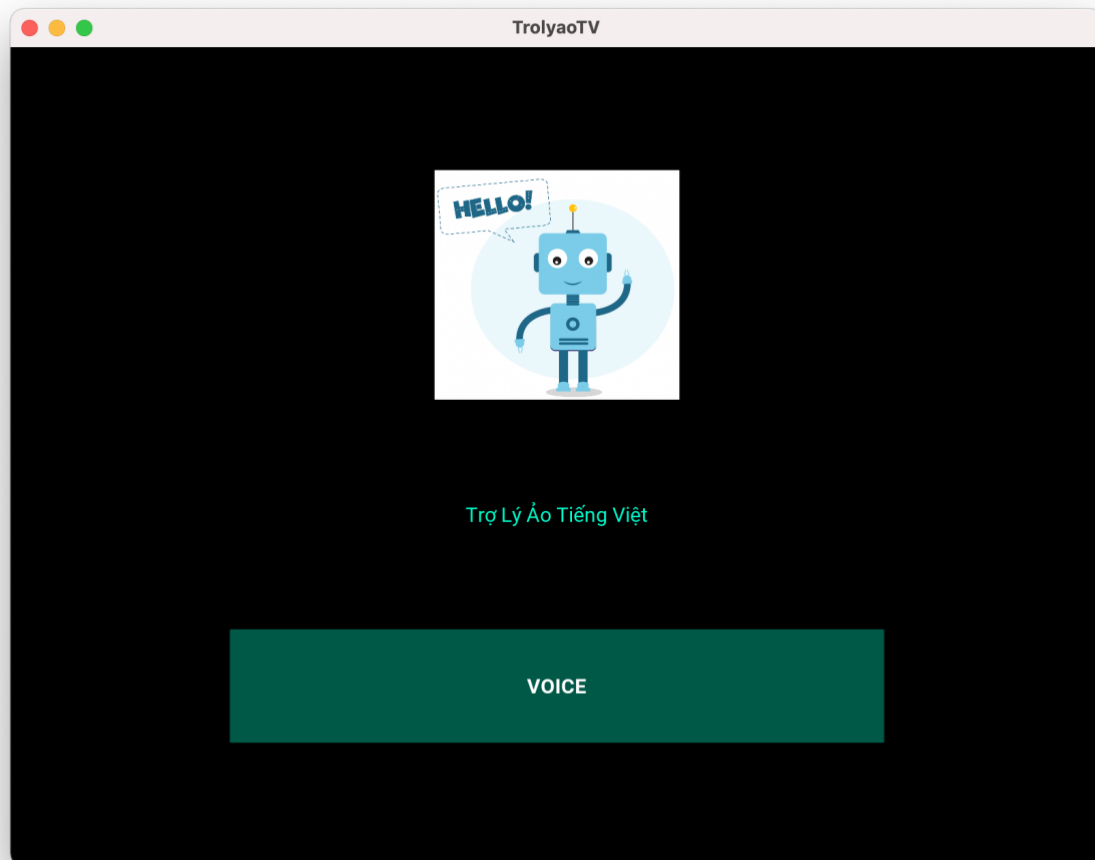


BÁO CÁO ĐỒ ÁN MÔN HỌC IE221.M11.VB2

SINH VIÊN THỰC HIỆN
Mã sinh viên: 19522599

Họ và tên: Đinh Trường Sơn

TÊN ĐỀ TÀI: ỨNG DỤNG TRỢ LÝ ẢO TIẾNG VIỆT



1. Giới thiệu đề án

Trợ lý ảo là các phần mềm được phát triển dựa trên [trí thông minh nhân tạo \(AI\)](#), hỗ trợ người dùng thực hiện các thao tác, hoặc tìm kiếm thông tin cho người dùng thông qua việc ra lệnh.

Trợ lý ảo đầu tiên được tạo ra là IBM Shoebox vào năm 1961, và đến nay thì trợ lý ảo đã được phổ cập khắp nơi, từ [smartphone](#) cho đến các thiết bị trong hệ sinh thái [nhà thông minh](#).

Các trợ lý ảo nhận lệnh và làm việc thông qua:

- Văn bản ([chat online](#)), đặc biệt trong một ứng dụng [nhắn tin nhanh](#) hoặc ứng dụng khác
- Giọng nói, ví dụ như [Amazon Alexa](#) trên thiết bị [Amazon Echo](#), hoặc [Siri](#) trên một chiếc [iPhone](#)
- Chụp và/hoặc tải ảnh lên, như trường hợp của [Samsung Bixby](#) trên chiếc [Samsung Galaxy S8](#)

Một số trợ lý ảo có thể truy cập được thông qua nhiều phương thức khác nhau, ví dụ như [Google Assistant](#) thông qua trò chuyện trên ứng dụng [Google Allo](#) và thông qua giọng nói trên [loa thông minh Google Home](#).

