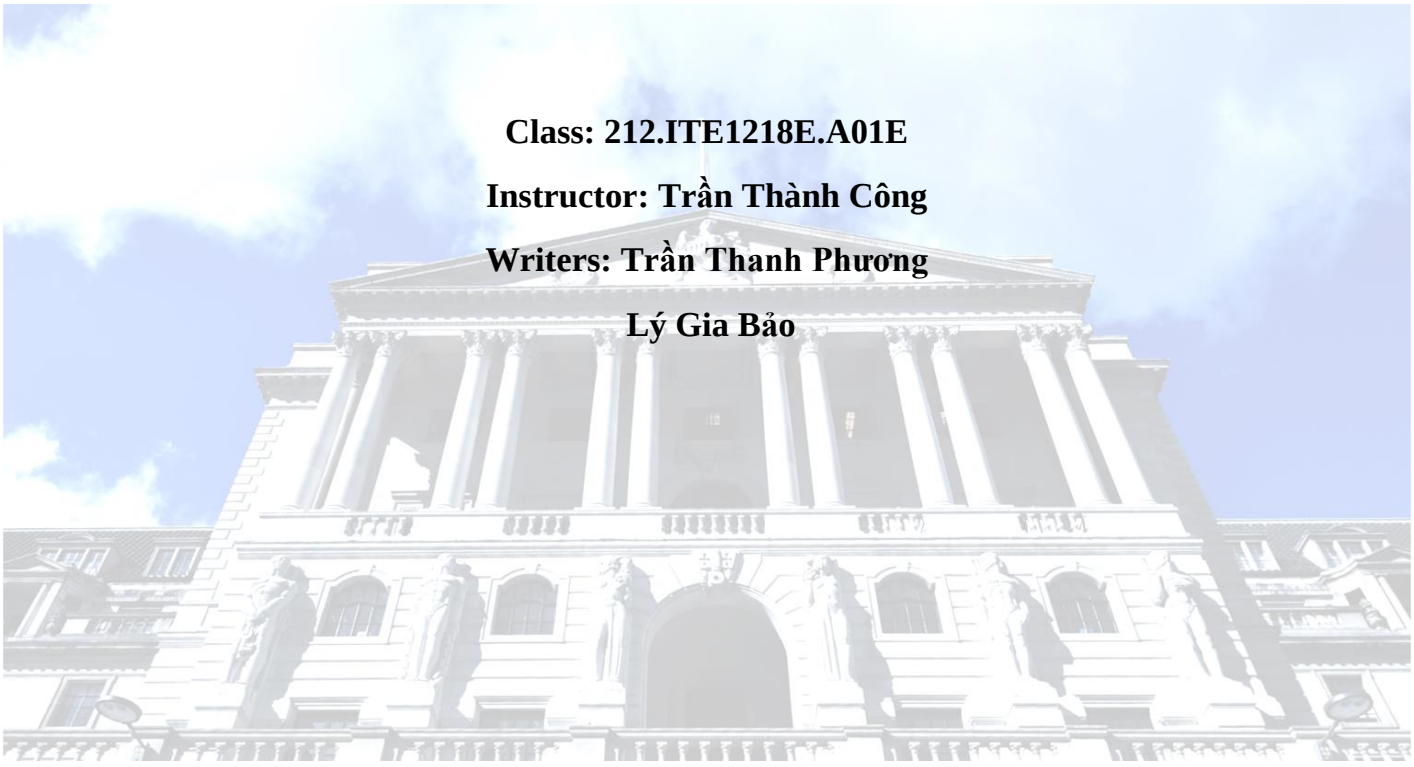
THE UNIVERSITY OF ECONOMICS AND FINANCE

INFORMATION TECHNOLOGY DEPARTMENT

THE REPORTS

SUBJECT: INTRODUCTION TO ARTIFICIAL INTELLIGENCE

PROJECT: CONSTRUCTING A VIRTUAL ASSISTANT BY PYTHON - BEE



Class: 212.ITE1218E.A01E

Instructor: Trần Thành Công

Writers: Trần Thanh Phương

Lý Gia Bảo

22/04/2022

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

In recent years, science and technology have made great strides. If ten years ago, Virtual Assistant was an unfamiliar term, it has Been researched and developed at a rapid pace by research institutes, universities and the tycoons in the software industry. Virtual Assistant, also known as Chatbot, is a rudimentary form of artificial intelligence, a computer created program conducts conversations through keyboard inputs, audio and returns the desired results by voice for human. It can answer the questions, handle given problems, tasks and interact with human through a pre-programmed artificial intelligence. The interface of a virtual assistant would be natural for human, which leads to the convenience of automation of various activities in daily life.

