transparent;\ncolor:white;"
                                       "border-radius:none;\n"
                                       "")
        self.textBrowser.setObjectName("textBrowser")
        self.textBrowser_2 = QtWidgets.QTextBrowser(self.centralwidget)
        self.textBrowser_2.setGeometry(QtCore.QRect(930, 30, 291, 61))
        self.textBrowser_2.setStyleSheet("font: 75 16pt \"MS Shell Dlg 2\";\n"
                                         "background-color:transparent;\ncolor:white;"
                                         "border-radius:none;")
        self.textBrowser_2.setObjectName("textBrowser_2")
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 1440, 26))
        self.menubar.setObjectName("menubar")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.pushButton.setText(_translate("MainWindow", "Run"))
        self.pushButton_2.setText(_translate("MainWindow", "Exit"))
if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```

MAIN VOICE CONTROL PYTHON FILE


```
import pyttsx3
import datetime
import speech_recognition as sr
import pyaudio
import smtplib
import geocoder
import wikipedia
#from newsapi import NewsApiClient
import newsapi
import pywhatkit
import requests
# from loginCREDS import senderMAIL, pwd, to
import pyautogui
import webbrowser as wb
import clipboard
import os
import time as tt
import pyjokes
import string
import subprocess as sp
import random
from time import sleep
import openai
from PyQt5 import QtWidgets, QtCore, QtGui
from PyQt5.QtCore import QTimer, QTime, QDate, Qt
from PyQt5.QtGui import QMovie
from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUiType
import sys
from guimain import Ui_MainWindow
# import pygame
import serial
ser = serial.Serial('COM3', 9600)
import pandas as pd
df = pd.read_csv('Active Faculty Data.csv')


def speak(text):        # Function to Speak
    engine.say(text)
    engine.runAndWait()


def time():     # Function to "Speak" Time
    currTime = datetime.datetime.now().strftime("%I:%M:%S")
    speak("Current Time is: ")
    speak(currTime)


def date():     # Function to "Speak" Date
    months = {1: 'Janauary',
              2: 'February',
              3: 'March',
              4: 'April',
              5: 'May',
              6: 'June',
              7: 'July',
              8: 'August',
              9: 'September',
              10: 'October',
              11: 'November',
              12: 'December'
              }
    dates = {1: 'First',
                2: 'Second',
                3: 'Third',
                4: 'Fourth',
                5: 'Fifth',
                6: 'Sixth',
                7: 'Seventh',
                8: 'Eighth',
                9: 'Ninth',
                10: 'Tenth',
                11: 'Eleventh',
                12: 'Twelfth',
                13: 'Thirteenth',
                14: 'Fourteenth',
                15: 'Fifteenth',
                16: 'Sixteenth',
                17: 'Seventeenth',
                18: 'Eighteenth',
                19: 'Nineteenth',
                20: 'Twentieth',
                21: 'Twenty-First',
                22: 'Twenty-Second',
                23: 'Twenty-Third',
                24: 'Twenty-Fourth',
                25: 'Twenty-Fifth',
                26: 'Twenty-Sixth',
                27: 'Twenty-Seventh',
                28: 'Twenty-Eighth',
                29: 'Twenty-Ninth',
                30: 'Thirtieth',
                31: 'Thirty-First'
            }

    year = str(datetime.datetime.now().year)
    month = int(datetime.datetime.now().month)
    month = months[month]
    date = int(datetime.datetime.now().day)
    date = dates[date]
    text = "Today's Date is "+date+" of "+month+" "+year
    text = str(text)

    speak(text)


def greet():        # Function to "Speak" a greeting pertaining to the current time
    hour = datetime.datetime.now().hour
    if hour <= 6 and hour < 12:
        speak("Good Morning to you!")

    elif hour >= 12 and hour < 18:
        speak("Good Afternoon to you!")

    elif hour >= 18 and hour < 24:
        speak("Good Evening to you!")

    else:
        speak("Good Night to you!")


def wishme():       # Funtion to "Speak" Initial Lines
    speak("HELLO YOU!")
    date()
    time()
    greet()
    speak("Dexi at your service!")


def takeCMD():      # Function to take CMD Commands input
    query = input("Here, input the CMD Command you want: ")
    return query


def takeMIC():      # Function to take MIC inputs
    r = sr.Recognizer()
    listenCMD = [
        'I am listening to you now...',
        'Go on, say something...',
        'Ask away, I am here...',
        'Tell me what do you need?',
        'Whats up?'
    ]
    with sr.Microphone() as source:
        r.adjust_for_ambient_noise(source)
        speak(random.choice(listenCMD))
        print("Listening to you now...")
        r.pause_threshold = 1
        audio = r.listen(source)
    try:
        print("I heard that ")
        query = r.recognize_google(audio, language="en-IN")
        print(query)
        print("You said: "+query)
    except Exception as e:
        print(e)
        speak("Say that again please...")