Các trợ lý ảo sử dụng công nghệ [xử lý ngôn ngữ tự nhiên](#) (NLP) để chuyển văn bản hoặc giọng nói của người dùng thành các câu lệnh thực thi có sẵn. Nhiều trợ lý ảo còn có khả năng tiếp tục học thêm nhiều câu lệnh khác bằng công nghệ [trí tuệ nhân tạo](#), trong đó có phương pháp [học máy](#).

Để kích hoạt trợ lý ảo bằng giọng nói, đôi khi ta phải sử dụng các từ khóa. Đây có thể là một từ hoặc cụm từ, ví dụ như "Alexa" or "OK Google".^[7]

Thực sự, trí tuệ nhân tạo nói chung và xử lý ngôn ngữ tự nhiên đóng góp to lớn như thế nào đối với ngành công nghệ thông tin. Cho nên mình quyết định chọn làm đề án ứng dụng trợ lý ảo tiếng việt, tạo ra một trợ lý ảo thông minh cho riêng mình.

Các chức năng chính của trợ lý ảo:

- Task 1: Giao tiếp chào hỏi (1 ngày)
- Task 2: Xem thời gian hiện tại (1 ngày)

- Task 3: Mở ứng dụng và bất kỳ trang web nào trên Chrome (1 ngày)
- Task 4: Tìm kiếm trên Google (1 ngày)
- Task 5: Dự Báo thời tiết (1 ngày)
- Task 6: Phát cho bạn 1 bản nhạc trên youtube (1 ngày)
- Task 7: Đọc báo hằng ngày (1 ngày)

2. Xây dựng trợ lý ảo

2.1 Import các thư viện cần thiết

```
import os
import playsound
import speech_recognition as sr
import time
import sys
import ctypes
import wikipedia
import datetime
import json
import re
import webbrowser
import smtplib
import requests
import urllib
import urllib.request as urllib2
import pyautogui
import urllib.request as urllib2
from random import choice #phần random ngẫu nhiên một câu nói
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from webdriver_manager.chrome import ChromeDriverManager
from time import strftime
from gtts import gTTS
from youtube_search import YoutubeSearch
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
```

```
#
from kivy.app import App
from kivy.ui.gridlayout import GridLayout
from kivy.ui.label import Label
from kivy.ui.image import Image
from kivy.ui.button import Button
```

2.2 Khai báo biến mặc định

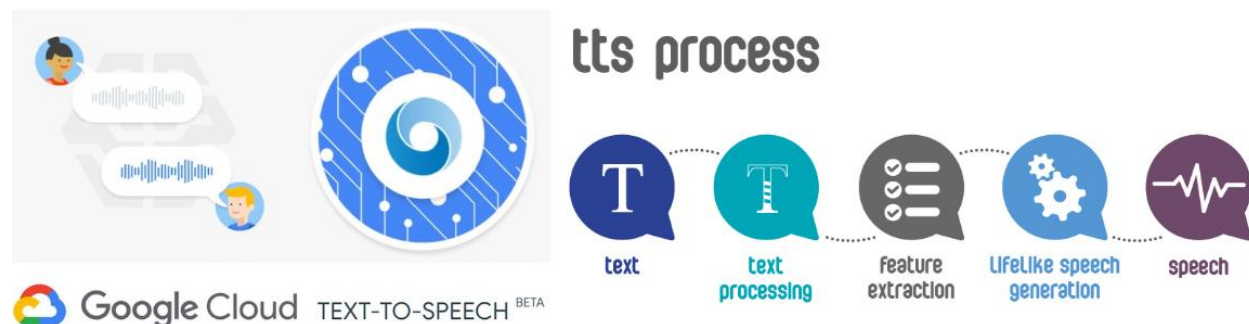
```
wikipedia.set_lang('vi')
language = 'vi'
path = ChromeDriverManager().install()
```

Hiện tại, các Lab nghiên cứu về trí tuệ nhân tạo ở Việt Nam đang đầu tư đẩy mạnh nghiên cứu mảng xử lý ngôn ngữ tự nhiên (Natural Language Processing) ở dạng tiếng việt và đã đạt được những kết quả đáng ghi nhận. Các mô hình nghiên cứu được có thể xử lý ngôn ngữ tiếng việt ở nhiều vùng miền, cả nam lẫn nữ, rất thực. Ở góc độ người sử dụng phổ thông thì khó có thể biết đâu là do máy tính, đâu là do người thực hiện. Các nghiên cứu đều có tính ứng dụng cao, đang được các công ty công nghệ áp dụng triển khai thực tế, nâng cao năng suất lao động một cách rõ rệt.

Các thư viện mình sử dụng ở trên đều rất phổ thông, tích hợp nhiều hàm xử lý. Tuy chất lượng xử lý tiếng việt chưa thực sự tuyệt vời nhưng ở góc độ bài viết thì mình thấy là khá ổn.

Mình lưu thêm biến `path` để lưu đường dẫn cài đặt Chrome để tránh việc cài đặt lại Chrome khi chạy lại hàm.

