Query Response using Voice Bot

Submitted in partial fulfilment of the requirements for the degree of

Bachelor of Technology
in
ARTIFICIAL INTELLIGENCE

by

B.Adarsh Reddy -18BCI0196
Vikas -18BCI0193

Under the guidance of
Prof. Subhashini R Scope VIT, Vellore.

VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

***Abstract* - Human-computer interaction is increasing with the advancement of technology and so is the Voice Interaction with the system. We are trying to develop a voice-based search engine which recognizes the questions asked by the user and searches the answer for them in a very optimized way and gives the appropriate results. Queries asked to the bot are processed and the bot searches for the keywords on various search engine platforms and gives the results.**

Index Terms – **Chat Bot, Interface, Text to Speech**

I. Introduction

Speech probably is the most natural and efficient way to communicate. Thus, it is the best way for communication, it could also be a useful interface for communication with systems and machines like Interactive Voice system. This system when combined with speech recognition technology can play an important role in providing efficient and easy user service[1]. It can offer new services and increase the user satisfaction. Speech Recognition is now dominating the market technology and it's taking over the traditional way of using hectic interfaces such as mouse and keyboards as input source to the computer system. Speech recognition based applications will make life easier due to the fact that people will get fast and easy access to the information they need. Therefore the popularity of automatic speech recognition system has seen leaps in the past few years.

The work of speech recognition further helps in establishing an easy way communication between interactive response system and users i.e. as a part of post processing of the speech recognition we can execute some computational tasks with such a system making speech as a trigger to start doing some task in the system.

In this project our main focus is to develop such an application where users can simply command the system with their speech and in response the system does its task as per the user's request.

II. Literature Survey

- **An Intelligent Web-Based Voice Chat Bot**

  Authors: Manoj Lall, S. Sinha, S.J. du Preez.

This paper provides a web-based technology which consists of three parts. They are the client, server and data acquisition. The client has to collect the audio data from the user. Processing the audio data and converting into text can be done on the client -side as well as the server side. Here we follow client -side processing to reduce the computational time and resources when we are dealing many parallel inputs at the same time for server. The data acquisition phase collects the text data searches for key words and comes up with a solution. The testing of the software is done using the black box technique.

- **Smart Voice Search Engine**

  Authors: Shahendra Sahan

This paper provides us with smart voice-based search engine. The paper focuses on the speech recognition phase. Initially the audio signal is taken, it is converted into digital signals. These digital signals are split into frames and sent to the acoustic analysis. After the analysis, the keyword is generated and is sent to the search engine for getting the result. The final query result is given to its user. The problems faced by the designers are explained later in the paper.

- **Interactive Voice Response System**

  Authors: John Brian Pickering

This paper provides a method for processing an interactive voice processing system. It consists of receiving a voice signal from the user, extracting a set of measurements from the voice signal; calculating an average of the said measurements, locating a reference characteristic matching the average; and using text associated with the closest reference characteristic as an estimate of the text of the Voice Signal.[3]

- **Chatbot Using Knowledge in Database: Human to Machine Conversational Modelling**

  Authors: Bayu Setiaji and Ferry Wahyu Wibowo

A chatbot tries to communicate with humans in a natural way. It takes input from the user, finds the similarity between the words of input and various referential knowledge of the system. It consists of the Relational Database

Management System (RDBMS) and the interface that is accessing the core of RDBMS. The similarity decides the strength of the answer given by the system. The interface is designed using the languages Pascal and Java

- **Development of Speech-to-Text Chatbot Interface Based on Google API**

  Authors: Nataliya Shakhovska, Oleh Basystiuk and Khrystyna Shakhovska

This paper describes possibilities which are provided by open API's like Google for creating unified interfaces. Speech recognition is growing popularly in the field of natural language processing as it saves the communication time. Audio data is collected from various sources such as the social networks and neural networks are used for training this data. Python language and web framework Flask are used for this paper. The complexity of finding the keyword in the database is reduced using Hash bot algorithm

III. Methodology

Chat bots typically provide a speech and text-based user interface, allowing the user to speak the commands and receive as speech to text response. Chat bots are usually a stateful services[4]. When chat bot technology is integrated with popular web services such as Wikipedia, it can be utilized securely by an even larger audience. So, in this project, we designed a bot that receives the text from converting the speech to text and then querying the text through the Wikipedia and giving the response about the information related to the query.

- A Wiki-Chat Bot project is developed using AI techniques that analyze user's queries and processes the query.
- This system is an application which provides appropriate responses to the query effectively.
- Users just have to query through the bot through speech
- The System uses artificial intelligence to answer the query.
- The results are accurate according to the user's queries.
- The system analyzes the query and then responds to the user.
- The user can query about any questions with the help of this application.
- This system helps the users to search easier.