This project aims to build a personal-assistant called BEE, which has ability to execute duties by employing certain well-defined commands with user-friendly interface. Users can interact with the assistant either through voice commands or using keyboard input. As a personal assistant, BEE assists the end-user with basic activities like general human conversation, searching queries in Google, Wikipedia, announcing date and time.

BEE mainly built based on Python programming language, takes inputs as voice from devices as microphone, automatedly replies and carries out the missions.

Chapter 1: OVERVIEW

1.1. The reason to choose this topic:

Nowadays, Virtual Assistant is not an unfamiliar term with everyone, especially, with the appearance of some software like Cortana of Microsoft, Siri of Apple, Google Now of Google or Alexa of Amazon. They have a bunch of applications in different fields such as entertainment, medical, education, commerce, automation and so on. According to Gartner prediction (the world's leading information technology research and consulting company) that over 85% customer interactions will be managed without the human involvement. Furthermore, Gartner predicts that, by 2025, 50% of knowledge workers will use a virtual assistant on a daily basis, up from 2% in 2019. By 2023, Gartner predicts that 25% of employee interactions will be voice-based communications.

The operation modes on the computer and Smartphone are by working with gestures and through the keyboard. It is not a convenient way for users with completely manually input. The common way of communication used by people in daily life is through the speech. If the mobile phone and digital gadgets can listen to the user for the request or handle the daily affairs, then give the right response, it will be much easier. At that time, the digital gadgets can unleash its full potential and become as intelligent as the human assistant.

One of the factors that make up the power of virtual assistants is the ability to self-study. The more used and interacted with users, the smarter the digital assistant platform becomes. Intelligent chatbot have the ability to self-learn based on input data without having to be specifically programmed (which was called machine learning). This makes it easy for developers to create chat programs and automate conversations with users.

As virtual assistance is a weak AI, which means that it is a basic form of artificial intelligence, so we choose this topic not only based on its practicality and development trend, but also we hope to have an opportunity to learn algorithms, models to build a personal virtual assistant as well as issues surrounding it.

1.2. The targets of this project:

The chief purpose of this project is to learn about the structure of a virtual assistant and its application in reality, specifically through building a personal virtual assistant named BEE that can perform plain tasks given by humans.

1.3. The contents of research:

- Content 1: Learn about artificial intelligence, chatbots and virtual assistants
- Content 2: Learn about the Python programming language and its packages support construction of BEE
- Content 3: Learn about the modules as well as methods needed in creating a virtual assistant
- Content 4: Learn how to build a virtual assistant with Python code
- Content 5: Evaluate the results
- Content 6: Write a report

1.4. The scope:

Presently, BEE is being developed as an automation tool and virtual assistant. Among the various roles played by BEE are:

1. Greeting to users and introduce about itself
2. Take commands of human by voices and returns the output by voice or completing the mission be given
3. Announce the date and time to users in request
4. Searching specified information on the web
5. Look up the information from Wikipedia page
6. Open the video on YouTube platform

1.5. The layout of the report:

- **Chapter 1: The overview**

This chapter presents the reason to choose topic about the virtual assistant, targets, research contents, the cope and the layout of the project.

- **Chapter 2: Theoretical basis**

  Introduce about Artificial Intelligence, Virtual Assistants as well as its applications.

- **Chapter 3: Preparing for building a virtual assistant by Python**

  The information about preparing environment, programming language, modules needed and methods used for virtual creating the virtual assistant BEE are provided in this chapter.

- **Chapter 4: Constructing the system of BEE**

  The steps of installing the modules needed as well as building BEE are expressed. Besides, the functionalities of BEE are also revealed.

- **Chapter 5: Result**

  The presentation of the results obtained after successfully building the BEE.

- **Chapter 6: Conclusion and development**

  Summary of results achieved, limitations and future development directions.

Chapter 2: THEORETICAL BASIS

2.1. Introduction about Artificial Intelligence:

Today, artificial intelligence appears everywhere. AI is the tendency that giant technology companies are aiming for namely Google, Facebook, Amazon, Microsoft and others. Those magnates are investing heavily in artificial intelligence to create products that serve human interests.

The artificial intelligence is the intelligence represented by any artificial system. This terminology generally refers to computers of no particular purpose and the science that studies the theories and applications of artificial intelligence. It deals with machine behavior, learning and smart adaptability.

Bellman (1978) defined that AI is the automation of activities consistent with human thinking, such as decision-making, problem solving, and so on.

Rich and Knight (1991) believed that: AI is the science which studies about how computers can do jobs that humans can currently do better than computers.