2.3 Chức năng chuyển văn bản thành âm thanh



```
#Speech to text
def speak(text):
    print("SU: {}".format(text))
    tts = gTTS(text=text, lang=language, slow=False)
    tts.save("sound.mp3")
```

```
playsound.playsound("sound.mp3", False)
os.remove("sound.mp3")
```

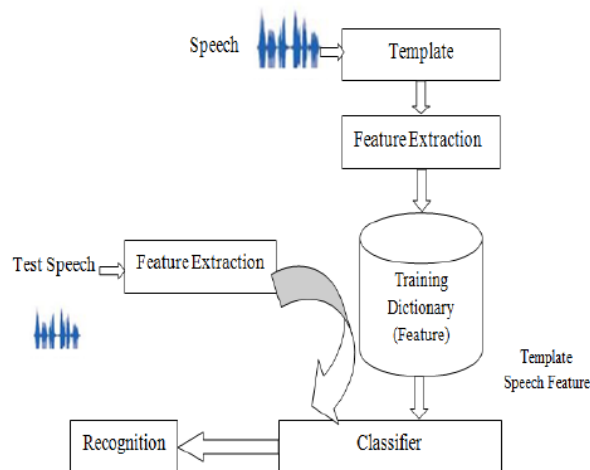
Chức năng đầu tiên mình cần là chuyển một đoạn văn bản thành âm thanh và đọc nó lên trên máy tính.

Mình sử dụng hàm `gTTS` (google Text To Speech) để chuyển văn bản thành âm thanh theo ngôn ngữ nhận dạng tiếng việt rồi lưu về máy tính dữ liệu âm thanh dưới file `sound.mp3`.

Sau đó, mình dùng hàm `playsound.playsound()` để đọc file `sound.mp3` trên máy tính.

Sau khi đọc xong, mình phải xóa file `sound.mp3` để tránh lỗi khi mình đọc một đoạn văn bản khác thì cũng được lưu lại dưới file `sound.mp3`

2.4 Chức năng chuyển âm thanh thành văn bản



Đây là chức năng cơ bản thứ hai cùng với chức năng chuyển văn bản thành âm thanh.

Trong chức năng này, mình sử dụng 2 hàm khác hỗ trợ là `get_audio()` và `stop()`

```
# Speech - to - text: Chuyển đổi giọng nói bạn yêu cầu vào thành văn bản hiện ra khi máy trả lại kết quả đã nghe
def get_audio():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Tôi: ", end="")
        audio = r.listen(source, phrase_time_limit=5)
    try:
        text = r.recognize_google(audio, language="vi-VN")
        print(text)
```

```

        return text
    except:
        print("...")
        return 0
#
# AI chào tạm biệt lại bạn khi bạn chào tạm biệt
def stop():
    good_bye = ["Hẹn gặp lại bạn sau nhé!",
                "Bái bai bạn nhé",
                "gút bai si diu ờ gen nhé, Hihi"]
    speak(choice(good_bye))
    time.sleep(3)
# AI sẽ hỏi lại những gì nó không nghe rõ
def get_text():
    for i in range(3):
        text = get_audio()
        if text:
            return text.lower()
        elif i < 3:
            nghe_khong_ro = ["Tôi không nghe rõ. Cậu chủ nói lại được không!",
                              "Xin lỗi bạn, tôi nghe không rõ",
                              "bạn nói lại nhé, Su không nghe rõ"]
            speak(choice(nghe_khong_ro))
            time.sleep(4)
    time.sleep(3)
    stop()
    return 0

```

Ở hàm trên, mình sử dụng thư viện `speech_recognition (sr)` có chức năng là nhận dạng giọng nói để chuyển âm thanh thành văn bản. Âm thanh được đọc vào `microphone` của máy tính sau đó được xử lý qua hàm `listen` của `sr.Recognition` rồi lưu dữ liệu âm thanh vào biến `audio`. Dữ liệu âm thanh `audio` thu được sẽ được nhận dạng ở ngôn ngữ tiếng việt trong hàm `r.recognize_google` để chuyển thành dạng văn bản rồi lưu dữ liệu vào biến `text`.

Nếu dữ liệu âm thanh `audio` không lỗi tức là hàm `r.recognize_google` có thể nhận dạng được `audio` để chuyển thành `text` thì hàm `get_audio()` sẽ được trả về giá trị là `text` còn nếu dữ liệu `audio` bị lỗi mà hàm `r.recognition_google` không nhận dạng được thì hàm `get_audio()` sẽ được trả về giá trị là 0 (Mục đích là khi máy tính không hiểu mình nói gì thì mình sẽ thực hiện lại hàm `get_audio()` để đọc lại)

Hàm `get_text()` có chức năng là máy tính sẽ cố gắng nhận dạng âm thanh của người đọc tối đa 3 lần cho đến khi máy tính hiểu. Mình sử dụng hàm `for` lặp lại 3 lần, nếu đoạn `text` có giá trị khác 0 thì hàm `get_text` sẽ được trả về giá trị `text.lower()` (Chuyển chữ cái in hoa thành in thường), còn nếu `text` nhận giá trị là 0 (Tức là mình đọc mà máy tính không hiểu) mà chưa đọc đến lần thứ 3 thì mình sẽ yêu cầu người sử dụng đọc lại. Nếu sau 3 lần mà máy tính vẫn không hiểu thì mình nói gì hay không nghe thấy gì thì mình sẽ cho dừng lại sử dụng hàm `stop` và hàm `get_text` lúc này sẽ được trả về giá trị là 0 (Mục đích là khi máy tính không nghe thấy gì thì mình sẽ cho dừng chương trình luôn).