        return "Didn't get you, sorry"
    return query
```

```
# def sendEmail():        # Function to send E-Mails
#     server  = smtplib.SMTP_SSL('smtp.gmail.com', 465)
#     server.starttls()
#     server.login(senderMAIL, pwd)
#     server.sendmail(senderMAIL, to, 'test mail hai bhay')
#     server.close()

def sendWAPPmsg(pnum, msg):        # Function to send WhatsApp messages
    message = msg
    wb.open('https://web.whatsapp.com/send?phone='+pnum+'&text='+message)
    sleep(10)
    pyautogui.press('enter')


def searchGGL(res):        # Function to search a topic on Google
    wb.open('https://www.google.com/search?q='+res)


def weather():      # Function to look up current Weather Conditions
    g = geocoder.ip('me')
    lat = str(g.lat)
    lon = str(g.lng)
    wkey = '9bbd336b60745f2810e848836ee7fc11'
    lnk = 'https://api.openweathermap.org/data/2.5/weather?lat=' + \
        lat+'&lon='+lon+'&appid='+wkey
    print(lnk)
    return lnk


# def news(topic):        # Function to get top news headlines on specific topic
#     newsapi = NewsApiClient(api_key='424b4d1da9614c749f2b783188acf6a9')
#     data = newsapi.get_top_headlines(q=topic,
#                                      language='en',
#                                      page_size=5)
#     ndata = data['articles']
#     for x, y in enumerate(ndata):
#         print(f'{y["description"]}')
#         speak(f'{y["description"]}')

#     speak("that's all for today.")


def t2s():        # Function to read clipboard data
    text = clipboard.paste()
    print(text)
    speak(text)


def open_cmd():
    os.system('start cmd')


def open_notepad():
    notepad = "C:\\Windows\\System32\\notepad.exe"
    os.startfile(notepad)


def open_camera():
    sp.run('start microsoft.windows.camera:', shell=True)


def open_calculator():
    calculator = "C:\Windows\System32\calc.exe"
    sp.Popen(calculator)


def get_random_advice():
    res = requests.get("https://api.adviceslip.com/advice").json()
    res = res['slip']['advice']
    print(res)
    speak(res)

# def snap():        # Function to take a screenshot
#     name_img = tt.time()
#     name_img = f'C:\\work (v boring)\\ecs\\screenshot\\{name_img}.png'
#     img = pyautogui.screenshot(name_img)
#     speak("Took the screenshot!")


def pwdGEN():       # Function to Generate a new password
    s1 = string.ascii_uppercase
    s2 = string.ascii_lowercase
    s3 = string.digits
    s4 = string.punctuation
    pwdLEN = 8

    s = []
    s.extend(list(s1))
    s.extend(list(s2))
    s.extend(list(s3))
    s.extend(list(s4))

    random.shuffle(s)
    newPASS = ("".join(s[0:pwdLEN]))

    print(newPASS)
    speak(newPASS)


def gpt(q):
    openai.api_key = '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'

    completion = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "user", "content": q}
        ]
    )

    cnt = completion.choices[0].message
    val = cnt["content"]
    val = val.replace('"', '')

    print(val)
    speak(val)
```

```
# def sendEmail():        # Function to send E-Mails
#     server  = smtplib.SMTP_SSL('smtp.gmail.com', 465)
#     server.starttls()
#     server.login(senderMAIL, pwd)
#     server.sendmail(senderMAIL, to, 'test mail hai bhay')
#     server.close()

def sendWAPPmsg(pnum, msg):        # Function to send WhatsApp messages
    message = msg
    wb.open('https://web.whatsapp.com/send?phone='+pnum+'&text='+message)
    sleep(10)
    pyautogui.press('enter')


def searchGGL(res):        # Function to search a topic on Google
    wb.open('https://www.google.com/search?q='+res)


def weather():      # Function to look up current Weather Conditions
    g = geocoder.ip('me')
    lat = str(g.lat)
    lon = str(g.lng)
    wkey = '9bbd336b60745f2810e848836ee7fc11'
    lnk = 'https://api.openweathermap.org/data/2.5/weather?lat=' + \
        lat+'&lon='+lon+'&appid='+wkey
    print(lnk)
    return lnk


# def news(topic):        # Function to get top news headlines on specific topic
#     newsapi = NewsApiClient(api_key='424b4d1da9614c749f2b783188acf6a9')
#     data = newsapi.get_top_headlines(q=topic,
#                                      language='en',
#                                      page_size=5)
#     ndata = data['articles']
#     for x, y in enumerate(ndata):
#         print(f'{y["description"]}')
#         speak(f'{y["description"]}')

#     speak("that's all for today.")


def t2s():        # Function to read clipboard data
    text = clipboard.paste()
    print(text)
    speak(text)


def open_cmd():
    os.system('start cmd')


def open_notepad():
    notepad = "C:\\Windows\\System32\\notepad.exe"
    os.startfile(notepad)


def open_camera():
    sp.run('start microsoft.windows.camera:', shell=True)


def open_calculator():
    calculator = "C:\Windows\System32\calc.exe"
    sp.Popen(calculator)


def get_random_advice():
    res = requests.get("https://api.adviceslip.com/advice").json()
    res = res['slip']['advice']
    print(res)
    speak(res)

# def snap():        # Function to take a screenshot
#     name_img = tt.time()
#     name_img = f'C:\\work (v boring)\\ecs\\screenshot\\{name_img}.png'
#     img = pyautogui.screenshot(name_img)
#     speak("Took the screenshot!")


def pwdGEN():       # Function to Generate a new password
    s1 = string.ascii_uppercase
    s2 = string.ascii_lowercase
    s3 = string.digits
    s4 = string.punctuation
    pwdLEN = 8

    s = []
    s.extend(list(s1))
    s.extend(list(s2))
    s.extend(list(s3))
    s.extend(list(s4))

    random.shuffle(s)
    newPASS = ("".join(s[0:pwdLEN]))

    print(newPASS)
    speak(newPASS)


def gpt(q):
    openai.api_key = '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'

    completion = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "user", "content": q}
        ]
    )

    cnt = completion.choices[0].message
    val = cnt["content"]
    val = val.replace('"', '')

    print(val)
    speak(val)
```