Each concept, definition has its own correct points, however, for simplicity, AI can be understood as a branch of computer science. It is built on a solid theoretical foundation and can be applied in automating the intelligent behavior of computers. It helps computers acquire human intelligences such as: thinking and reasoning to solve problems, communicating by understanding language and speech, learning and self-adaptation. It is something we use to reply to text messages automatically, learn to drive, fly an airplane so we can sit there and watch without needing to control it, rearrange pictures of different outings into separate albums. It even helps to manage home while away or shopping.

2.2. Virtual assistants:

2.2.1. Definitions:

A virtual assistant is a software agent capable of performing tasks or services for individual users. Sometimes people also use the term "chatbot" to refer to virtual assistants accessed by online chat software (or to refer to online chat programs that are entertaining and do not bring many uses). Calculated to 2017, the features and users of virtual assistants are growing rapidly with new products coming to market all the time. An online survey in May 2017 found that, the most widely used virtual assistants in the United States are Apple Siri 34%, Google Assistant 19%, Amazon Alexa 6%, and Microsoft Cortana 4%. Apple and Google have the most installed virtual assistants on smartphones, and Microsoft is on personal computers running its Windows operating system (its virtual assistant Cortana is integrated into this operating system with smart phones and speakers); while, Alexa is the first product with the ability to order online from Amazon.

For example, if you asked Siri of Apple or Cortana of Microsoft something like: “How is the weather today?”, it means that you have worked with a Virtual Assistant or a Chatbot.

2.2.2. The history of Virtual Assistant:

a. <u>Turing Test (1950):</u>

The Turing test was invented in 1950 by Alan Turing. The Turing test is a test of intellectual ability of a computer. Standard model of the Turing test, in which player C acts as the interrogator, whose task is to determine 2 player A and B, which side is computer, and which side is human, by asking and receiving questions answers from A and B. The Turing test is based on the assumption that one can judge the "intelligence" of a computer by comparing its behavior with that of a human. The question is: can test results reflect reality, while only considering behavior and comparing it with human behavior? For this and other reasons, artificial intelligence researchers have wondered the usefulness of the test. In practice the outcome of an experiment can easily be

influenced not by the intelligence of the computer, but by the skill, attitude, or naivety of the questioner.

b. Eliza (1966):

The first chatbot was born in 1960, named Eliza, and is a computer program of Joseph Weinbaum (Massachusetts Institute of Technology, USA). The program is designed in a way that mimics human conversation. The Eliza chatbot works by passing words that the user has entered into the computer and then concatenating them into a list of scripted responses. It uses a scenario that simulates a psychotherapist. The script proved to be a significant impact on natural language processing and artificial intelligence and was one of the first programs that could pass the Turing test.

c. IBM Shoebox (1962), Harpy (1970s) and Tangora (1986):

Not much later after Eliza was announced, another advancement in digital speech recognition was produced by IBM with its Shoebox voice-activated calculator, presented to the general public during the 1962 Seattle World's Fair. The 1970s was the decade of voice recognition where companies and academia including IBM, Carnegie Mellon University, and Stanford Research Institute collaborated. The result was "Harpy," a machine that mastered about 1,000 words, with the vocabulary of a three-year-old that could understand sentences. It could process speech that followed pre-programmed vocabulary, pronunciation, and grammar structures to determine which sequences of words made sense together, and thus reducing speech recognition errors.

d. Dr. Sbaitso (1992):

Dr. Sbaitso is a chatbot created by Creative Labs for MS-Dos in 1992. It was one of the earliest attempts of incorporating AI into a chatbot and is recognized for its full dialogue program. The program "chats" with the user like a psychologist, most often the question is "How do you feel?" rather than complex interactions. When faced with a complex question, Dr. Sbaitso often replies: "It's not my problem".

e. IBM Simon (1994):

The birth of the first virtual assistant; however, began with IBM Simon in the early 1990s. It was a digital speech recognition technology that became a feature of the personal computer with IBM and Philips.

f. Alice (1995):

Alice is a universal language processing chatbot that uses a heuristic interaction pattern to conduct conversations. In it, heuristics are algorithms based on experience to solve a problem, learn or discover in order to provide a solution at the common sense level. In 1995, Richard Wallace pioneered the construction of Alice.