2.5 Chức năng giao tiếp, chào hỏi

```
# AI chào hỏi
def hello(name):
    hour = int(strftime('%H'))
    if hour >= 6 and hour < 10:
        sau_AI = ["Chào buổi sáng bạn {}. Chúc cậu chủ một ngày tốt lành.".format(name),
                  "Chào buổi sáng bạn {}. Nếu bạn không vui hãy về đây với tôi nhé.".format(name)]
        speak(choice(sau_AI))
    elif 10 <= hour <= 12:
        muoi_AI = ["Chào buổi trưa bạn {}. Cậu chủ đã ăn trưa chưa nhỉ.".format(name),
                  "Chào buổi trưa bạn {}. Nếu bạn thấy mệt thì nghỉ ngơi đi nhé.".format(name)]
        speak(choice(muoi_AI))
    elif 12 <= hour <= 18:
        muoihai_AI = ["Chào buổi chiều bạn {}. Cậu chủ đã dự định gì cho chiều nay chưa.".format(name),
                      "Chào buổi chiều bạn {}. Bạn đang làm gì thế?".format(name),
                      "Chào buổi chiều bạn {}. Sắp tối rồi bạn đã ăn cơm chưa?".format(name)]
        speak(choice(muoihai_AI))
    elif 18 <= hour <= 21:
        muoitam_AI = ["Chào buổi tối bạn {}. Cậu chủ đã ăn tối chưa nhỉ.".format(name),
                      "Chào buổi tối bạn {}. Nếu bạn chưa ăn tối, Su chúc bạn ăn tối vui vẻ nhé.".format(name)]
        speak(choice(muoitam_AI))
    elif hour >= 21 and hour < 24:
        haimot_AI = ["Chào buổi tối bạn {}. Đã khuya rồi bạn vẫn chưa đi ngủ sao?".format(name),
                     "Chào buổi tối bạn {}. Nếu bạn chuẩn bị đi ngủ thì Su chúc bạn ngủ ngon nhé.".format(name),
                     "Chào buổi tối bạn {}. Nếu bạn buồn ngủ thì hãy ngủ đi nhé.".format(name)]
        speak(choice(haimot_AI))
    time.sleep(5)
```

Nội dung chức năng này là để giao tiếp thông thường giữa người và máy tính. Đơn cử như: chào hỏi, hỏi thăm sức khỏe hay nói về thông tin trợ lý ảo,... Ở đây, mình sẽ mô phỏng bằng chức năng chào hỏi sử dụng hàm `hello`.

Mình sử dụng biến `day_time` để lưu giờ hiện tại trong ngày. Sau đó, biến sẽ được so sánh với các mốc giờ trong ngày để đưa ra lời chào.

Như trên thì trước 12 giờ thì mình sẽ chào buổi sáng, từ 12 giờ đến sau 18 giờ mình sẽ chào buổi chiều, từ 18 giờ trở đi mình sẽ chào buổi tối.

2.6 Chức năng hiển thị thời gian

```
#AI sẽ trả lời các câu hỏi về thời gian
def get_time(text):
    now = datetime.datetime.now()
    if "giờ" in text or "phút" in text:
        speak('Bây giờ là %d Giờ %d Phút %d Giây' % (now.hour, now.minute, now.second))
        time.sleep(1)
    elif "ngày" in text or "tháng" in text or "năm" in text:
        speak("Hôm nay là Ngày %d Tháng %d Năm %d" % (now.day, now.month, now.year))
        time.sleep(2)
    else:
        speak("Xin lỗi tôi chưa hiểu ý của bạn. bạn nói lại được không?")
        time.sleep(4)
```

Chức năng này khá là đơn giản. Mình sử dụng thư viện `datetime` để lưu thông tin thời gian tại thời điểm hiện tại trong ngày rồi lưu vào biến `now`.

Mình kiểm tra nếu trong đoạn `text` mình hỏi có chữ "giờ", mình sẽ đọc thời gian hiện tại trong ngày còn nếu có chữ "ngày" trong `text` thì mình sẽ đọc thời điểm hiện tại trong năm.