Features:

The system consists of three modules as mentioned below:

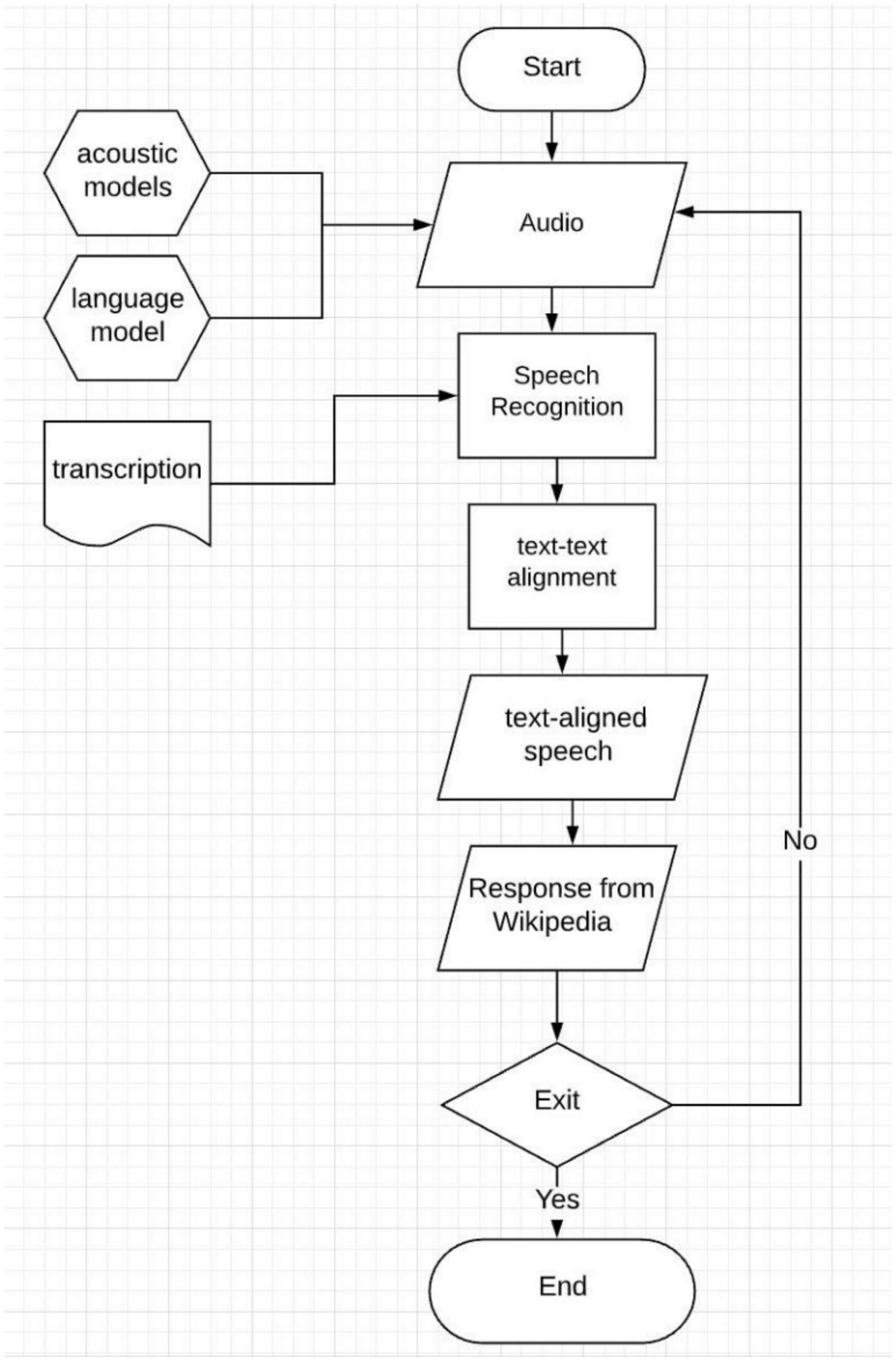
1. **User Interface**
2. **Speech to Text**
3. **Wiki-Bot Search**

Description:

1. **User Interface**
   - User interface in which user can ask the queries to the system with the help of wiki-bot.
2. **Speech to Text :**
   - User asks his query and through speech to text conversion this module provides the text to the next module.

3. **Wiki-Bot Search:**
   - Bot obtains the text from the above module and then searches it in Wikipedia and gives the response on the Interface.

Complete Design:



Complete Description:

TKINTER

- It is an inbuilt module that comes with python that allows you to create graphical user interfaces relatively easy and quickly.
- It creates a graphical user interface, we can set the size of the window, create buttons, print text onto the interface and run the loop
- To download the package onto the system we use the following command: sudo apt – get install python3 – tk
- Tkinter is imported into the program. Tkinter GUI operations are all performed as considering objects as widgets
- The root widget is declared and run till the program is closed using the main loop function.
- Two different windows are created acting as the input and the output window. Their titles are set, and the pack() function is used to print the data into the interface

WOLFRAMALFA

- WolframAlpha is a library which helps in building the main program which searches for the query and provides us with the result of the query.
- The package is downloaded from the internet using the command: python3 install –m pip install WolframAlpha
- This package is imported into the program and a variable is created called appID which stores the app ID of the WolframAlpha application.
- We give this ID to the client and open up the connection.
- When the query is converted from speech to text, it is given as input for the WolframAlpha using the query() function.
- The result of the query is taken out and forwarded to the text to speech to give output tao the user.