Alice is a program that works with an XML schema called an artificially intelligent markup language (AlMI), which helps define chat rules. In 1998, the program was modified in Java, and in 2001, Wallacc specified the specification of AIMI in Alice. Since then, other developers have compiled free and extensible Alice sources in various programming languages and a variety of foreign languages.

g. SmarterChild (2001):

In the early 2000s, the first chatbot, familiar to most today, was technically invented by Colloquis, which launched SmarterChild on platforms like AIM and MSN Messenger. It was entirely text-based and was able to play games, check the weather, look up facts, and converse with users. It is also considered the precursor to Apple's Siri.

h. Watson of IBM (2006):

IBM Watson's supercomputer, named after the company's first CEO, was developed with the ability to answer questions posed in natural language. In 2011, computers competed on the game show Jeopardy beating former winners Brad Rutter and Ken Jennings. To this day, Watson is supplied to countless businesses across different industries

i. Siri (2010), Google Now (2012), Alexa (2015), Cortana (2015):

The early years of the 21st century have witnessed the birth of a series of virtual assistants that are big in the technology industry such as:

- ***Siri:*** The intelligent personal voice assistant of Apple, is developed by Siri Inc. and released as a standalone application. It was the first modern digital virtual assistant installed on a smartphone, introduced later as a feature of the iPhone 4S on 4 October 2011. After being acquired by Apple that year, the program was integrated into iOS, with the ability to interact with some of Apple's default apps. Today Siri can be used on apps in Apple's iOS, watchOS, tvOS, and Mac OS.
- ***Google Now:*** The intelligent personal assistant of Google, is released for Android. Using a natural language user interface, bots can answer questions, make recommendations, and perform actions on various web services. In 2016, an evolved version of Google Now with the ability to engage in a two-way dialogue, called the Google Assistant, was announced.
- ***Alexa:*** is an intelligent personal assistant developed by Amazon. It was introduced in 2014 and is now integrated with devices like Amazon Echo, Echo Dot, Echo Show and many more. All you have to do is say "Alexa, play some music" or "Alexa, find me an Italian restaurant" and it will help you. Just using the sound of your voice can search the web, play music, create to-do or shopping lists, set alarms, give audiobooks, get news or weather reports, control home products your smarts and more
- ***Cortana of Microsoft:*** Similar to Apple's Siri, Microsoft released their own smart personal processor. Cortana is available in languages whose features serve as a key component of the Microsoft operating system "makeover"

2.2.3. Method of interaction:

Virtual assistants work via:

- Text, including: online chat (especially in an instant messaging application or other application), SMS text, e-mail or other text-based communication channel, for instance Conversica's intelligent virtual assistants for business.
- Voice, for example with Amazon Alexa on the Amazon Echo device
- By taking or uploading images, as in the case of Samsung Bixby on the Samsung Galaxy S8

Some virtual assistants are accessible via multiple methods, such as Google Assistant via chat on the Google Allo and Google Messages app and via voice on Google Home smart speakers.

Virtual assistants use natural language processing (NLP) to match user text or voice input to executable commands. Many continually learn using artificial intelligence techniques including machine learning. Some of these assistants like Google Assistant(which contains Google Lens) and Samsung Bixby also have the added ability to do image processing to recognize objects in the image to help the users get better results from the clicked images.

2.2.4. Advantages of having virtual assistant:

Outstanding features of virtual assistants:

- Speech recognition and analysis
- Excellent reading comprehension skill
- Optimal information search function
- Self-learning ability from the users
- The feature of linking information from vary sources
- Role of virtual assistant in smart home

Virtual assistants can provide a wide variety of services. These include:

- Provide information such as weather, facts from e.g. Wikipedia or IMDb, set an alarm, make to-do lists and shopping lists.

- Play music from streaming services such as Spotify and Pandora; play radio stations; read audiobooks
- Play videos, TV shows or movies on televisions, streaming from e.g. Netflix
- Conversational commerce.
- Assist public interactions with government.
- Complement and replace customer service by humans. One report estimated that an automated online assistant produced a 30% decrease in the work-load for a human-provided call center.

Chapter 3: PREPARING FOR BUIDING A VIRTUAL ASSISTANT BY PYTHON

3.1. The environment and programming language:

- ***System requirements*:** Python 3.7.6, Visual Studio, Jupiter notebook, Anaconda
- ***Environment programming*:** Visual Studio
- ***Programming language:*** Python

Virtual assistant BEE is programmed based on Python programming language. Python was chosen because it is a programming language with a general-purpose library, so artificial intelligence is now simpler.

Python is a general purpose high-level programming language, created by Guido van Rossum and first released in 1991. Python is designed with the strong advantage of being easy to read, learn, and remember. Python is a language with a very bright appearance, clear structure, convenient for beginners and easy to learn; widely used in the development of artificial intelligence

Python's extensive set of standard libraries is one of Python's greatest strengths, providing the right tools for a variety of jobs. It is also the explanation for choosing Python to construct a personal assistant.