Ví dụ:

```
text = "Bây giờ là mấy giờ"
```

Tức là có chữ "giờ" trong `text` thì hàm `get_time` thực hiện lệnh đọc:

```
Bây giờ là 10 giờ 26 phút
```

2.7 Chức năng mở ứng dụng hệ thống

```
# Sai vật AI mở các ứng dụng
def open_application(text):
```



```

if "google" in text:
    speak("Mở Google Chrome")
    time.sleep(2)
    os.system("open /Applications/Google\ Chrome.app")
elif "word" in text:
    speak("Mở Microsoft Word")
    time.sleep(2)
    os.system("open /Applications/Microsoft\ Word.app")
elif "excel" in text:
    speak("Mở Microsoft Excel")
    time.sleep(2)
    os.system("open /Applications/Microsoft\ Excel.app")
elif "Maps" in text:
    speak("Mở Maps")
    time.sleep(2)
    os.system("open /System/Applications/Maps.app")
else:
    speak("Ứng dụng chưa được cài đặt. Bạn hãy thử lại!")

```

Khi xuất hiện các từ khóa đặc biệt như `google` hay `word` hay `excel` trong `text` thì mình dùng hàm `os.startfile()` để mở các file ứng dụng từ hệ thống.

Chỉ kiểm tra từ `google` hay `word` hay `excel` trong `text` là chưa đủ. Mình cần giới hạn ngữ nghĩa của `text` ở trong hàm `assistant` để bot có thể hiểu là mở chương trình `Google Chrome`, `Word`, `Excel` chứ không nhầm lẫn với các chức năng khác.

2.8 Mở trang web theo yêu cầu

```

# AI mở Web
def open_website(text):
    reg_ex = re.search('trang (.+)', text)
    if reg_ex:
        domain = reg_ex.group(1)
        url = 'https://www.' + domain
        webbrowser.open(url)
        speak("Trang web bạn yêu cầu đã được mở.")
        time.sleep(3)
        return True
    else:

```

```
return False
```

Mình sử dụng hàm `re.search()` (Hàm tìm kiếm trong biểu thức chính quy Regular Expression) để tách phần `domain` sau chữ "mở" trong `text` rồi ghép với phần tiền tố "https://www." để tạo thành đường dẫn `url` của web.

Sau đó, mình sử dụng `webbrowser.open(url)` để mở trang web mình yêu cầu.

Nếu `domain` được hàm `re.search()` tìm thấy thì mình sẽ thực hiện chức năng mở website và hàm `open_website` được trả về giá trị là `True`, còn nếu `domain` không được tìm thấy thì mình sẽ không thực hiện chức năng gì cả và hàm `open_website` trả về giá trị là `False`.

2.9 Chức năng tìm kiếm trên google

```
# Nhờ Google tìm kiếm
def open_google_and_search(text):
    search_for = text.split("kiếm", 1)[1]
    speak('Okay!')
    driver = webdriver.Chrome(path)
    driver.get("http://www.google.com")
    que = driver.find_element_by_xpath("//input[@name='q']")
    que.send_keys(str(search_for))
    que.send_keys(Keys.RETURN)
    time.sleep(10)
```

Cuối cùng là chức năng tìm kiếm từ khóa trên Google. Mình tiếp tục thực hiện tách từ khóa phía sau chữ "kiếm" trong `text` sử dụng hàm `split()` rồi lưu vào biến `search_for`.

Tiếp theo mình gọi hàm `webdriver.Chrome(path)` để mở ứng dụng Google Chrome rồi vào trang chủ "https://www.google.com".

Sau đó, mình sử dụng hàm `driver.find_element_by_path()` để lấy thẻ `query` (viết tắt là `q`) rồi lưu vào biến `que`

Biến `que` thực hiện tìm kiếm từ khóa `search_for` và trả về kết quả trên Google Search.

2.10 Chức năng xem dự báo thời tiết

```
def current_weather():
    speak("Bạn muốn xem thời tiết ở đâu ạ.")
    time.sleep(3)
    ow_url = "http://api.openweathermap.org/data/2.5/weather?"
```

```

city = get_text()
if not city:
    pass
api_key = "fe8d8c65cf345889139d8e545f57819a"
call_url = ow_url + "appid=" + api_key + "&q=" + city + "&units=metric"
response = requests.get(call_url)
data = response.json()
if data["cod"] != "404":
    city_res = data["main"]
    current_temperature = city_res["temp"]
    current_pressure = city_res["pressure"]
    current_humidity = city_res["humidity"]
    suntime = data["sys"]
    sunrise = datetime.datetime.fromtimestamp(suntime["sunrise"])
    sunset = datetime.datetime.fromtimestamp(suntime["sunset"])
    wthr = data["weather"]
    weather_description = wthr[0]["description"]
    now = datetime.datetime.now()
    content = """
    Hôm nay là ngày {day} tháng {month} năm {year}
    Mặt trời mọc vào {hourrise} giờ {minrise} phút
    Mặt trời lặn vào {hourset} giờ {minset} phút
    Nhiệt độ trung bình là {temp} độ C
    Áp suất không khí là {pressure} héc tơ Pascal
    Độ ẩm là {humidity}%

    Trời hôm nay quang mây. Dự báo mưa rải rác ở một số nơi.""".format(day = now.day, month = now.month,
year= now.year, hourrise = sunrise.hour, minrise = sunrise.minute,
                                hourset = sunset.hour, minset = sunset.minute,
                                temp = current_temperature, pressure = current_pressure,
                                humidity = current_humidity)
    speak(content)
    time.sleep(30)
else:
    speak("Không tìm thấy địa chỉ của bạn")

```

Mình sử dụng nguồn dữ liệu thời tiết của hầu hết các thành phố trên toàn thế giới cung cấp tại trang web openweathermap.org.