WIKIPEDIA and other packages

- Wikipedia is a vast source of information over the internet.
- Adding Wikipedia to the program helps us find apt answers to the keyword we have searched.
- It is downloaded into the Ubuntu system using the command:
  python3 –m pip install Wikipedia
- Some of the other packages include the Speech Recognition, pyaudio which stands for Python Audio, play sound and gTTs
- Downloading the packages and installing into the system is as follows:
  python3 –m pip install SpeechRecognition
  python3 –m pip install gTTs
  python3 –m pip install playsound
  python3 –m pip install pyaudio

Implementation:

Code:-

```
from tkinter import *
import tkinter as tk
import wolframalpha
import wikipedia
import requests
import webbrowser
import speech_recognition as sr
from gtts import gTTS
from playsound import playsound
import os
appId = 'E49Q54-Y8A9UHJ8LT'
client = wolframalpha.Client(appId)
root = Tk()
root.title('question page')
root.geometry("300x150")
pri = Text(root, height=3)
reswin = Tk()
reswin.title('answer page')
robo = Text(reswin)
def search_wiki(keyword=''):
 searchResults = wikipedia.search(keyword)
 if not searchResults:
  print("No result from Wikipedia")
  robo_print("No result from Wikipedia",robo)
 try:
  page = wikipedia.page(searchResults[0])
 except (wikipedia.DisambiguationError, err):
  page = wikipedia.page(err.options[0])
 wikiTitle = str(page.title.encode('utf-8'))
 wikiSummary = str(page.summary.encode('utf-8'))
 print(wikiSummary)
 robo_print(wikiSummary,robo)
def search(text=''):
 res = client.query(text)
 if res['@success'] == 'false':
   search_wiki(text)
 else:
  result = ''
  pod0 = res['pod'][0]
  pod1 = res['pod'][1]
  if (('definition' in pod1['@title'].lower()) or ('result' in pod1['@title'].lower()) or (pod1.get('@primary','f
alse') == 'true')):
   result = resolveListOrDict(pod1['subpod'])
   print(result)
   robo_print(result,robo)
   textspeech = gTTS(text = result, lang='en')
   textspeech.save('test.mp3')
   playsound('test.mp3')
   os.remove('test.mp3')
   question = resolveListOrDict(pod0['subpod'])
   question = removeBrackets(question)
   primaryImage(question)
  else:
   question = resolveListOrDict(pod0['subpod'])
   question = removeBrackets(question)
   search_wiki(question)
   primaryImage(question)
def removeBrackets(variable):
 return variable.split('(')[0]
def resolveListOrDict(variable):
 if isinstance(variable, list):
  return variable[0]['plaintext']
 else:
  return variable['plaintext']
def primaryImage(title=''):
  url = 'http://en.wikipedia.org/w/api.php'
  data = {'action':'query', 'prop':'pageimages','format':'json','piprop':'original','titles':title}
  try:
    res = requests.get(url, params=data)
    key = res.json()['query']['pages'].keys()[0]
    imageUrl = res.json()['query']['pages'][key]['original']['source']
    print(imageUrl)
    webbrowser.open(imageUrl)
  except (Exception, err):
    print('')
def main_function():
 flag = 0
 r = sr.Recognizer()

 with sr.Microphone() as source:
  print('say now')
  audio = r.listen(source)
 try:
  q=r.recognize_google(audio)
  flag=0
 except:
  robo_print('didnt understand, plese repeat your sentence',robo)
  human('sorry didnt understand',0)
  flag = 1
 if flag == 0:
  print(r.recognize_google(audio))
  human(r.recognize_google(audio),1)
  word = ‘Subhashini R'
  qu = 'exit'
  if word in q.lower():
   matter='she is my AI teacher'
   textspeech = gTTS(text = matter, lang='en')
   robo_print(matter,robo)
   playsound('testos.mp3')
  if qu in q.lower():
   exit()
  else:
   search(q)
```

```
def wiki_print(reply):
 robo.delete('1.0',END)
 robo.insert(END,reply)
 robo.pack(side = LEFT, fill = X)
def robo_print(reply,robo):
   robo.delete('1.0',END)
   robo.insert(END,"Robo: \n"+reply)
   robo.pack(side = LEFT, fill = X)
def human(question,i):
 global pri
 pri.delete('1.0',END)
 pri.pack(side = RIGHT, fill = X)
 if i == 1:
  pri.insert(END,"User: "+question)
 else:
  pri.insert(END,"Sorry didnt understand your statement")
voice = Button(root, text="speak", width=40, bg="red", command = main_function)
voice.pack()
reswin.mainloop()
root.mainloop()
```

Result and Discussion:

In this project the Dataset Used is WikiQA corpus. The WikiQA corpus is a new opensource dataset which consists of a set of question and their respective answers, collected for usage on open-domain question answering.