- ***Support tool***: Anaconda

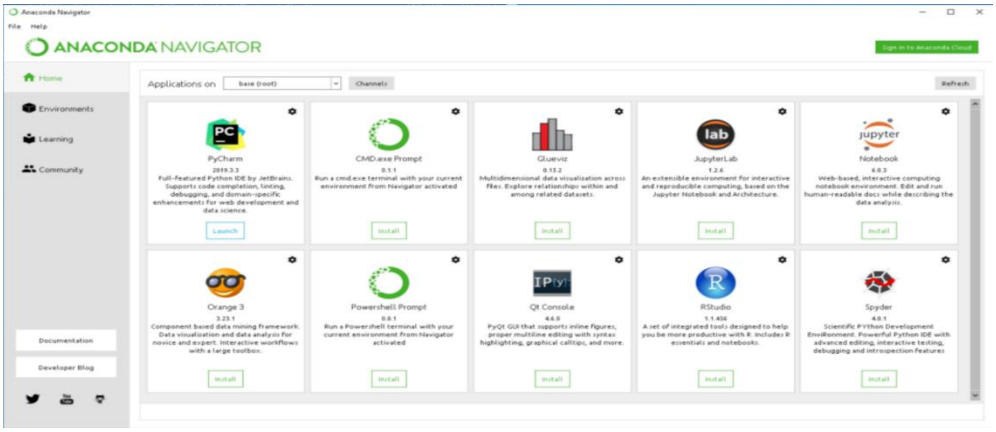


Figure 1: Illustration for Anaconda

Anaconda is a free and open source distribution of Python and R. It simplifies the installation, management and deployment of packages, such as numpy, scipy, tensorflow,...). Anaconda serves many purposes, especially in data science, machine learning, big data or image processing.

After completing the installation of Anaconda, we open Anaconda Prompt (which can be found in Anaconda installation folder) and move to the folder that contains project to run Jupiter Notebook by this commands: Jupiter notebook

That means we finished preparing progress about the environment and language for BEE.

3.2. Modules needed for building BEE – virtual assistant:

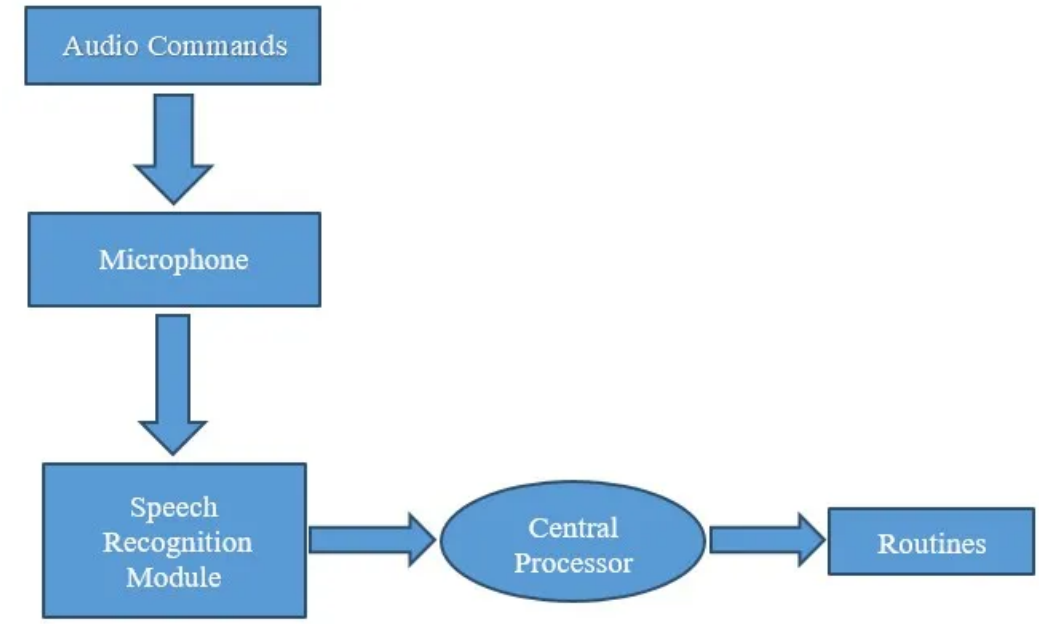


Figure 2: The basic workflow of voice assistant

The workflow of the basic process of the voice assistant is that speech recognition is used to convert the speech input to text. This text is then fed to the central processor which determines the nature of the command and calls the relevant script for execution.