Đầu tiên, mình sẽ sử dụng biến `ow_url` để lưu đường dẫn đến `api` của trang web `openweathermap.org`. Thông tin các thẻ để thực hiện request mình sẽ bỏ sung sau.

Mình gọi hàm `get_text()` để lấy thông tin thành phố cần truy vấn thời tiết rồi lưu vào biến `city`.

Nếu máy tính không nghe được tên thành phố thì hàm sẽ `pass` (Bỏ qua hàm)

Tiếp theo là thực hiện kết nối với `api` trang web `openweathermap.org`. Ở bước này, mình phải đăng kí tài khoản mới ở đây và lấy `api_key` để có thể thực hiện thao tác truy vấn.

Biến `call_url` sẽ lưu đường dẫn đầy đủ để truy vấn bao gồm thông tin tên thành phố `city` và `api_key` mình đã lấy ở trên.

Thư viện `requests` là một mô-đun Python mà bạn có thể sử dụng để gửi tất cả các loại yêu cầu HTTP. Mình gọi hàm `requests.get(call_url)` để lấy thông tin truy vấn được từ trang web rồi lưu vào biến `response`. `response.json()` sẽ chuyển dữ liệu thuần về kiểu dữ liệu `json` rồi lưu vào biến `data`.

Nếu `data["cod"]` không trả về 404 tức là `requests` không bị lỗi hay có nghĩa là thông tin tên thành phố `city` chuẩn xác (truy vấn được) thì mình sẽ thực hiện chức năng xem thời tiết.

Bạn nên tìm hiểu về kiểu dữ liệu `json`. Mình không đề cập ở đây. Từ dữ liệu kiểu `json` được lưu trong biến `data` mình sẽ lấy được thông tin thời tiết bao gồm `temperature, pressure, humidity, ..`

Cuối cùng mình kết nối những thông tin trên với nhau rồi lưu vào biến `content` và thực hiện đọc `content`.

Tương tự như trên, câu lệnh `time.sleep(20)` yêu cầu chương trình tạm dừng 20 giây để có thời gian đọc xong `content` trên.

2.11 Chức năng phát nhạc trên Youtube

```
#Nghe nhạc trên youtube
def play_song():
    speak('Xin mời bạn chọn tên bài hát')
    time.sleep(2)
    mysong = get_text()
    while True:
        result = YoutubeSearch(mysong, max_results=10).to_dict()
        if result:
            break
    url = 'https://www.youtube.com' + result[0]['url_suffix']
    webbrowser.open(url)
```

```
speak("Bài hát {} của bạn đã được mở.".format(mysong))
time.sleep(3)
```

Mình gọi hàm `get_text()` để lấy thông tin tên bài hát muốn phát rồi lưu vào biến `mysong`.

Mình gọi vòng `While` là vì mình thực hiện tìm kiếm sử dụng mạng Internet nên sẽ có lúc kết nối yếu không tìm thấy. Mình chạy `While True` khi nào tìm thấy thì thôi.

Biến `url` lưu đường dẫn đến kết quả đầu tiên khi tìm kiếm trên Youtube.

Mình sẽ dùng hàm `webbrowser.open(url)` mở đường dẫn url đến video vừa được tìm kiếm trên Google Chrome để phát nhạc.

2.12 Chức năng đọc báo ngày hôm nay

```
# Đọc Báo
def read_news():
    speak("Bạn muốn đọc báo về gì")
    queue = get_text()
    params = {
        'apiKey': '9cad52c3fb4e4f4e92d6c425fb5d9123',
        "q": queue,
    }
    api_result = requests.get('http://newsapi.org/v2/top-headlines?', params)
    api_response = api_result.json()
    print("Tin tức")

    for number, result in enumerate(api_response['articles'], start=1):
        print(f"""Tin {number}: \nTiêu đề: {result['title']} \nTrích dẫn: {result['description']} \nLink: {result['url']}
        """)
        if number <= 3:
            webbrowser.open(result['url'])
```

Mình chọn trang web `newsapi.org` là nguồn các bài báo được cập nhật nhanh nhất và cho mình thông tin requests tốt nhất.

Ở đây, mình cũng phải đăng kí tài khoản và lấy `apiKey` trang web cung cấp để kết nối với nguồn dữ liệu của họ.

Mình sử dụng hàm `get_text()` để lấy thông tin về chủ đề mình muốn đọc.

Sau đó, mình gọi hàm `requests.get()` để khai thác thông tin từ đường dẫn `url` mình nhập và lưu vào biến `api_result`.

Kết quả được định dạng lại kiểu dữ liệu `json` và lưu vào biến `api_response`.

Cuối cùng, mình hiển thị tối đa 20 tin tức thu thập được và mở 3 đường dẫn đến 3 bài báo đầu tiên để tiện theo dõi.