Innovation Introduced in Design:

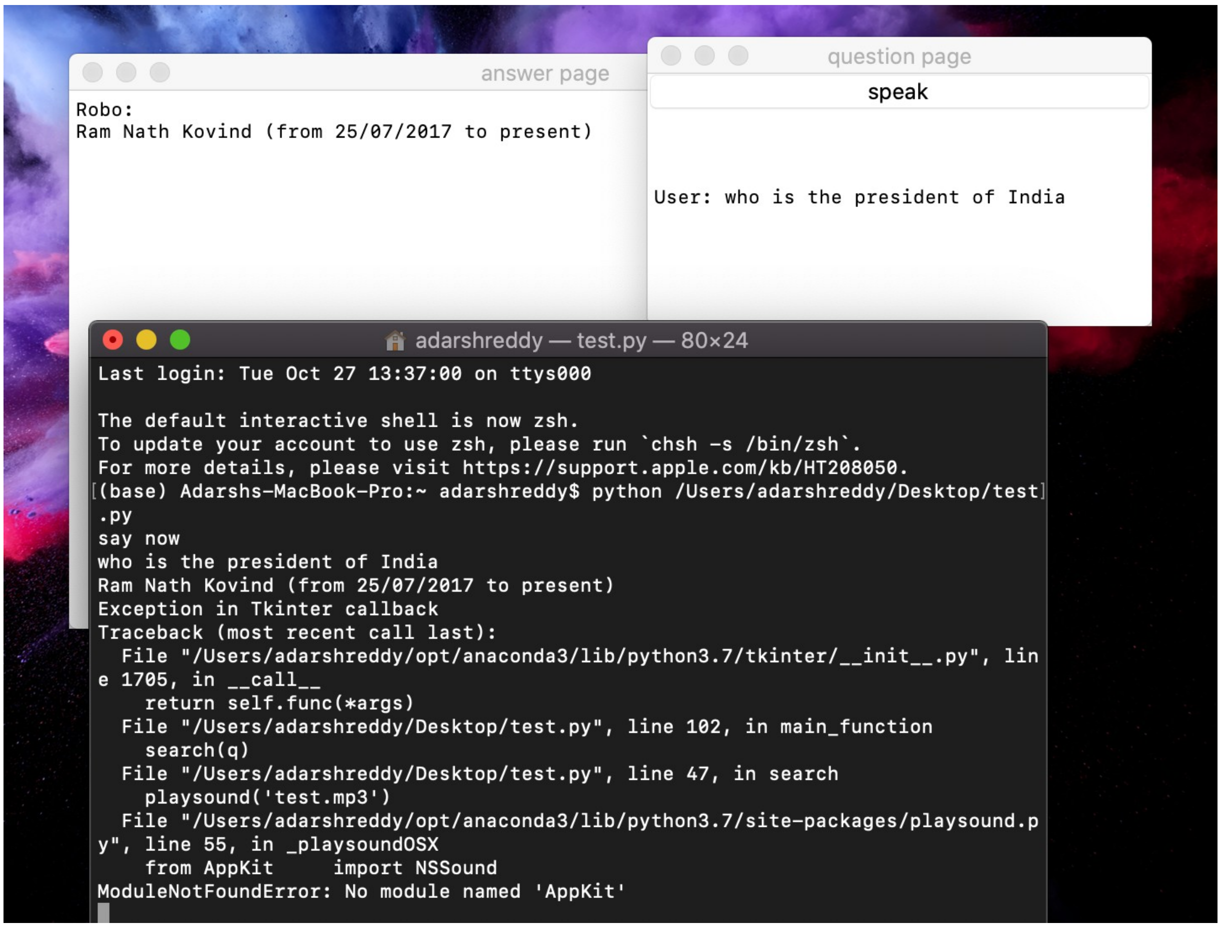
The major advantage of this project over other existing technologies is that it doesn't need a server and a client to perform the operation. All the operation is being performed on the client's i.e., on the user's computer.The program runs by itself and doesn't connect to any server which helps reduce the problem of heavy resources when parallelly processing large amounts of data.Another advantage is that it depends on one of the best search sites i.e., Wikipedia for its information. Wikipedia is the most reliable source of information on the internet.

New Learning experience gained in the process of the project Like using new software package, new fabrication technique etc:

After starting this project our team have got an idea query processing and using voice recognition in python. In order to build project from that idea we had to learn whole lot of new concepts. We learned about many inbuilt python modules such as Tkinter, Wolframaplha, wikipedia and so on. And we faced few problems integrating voice recognition for query-processing but, later sorted it using pysound and port audio in condo environment. The overall process was a learning curve and we get to understand few base concepts in artificial interrogation domain.

Result and Discussion:

Initial Question and Answer Page after running the program



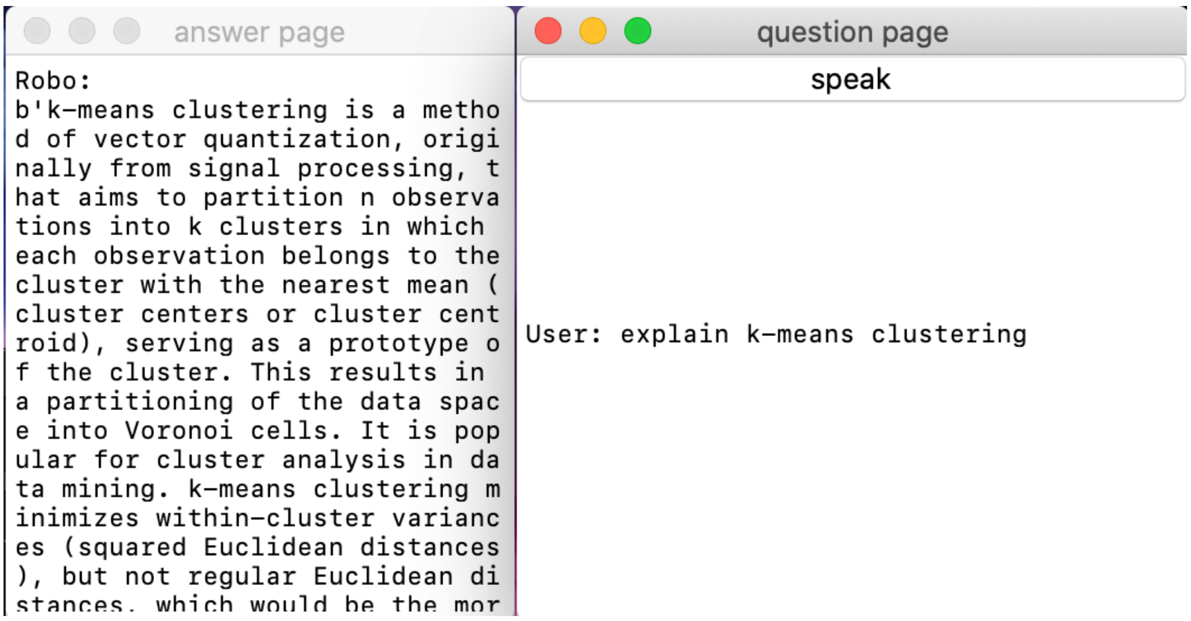
```
Robo:
Ram Nath Kovind (from 25/07/2017 to present)
```

```
User: who is the president of India
```

```
Last login: Tue Oct 27 13:37:00 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
(base) Adarshs-MacBook-Pro:~ adarshreddy$ python /Users/adarshreddy/Desktop/test
.py
say now
who is the president of India
Ram Nath Kovind (from 25/07/2017 to present)
Exception in Tkinter callback
Traceback (most recent call last):
  File "/Users/adarshreddy/opt/anaconda3/lib/python3.7/tkinter/__init__.py", lin
e 1705, in __call__
    return self.func(*args)
  File "/Users/adarshreddy/Desktop/test.py", line 102, in main_function
    search(q)
  File "/Users/adarshreddy/Desktop/test.py", line 47, in search
    playsound('test.mp3')
  File "/Users/adarshreddy/opt/anaconda3/lib/python3.7/site-packages/playsound.p
y", line 55, in _playsoundOSX
    from AppKit     import NSSound
ModuleNotFoundError: No module named 'AppKit'
```

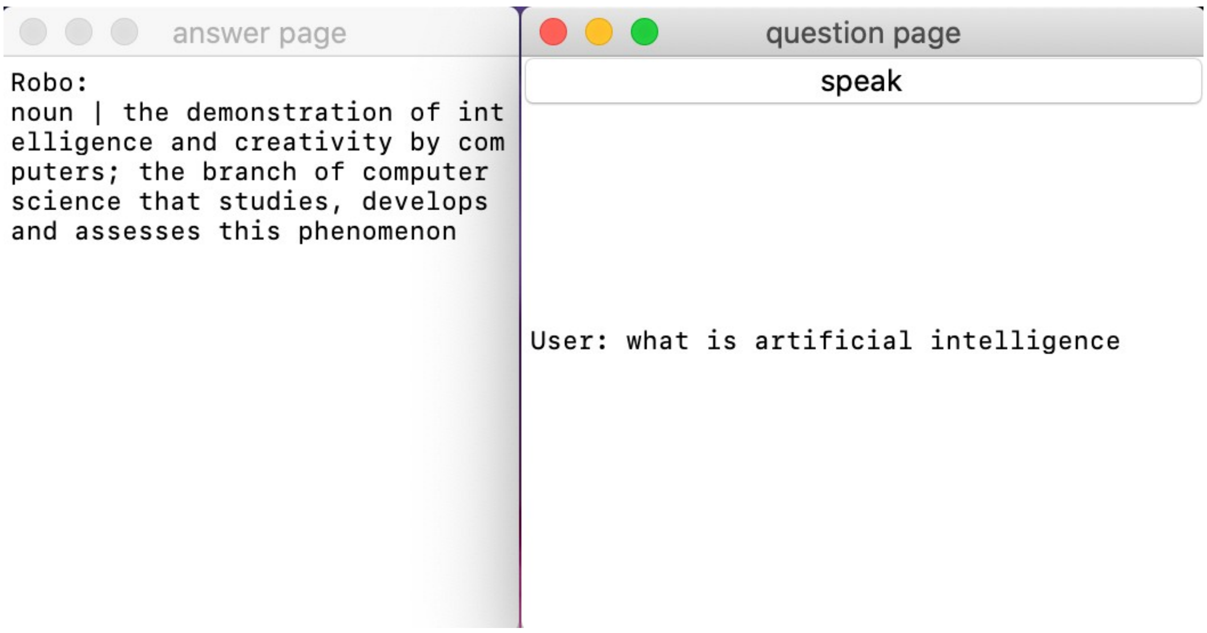
After clicking on the speak button, the user speaks which is converted to text and taken for query and result is generated.



```
Robo:
b'k-means clustering is a metho
d of vector quantization, origi
nally from signal processing, t
hat aims to partition n observa
tions into k clusters in which
each observation belongs to the
cluster with the nearest mean (
cluster centers or cluster cent
roid), serving as a prototype o
f the cluster. This results in
a partitioning of the data spac
e into Voronoi cells. It is pop
ular for cluster analysis in da
ta mining. k-means clustering m
inimizes within-cluster varianc
es (squared Euclidean distances
), but not regular Euclidean di
stances, which would be the mor
```