a. <u>Python Text To Speech Platform X (pyttsx3):</u>

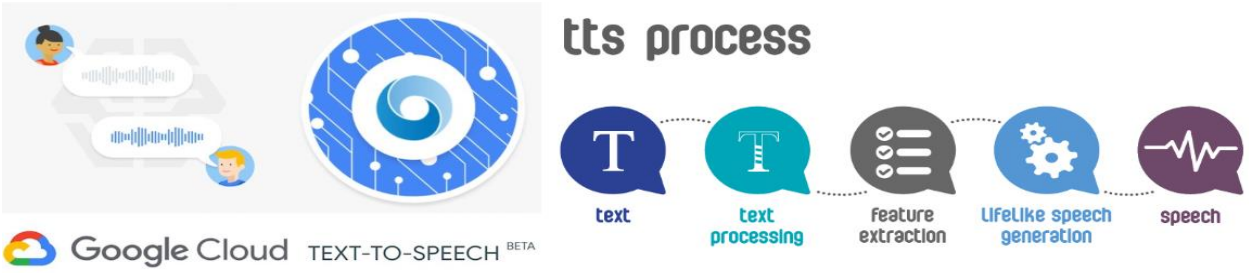


Figure 3: The process of text to speech in virtual assistant

Our assistant will need to convert users voiced question to a text one. And then, once the assistant looks up an answer online, it will need to convert the response into a voiceable phrase. For this purpose, pyttsx3 module was used. Pyttsx is a cross-platform text to speech library which is platform independent. The major advantage of using this library for text-to-speech conversion is that it works offline. To install this module type the below command in the terminal.

```
import pyttsx3
```

Figure 4: Code to install Text-to-speech module

b. <u>SpeechRecognition:</u>

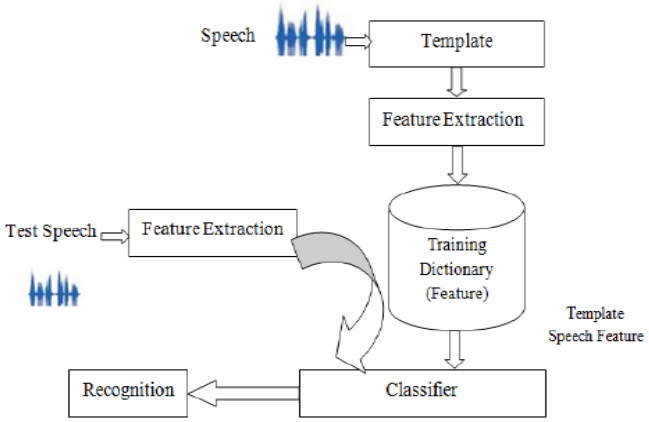


Figure 5: General process of recognizing speech

When users voice a question, BEE needs something that can capture it. The SpeechRecognition package allows Python to access audio from the machine's microphone, transcribe audio, save audio to an audio file, and other similar tasks. Besides, it allow BEE to convert audio into text for further processing. To install this module type the below command in the terminal.

```
import speech_recognition as sr
```

Figure 6: Code to install Speech Recognition module

c. Webbrowser:

As one of the functions of BEE is searching the particular information on the web, it means that, BEE needs a module which allows to tackle this problem. Webbrowser package in Python is the great solution. It provides a high-level interface which allows displaying Web-based documents to users. To install this module type the below command in the terminal.

```
import webbrowser as wb
```

Figure 7: Code to install Webbrowse module

d. Wikipedia:

It is used to fetch a variety of information from the Wikipedia website. To install this module type the below command in the terminal.

```
import wikipedia
```

Figure 8: Code to install Wikipedia module

Chapter 4: CONSTRUCTING THE SYSTEM OF BEE

BEE is programmed to be able to perform some simple functions such as: say hello, weather forecast,... And the installation of these functions, this function of BEE will be explained more clearly below with methods used for Virtual Assistant.

4.1. Speak Method:

Speak Method will help BEE in taking the voice from the machine. Here is the code explanation of Speak Method:

```
bee = pyttsx3.init()
#gets the current value of 'bee' property
voice = bee.getProperty('voices')
#[1] = female voice
bee.setProperty('voice', voice[1].id)

#function for the speaking of the assistant
def speak(audio):
    print('Bee: ' + audio)
    bee.say(audio)
    bee.runAndWait()
```

Figure 9: Code to install Speak Method

4.2. Take Query Method:

This method will check for the condition. If the condition is true it will return output. We can add any number if conditions for it and if the condition satisfy we will get the desired output.

```
welcome()
while True:
    query = command().lower()
    #identify each command separately
    if 'time' in query:
        time()
        speak('Anything else?')
    elif 'google' in query:
        google()
        speak('Anything else?')
    elif 'youtube' in query:
        youtube()
        speak('Anything else?')
    elif 'wikipedia' in query:
        wiki()
        speak('Anything else?')
    elif 'quit' or 'bye' in query:
        stop()
    else:
        speak("Sory I can't hear you. Try typing the command.")
        query=str(input('Your order is: '))
```