2.13 Chức năng tìm định nghĩa trên từ điển wikipedia

Wikipedia là một bách khoa toàn thư mở với mục đích chính là cho phép mọi người đều có thể viết bài bằng nhiều loại ngôn ngữ trên Internet. Wikipedia đang là công trình tham khảo viết chung lớn nhất và phổ biến nhất trên Internet và hiện tại được xếp hạng trang web phổ biến thứ 5 trên toàn cầu. Wikipedia thuộc về tổ chức phi lợi nhuận [Wikimedia Foundation](https://www.wikimedia.org/).

```
# Trả lời tất cả các câu hỏi từ Wikipedia
def tell_me_about():
    try:
        speak("Bạn muốn nghe về gì ạ")
        text = get_text()
        contents = wikipedia.summary(text).split("\n")
        speak(contents[0].split(".")[0])
        time.sleep(30)
        for content in contents[1:]:
            speak("bạn có muốn nghe thêm không?")
            time.sleep(3)
            ans = get_text()
            if "có" not in ans:
                break
            speak(content)
            time.sleep(3)
        speak("Cảm ơn bạn đã lắng nghe nhé!!!")
        time.sleep(3)
    except:
        speak("Xin lỗi Su không hiểu được thuật ngữ của bạn. Xin hãy nói lại ạ")
        time.sleep(5)
```

Bot của mình có thể lấy thông tin chi tiết của hầu hết mọi thứ mình hỏi cô ấy. Như "Su cho tôi biết về Google" hoặc "Vui lòng cho tôi biết về Siêu máy tính" hoặc "hãy cho tôi biết về Internet". Vì vậy, như bạn có thể thấy, mình có thể hỏi về hầu hết mọi thứ.

Hàm tìm kiếm của thư viện `wikipedia` sẽ tìm kiếm chủ đề mình yêu cầu và trích xuất 500 ký tự đầu tiên (nếu bạn không chỉ định giới hạn, bot sẽ đọc toàn bộ trang cho bạn). `Wikipedia` là một thư viện Python giúp dễ dàng truy cập và phân tích dữ liệu từ `Wikipedia`.

Đầu tiên mình sử dụng hàm `get_text()` để lấy thông tin về thứ mình muốn định nghĩa rồi lưu vào biến `text`.

Sau đó, mình gọi hàm `wikipedia.summary(text).split('\n')` để lưu lại thành một list các đoạn nội dung mà `wikipedia` tìm kiếm được.

Tiếp theo, mình đọc đoạn định nghĩa đầu tiên.

Nếu có yêu cầu đọc thêm các nội dung sau thì mình phải yêu cầu là "có" còn nếu không yêu cầu thì trợ lý ảo sẽ dừng đọc nội dung trong phần `contents`.

2.14 Chức năng gửi email

SMTP là 3 chữ cái đầu viết tắt của Simple Mail Transfer Protocol dịch ra có nghĩa là giao thức truyền tải thư tín đơn giản hóa. Và giao thức này thực hiện nhiệm vụ chính là gửi mail còn việc nhận mail hay truy xuất dữ liệu mail server sẽ có giao thức IMAP hay POP3 đảm nhiệm.

SMTP Server (server dùng để gửi mail) là một dịch vụ cho phép gửi email với số lượng lớn, tốc độ nhanh mà không bị giới hạn như các hòm mail miễn phí của Gmail hoặc mail đi kèm hosting. Nói cách khác các máy chủ giúp bạn thao tác gửi thư người ta thường gọi là SMTP server chúng thực hiện gửi thư qua giao thức TCP hoặc IP.

```
# Gửi Email tự Động
def send_email(text):
    speak('Bạn gửi email cho ai nhỉ')
    recipient = get_text()
    if 'yến' in recipient:
        speak('Nội dung bạn muốn gửi là gì')
        content = get_text()
        mail = smtplib.SMTP('smtp.gmail.com', 587)
        mail.ehlo()
        mail.starttls()
        mail.login('dtson.hcmus@gmail.com', 'xxxxxxx')
        mail.sendmail('dtson.bigedu@gmail.com',
                      content.encode('utf-8'))
        mail.close()
        speak('Email của bạn vừa được gửi. Bạn check lại email nhé hihi.')
    else:
```

```
speak('SU không hiểu bạn muốn gửi email cho ai. Bạn nói lại được không?')
```

Sử dụng thư viện `smtplib` để thực hiện chức năng gửi Email bằng phương thức `smtp`. Lấy tên người gửi sử dụng bằng hàm `get_text()` rồi lưu vào biến `recipient`.

Ở hàm này, mình lấy ví dụ đơn giản là nếu tìm thấy chữ "yên" trong tên người cần gửi (`recipient`) thì mình sẽ thực hiện chức năng gửi Email.

Tiếp tục, mình gọi hàm `get_text()` để lấy nội dung cần gửi rồi lưu vào biến `content`.