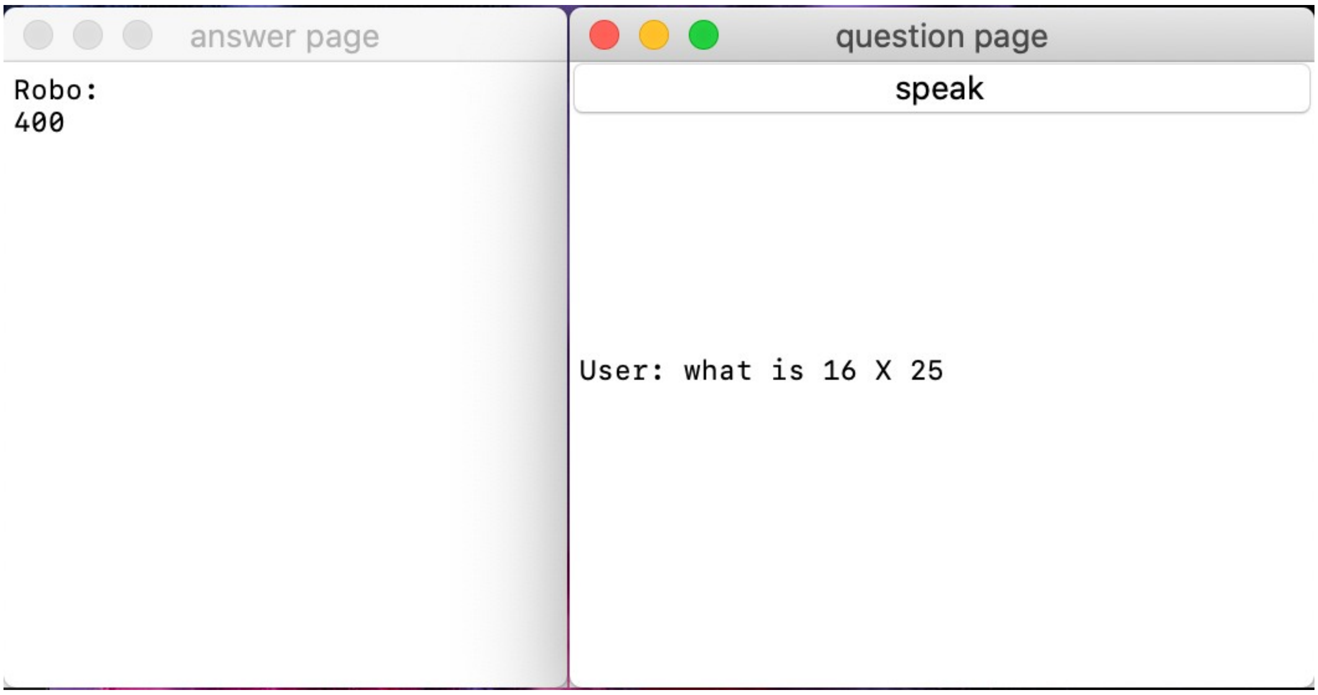
```
User: explain k-means clustering
```

Some Other test cases



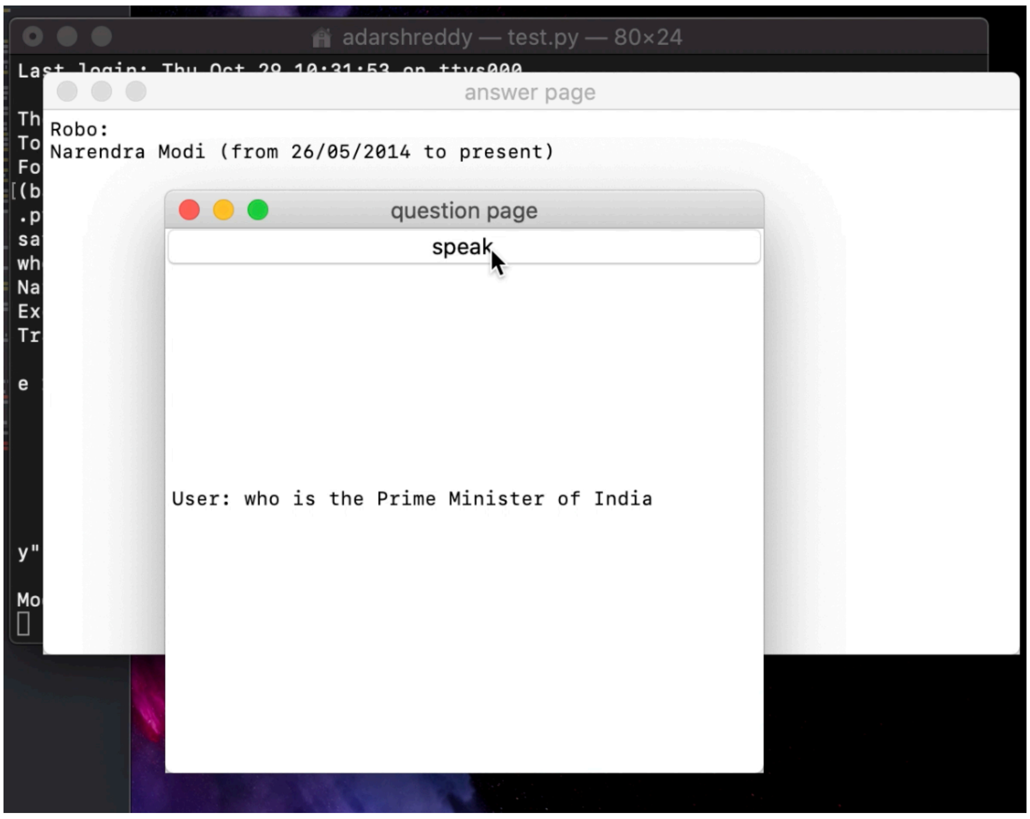
```
Robo:
noun | the demonstration of int
elligence and creativity by com
puters; the branch of computer
science that studies, develops
and assesses this phenomenon
```

```
User: what is artificial intelligence
```