Figure 10: Code to install Take Query Method

4.3. Take Command Method:

This method is for taking the commands and recognizing the command from the speech_Recognition module:

```
#function for getting and returning a query of user
def command():
    #getting the command from user
    c = sr.Recognizer()
    #using default microphone
    with sr.Microphone() as source:
        audio=c.listen(source)
    try:
        #using voice recognition of google and setting the language to English
        query=c.recognize_google(audio,language='en')
        #show the subtitle
        print("You :" + query)
        #in case the assistant can't hear what user says
    except sr.UnknownValueError:
        speak("Sory I can't hear you. Try typing the command.")
        #type the command using keyboard
        query=str(input('Your order is: '))
    return query
```

Figure 11: Code to install Take Command Method

4.4. Notice Time Method:

This method is for getting the current time

```
def time():
    Time=datetime.datetime.now().strftime("%I %M:%p")
    speak("Its " + Time)
```

Figure 12: Code to install Notice Time Method

4.5. Hello Method:

This is just used to greet the user with a hello message and query the request of user

```
def welcome():
    hour=datetime.datetime.now().hour
    if hour >= 6 and hour < 12:
        speak ("Good morning")
    elif hour >= 12 and hour < 18:
        speak ("Good afternoon")
    if hour >= 6 and hour < 12:
        speak ("Good evening")
    speak('How can I help you?')
```

Figure 13: Code to install Hello Method

4.6. Google Method:

This method is for searching the information on Google platform based on the keyword was given by user

```
def google():
    speak('What do you want to search?')
    #getting what keyword user want to search on google
    #changing the command to lower case will make it easier for the assistant to handle
    search = command().lower()
    #getting the link and open it by the default webbrowser on user's device
    url = f"https://google.com/search?q={search}"
    wb.get().open(url)
    speak(f'Here is your {search} on google')
```

Figure 14: Code to install Google Method

4.7. Wikipedia Method and YouTube Method:

These methods are just for retrieving information on Wikipedia page and open the specified video on YouTube. Obviously, the research results depend on the keyword that human want to search

```
def youtube():
    speak('What video do you want to play?')
    search = command().lower()
    url = f"https://youtube.com/search?q={search}"
    wb.get().open(url)
    speak(f'Here is your {search} on youtube')

#function for some short information of something on wikipedia
def wiki():
    speak("What do you want to know?")
    text = command().lower()
    #a list of content that wikipedia can find and use the first paragraph
    contents = wikipedia.summary(text).split('\n')
    speak(contents[0])
    time.sleep(3)
```

Figure 15: Code to install YouTube Method and Wikipedia Method

4.8. Close Method:

This is just assist BEE to close after the signal given by human through term “Bye” or “Quit”

```
#close the assistant
def stop():
    speak("Goodbye, see you later!")
    quit()
```

Figure 16: Code to install Close Method

Chapter 5: RESULT

5.1. Completed Code:

```
import pyttsx3
import datetime
import speech_recognition as sr
import wikipedia
import webbrowser as wb

bee = pyttsx3.init()
voice = bee.getProperty('voices')
bee.setProperty('voice', voice[1].id)

#function for the speaking of the assistant
def speak(audio):
    print('Bee: ' + audio)
    bee.say(audio)
    bee.runAndWait()
```

```
#say hi to users depending on specific time of a day
def welcome():
    hour=datetime.datetime.now().hour
    if hour >= 6 and hour < 12:
        speak ("Good morning")
    elif hour >= 12 and hour < 18:
        speak ("Good afternoon")
    if hour >= 6 and hour < 12:
        speak ("Good evening")
    speak('How can I help you?')

#close the assistant
def stop():
    speak("Goodbye, see you later!")
    quit()
```

```python
#function for getting and returning a query of user
def command():
    c = sr.Recognizer()
    with sr.Microphone() as source:
        audio=c.listen(source)
    try:
        query=c.recognize_google(audio,language='en')
        print("You :" + query)
    except sr.UnknownValueError:
        speak("Sory I can't hear you. Try typing the command.")
        query=str(input('Your order is: '))
    return query

#getting current time
def time():
    Time=datetime.datetime.now().strftime("%I %M:%p")
    speak("Its " + Time)
```

```python
#function for searching information using google
def google():
    speak('What do you want to search?')
    search = command().lower()
    url = f"https://google.com/search?q={search}"
    wb.get().open(url)
    speak(f'Here is your {search} on google')

#finding videos on youtube
def youtube():
    speak('What video do you want to play?')
    search = command().lower()
    url = f"https://youtube.com/search?q={search}"
    wb.get().open(url)
    speak(f'Here is your {search} on youtube')

#function for some short information of something on wikipedia
def wiki():
    speak("What do you want to know?")
    text = command().lower()
    contents = wikipedia.summary(text).split('\n')
    speak(contents[0])
    time.sleep(3)
```

```
welcome()
while True:
    query = command().lower()
    if 'time' in query:
        time()
        speak('Anything else?')
    elif 'google' in query:
        google()
        speak('Anything else?')
    elif 'youtube' in query:
        youtube()
        speak('Anything else?')
    elif 'wikipedia' in query:
        wiki()
        speak('Anything else?')
    elif 'quit' or 'bye' in query:
        stop()
    else:
        speak("Sory I can't hear you. Try typing the command.")
        query=str(input('Your order is: '))
```