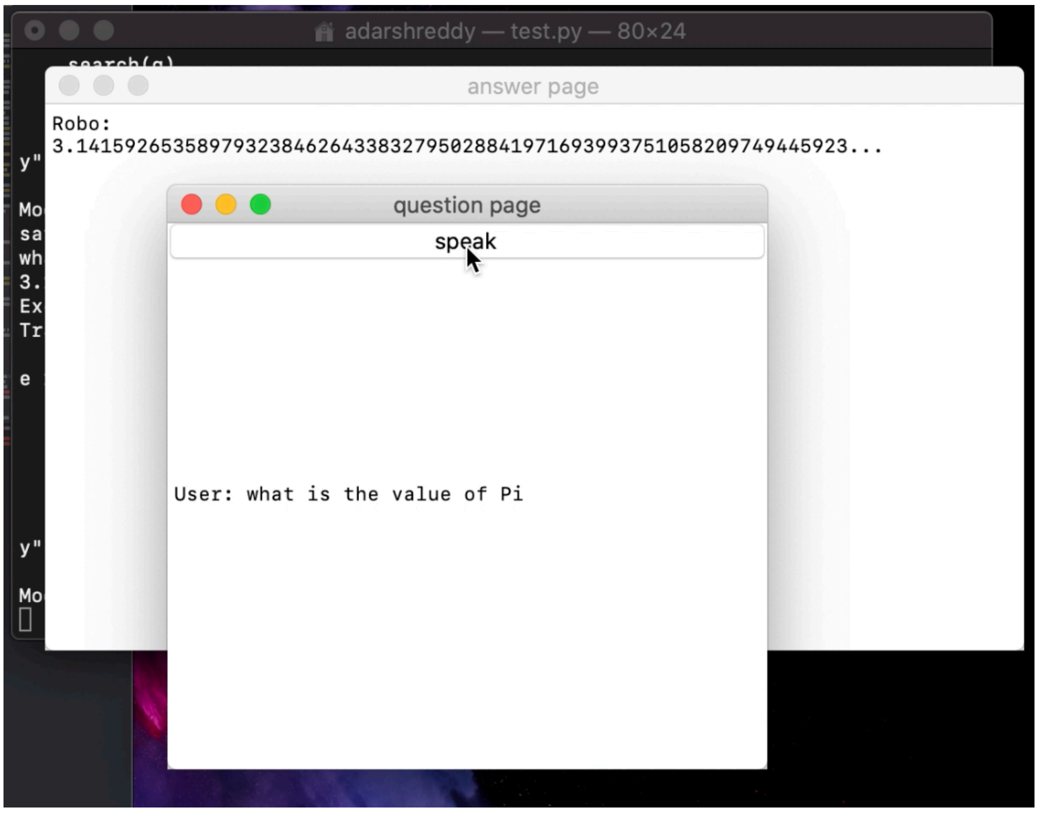
```
Robo:
400
```

```
User: what is 16 X 25
```