Figure 17: Completed code of BEE

5.2. Performance results of functions:

- BEE can getting in the requirements of users by voice with default microphone and returning a query of user. If BEE can not hear or recognize what user said, the output must be “Sorry I can’t hear you. Try typing the command.” And take the input by keyboard.
- Welcome the user: BEE say hi to users by one of these phrases “Good morning”, “Good afternoon” or “Good evening”, it depends on specific time that the user summon the presence of BEE. Then, BEE asks the commands of the user with “How can I help you?”
- Notice the exact time at the moment that user request to ask, with the keyword “time”, below the form: “Its + hh:mm AM (PM)” (For example: 22:00 PM)

- Searching data on the Google platform: When user confirms the term “google” BEE gets the keyword that user want to search on google through the audio question “What do you want to search?” and open the default web browser to search with that keyword. More further, if the operation is success, BEE sends a signal to the users that “Here is your {keyword} on google”
- Retrieve the information from Wikipedia: BEE was activated by the word “Wikipedia”. The operation is similar to searching the information in google. BEE takes the input as keyword, use that keyword to retrieve a list of content that Wikipedia can find, default page that BEE opens must be the first one on the list. The information user want would be read out loud by BEE
- Play video on YouTube: This function works as retrieving information on web and Wikipedia. However, instead of “google” or “Wikipedia”, “YouTube” would be the term to activated this feature.
- Stop the process: If in the voice demand that BEE received contain the term “quit” or “bye”, BEE might aware to stop serve user and say bye with “Goodbye, see you later”
- After one command was completely executed, BEE would actively asked for another requirement “Anything else?”

Chapter 6: CONCLUSION AND DEVELOPMENT

6.1. Conclusion:

After learning and implementing the topic of "Building a virtual assistant by Python", our group has constructed BEE, a plain personal assistant according to the initial basic requirements.

In the process of implementing the project and learning the theory as well as the application of artificial intelligence in creating a virtual assistant, we obtained the initial following results:

- Understand the foundation of artificial intelligence and virtual assistants
- Learn about the new programming language Python and its related environments and patterns to constructing a virtual assistants
- Understand the steps to build an application of artificial intelligence, which is BEE – a personal assistant.
- Successfully built a virtual assistant called BEE, whose functionalities include: Interact directly to human, introduction about itself, date and time announcement, searching specified information on the web and retrieving the information from Wikipedia or open the videos on YouTube platform.

As a result, BEE has met the requirements set out to perform the simple roles of a personal virtual assistant. However, there are still many limitations. The most important barrier is that BEE can only recognize the voice commands if English pronunciation is nearly perfect. This problem makes it difficult for users to make conversations to BEE. Besides, the model of BEE is still quite rudimentary, there are many other functions of personal assistants that BEE has not Been able to perform like weather forecasting or take note and so forth.

6.2. Development:

Within the scope of the project, our team was only able to perform some of the basic functions of a virtual assistant. Therefore, if there is an opportunity to do and study more deeply on how to build an artificial intelligence, it is certain that we will develop the project in the direction of cross-functionality and integrate the virtual assistant with other platforms, so that the applicability of BEE will be expanded.

REFERENCES

[1] https://www.diva-portal.org/smash/get/diva2:531796/FULLTEXT02.pdf

[2] https://www.activestate.com/blog/how-to-build-a-digital-virtual-assistant-in-python/

[3] file:///C:/Users/ACER/Downloads/JARVIS-_-Report-_2_.pdf

[4] https://en.wikipedia.org/wiki/Virtual_assistant#Experimental_decades:_1910s%E2%80%931980s

[5] https://olli.vn/blogs/alls/tro-ly-ao-la-gi

[6] https://extrudesign.com/virtual-assistant-using-python/

[7] https://codelearn.io/sharing/lap-trinh-tro-ly-ao-tieng-viet-python

[8] https://www.geeksforgeeks.org/build-a-virtual-assistant-using-python/