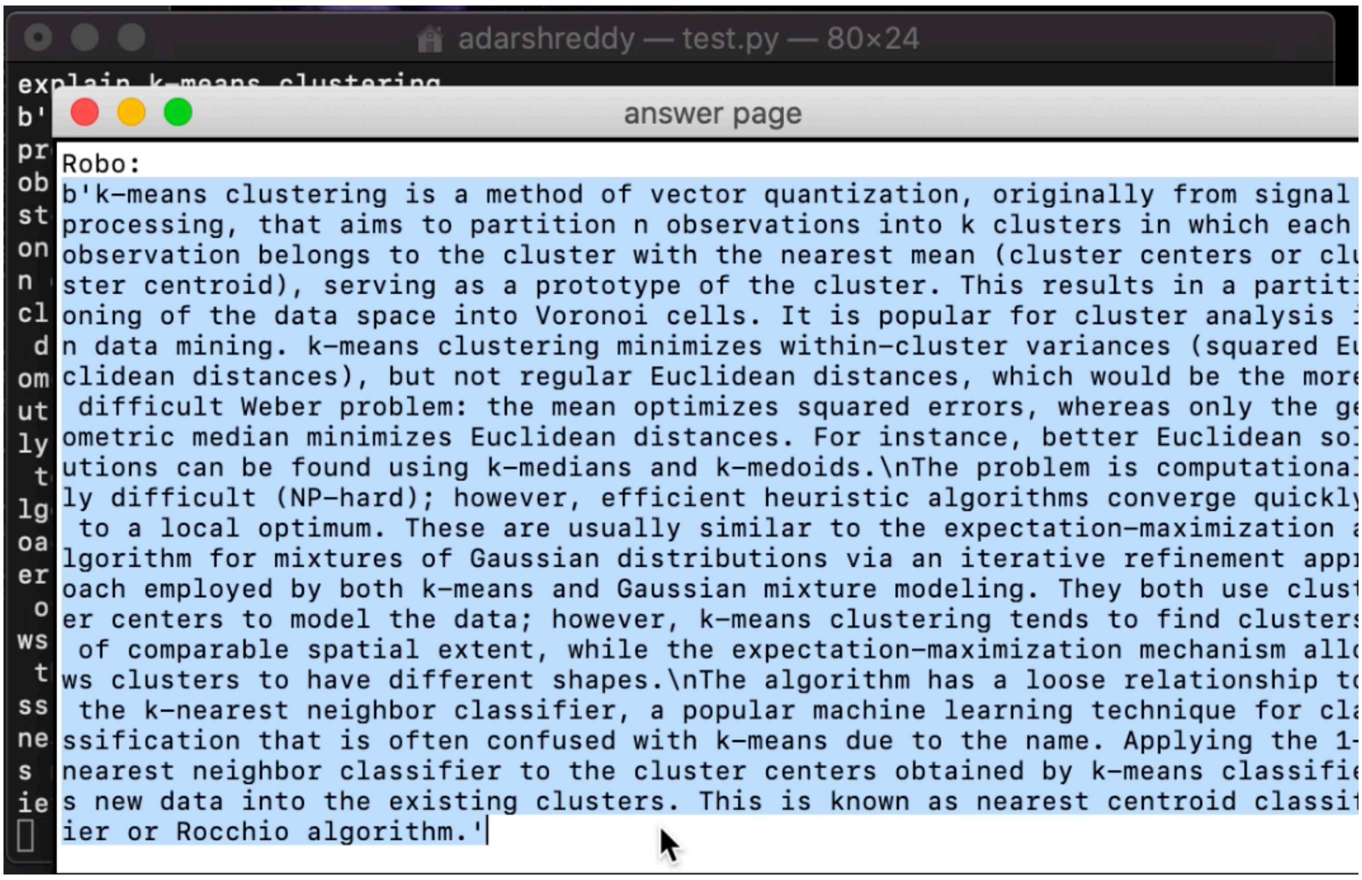
```
Robo:
Narendra Modi (from 26/05/2014 to present)
```

```
User: who is the Prime Minister of India
```



```
Robo:
3.1415926535897932384626433832795028841971693993751058209749445923...
```

```
User: what is the value of Pi
```



```
explain k-means clustering
b'
pr
ob
st
on
n
cl
 d
om
ut
ly
 t
lg
oa
er
 o
ws
 t
ss
ne
s
ie
```

```
Robo:
b'k-means clustering is a method of vector quantization, originally from signal
processing, that aims to partition n observations into k clusters in which each
observation belongs to the cluster with the nearest mean (cluster centers or clu
ster centroid), serving as a prototype of the cluster. This results in a partit
oning of the data space into Voronoi cells. It is popular for cluster analysis i
n data mining. k-means clustering minimizes within-cluster variances (squared Eu
clidean distances), but not regular Euclidean distances, which would be the more
 difficult Weber problem: the mean optimizes squared errors, whereas only the ge
ometric median minimizes Euclidean distances. For instance, better Euclidean sol
utions can be found using k-medians and k-medoids.\nThe problem is computationa
ly difficult (NP-hard); however, efficient heuristic algorithms converge quickly
 to a local optimum. These are usually similar to the expectation-maximization a
lgorithm for mixtures of Gaussian distributions via an iterative refinement appr
oach employed by both k-means and Gaussian mixture modeling. They both use clust
er centers to model the data; however, k-means clustering tends to find clusters
 of comparable spatial extent, while the expectation-maximization mechanism allo
ws clusters to have different shapes.\nThe algorithm has a loose relationship to
 the k-nearest neighbor classifier, a popular machine learning technique for cla
ssification that is often confused with k-means due to the name. Applying the 1-
nearest neighbor classifier to the cluster centers obtained by k-means classifie
s new data into the existing clusters. This is known as nearest centroid classif
ier or Rocchio algorithm.'
```

Conclusion:

Using Speech Recognition is one of the emerging technologies of the present time. Communication with a computer using audio helps us save time and provides us with efficient results.Wikipedia is one of the modern source of information and many of the websites depend on it for apt results. With the help of this wikichat bot, User does not have to personally check the answer from Browser. This application enables the users to obtain all the information that is present in Wikipedia. This application saves time and increase effectiveness of query processing.

Ultimate Utility value of the project to the society / Future Scope:

This wiki-bot will Respond to a speaker naturally in real time by knowing what is being said by using the combination of Speech Transcription, Language Identification, and also uses an NLP layer, for making sure that our voice bot always understands the context and reduces the errors.
voice bots have many real time applications such as working in call centers, your mobile Voice assistant, and AI-powered digital assistants that engage with customers with natural language to answer basic questions.