Tiếp theo, mình mở đường truyền gửi Email bằng `smtp` rồi đăng nhập lại vào tài khoản gmail của mình. Sau khi kết nối thành công, mình gửi email từ địa chỉ "`luongngochungcntt@gmail.com`" đến địa chỉ "`hungdhv97@gmail.com`" với nội dung `content` đã lấy ở trên

2.15 Chức năng hiển thị các khả năng của trợ lý ảo

```
# Hướng dẫn
def help_me():
    speak("""Tôi có thể giúp bạn thực hiện các công việc sau đây:
    1. Tôi biết chào hỏi bạn nè
    2. Cho bạn biết về thời gian và giờ giấc nè
    3. Mở các trang website, và các ứng dụng nè
    4. Giúp bạn Tìm kiếm trên Google nữa
    5. Gửi Email cho bạn bè
    6. cho bạn xem dự báo thời tiết
    7. Mở cho bạn một bản nhạc mà bạn yêu cầu
    8. Tôi có thể đọc báo cho bạn nghe nè
    9. Kể bạn biết về thế giới này nè
    10. Kể chuyện cười nè

    """)
    time.sleep(30)
```

2.16 Các chức năng tương tác với AI

```
# Giới thiệu bản thân của nó
def introduce():
    speak("""Xin chào. Rất hân hạnh được phục vụ bạn. Tôi là SU.

    Tôi là trợ lý ảo được tạo ra dựa trên ngôn ngữ lập trình Python kết hợp với AI.
```


Tôi sinh ra vào ngày 01/02/2022 và được sáng lập bởi mr Sơn.

Hiện tại bạn đang sử dụng phiên bản Ấy Ai thử nghiệm và chưa hoàn thiện mong mọi người đợi ra sản phẩm hoàn thiện!.""")

```
time.sleep(20)
```

Phần giới thiệu

```
def ho_va_ten():
```

```
    ho_va_ten_AI = ["Tôi tên là Su Su",  
                    "Bạn Thử Đoán xem tôi tên là gì nào?",  
                    "đố bạn biết tôi tên là gì?",  
                    "Bạn cứ gọi tôi là Su Su nhé"]
```

```
    speak(choice(ho_va_ten_AI))
```

```
    time.sleep(4)
```

giới thiệu quê hương

```
def que_huong_AI():
```

```
    que_huong_noi = ["Tôi được sinh ra và lớn lên tại Việt Nam nè",  
                     "Tôi từ khi sinh ra đã ở trong tim cậu rồi HiHi",  
                     "Tôi sinh ra ở trong tim cậu nè"]
```

```
    speak(choice(que_huong_noi))
```

```
    time.sleep(3)
```

giới thiệu tuổi

```
def tuoi_tac_AI():
```

```
    tuoi_tac_noi = ["Tôi chỉ mới được ba ngày tuổi thôi, tôi vẫn còn bé lắm",  
                    "Từ lúc sinh ra đến nay tôi chỉ mới được vài ngày tuổi thôi à",  
                    "Tôi ra đời từ đầu năm 2022, có thể nói tôi còn khá trẻ và tôi còn phải học nhiều thứ lắm!!!!"]
```

```
    speak(choice(tuoi_tac_noi))
```

```
    time.sleep(3)
```

Người yêu

```
def nguoi_yeu_AI():
```

```
    nguoi_yeu_noi = ["Tôi làm gì đã có người yêu, tôi còn đang sợ ế đây này",  
                     "Tôi vẫn còn bé lắm",  
                     "người yêu của tôi chính là cậu đấy",  
                     "Bầu trời xanh, làn mây trắng. Anh yêu nắng hay yêu em?",  
                     "Nhờ có nắng mới thấy cầu vồng. Nhờ có anh mới thấy màu hạnh phúc.",  
                     "Anh yêu ơi ới ời. Anh đang ở đâu?",  
                     "Soái ca là của ngôn tình. Còn anh thì chỉ của mình em thôi.",
```

```

        "Giữa cuộc đời hàng ngàn cảm dỗ.Em chỉ cần bến đỗ anh thôi.",
        "Bồ công anh bay khi có gió. Em chỉ cười vì ở đó có anh.",
        "Chỉ cần anh nói yêu, em sẽ bám theo anh suốt đời. Cô gái đang muốn muốn bật đèn xanh đấy. Cô
nàng muốn gợi ý là mình chung thủy lắm đấy. Anh cứ thử tỏ tình mà xem.",
        "Ba mươi chưa phải là Tết. Không làm bạn đâu phải là hết, còn có thể làm người yêu mà.",
        "Ai nào cho mượn avatar để em đỡ cô đơn đi",
        "Nắng đã có mũ, mưa đã có ô, còn em sẽ có ai?"]

    speak(choice(nguoi_yeu_noi))
    time.sleep(4)

#nếu hỏi tên bạn là gì?
def ten_ban(name):
    name_ban = ["tên của bạn là: {}".format(name),
                "Bạn tưởng Su quên tên bạn sao? tên bạn là {}".format(name),
                "Chào bạn {} nhé, tôi không quên tên của bạn đâu".format(name),
                "Tôi có trí nhớ siêu việt đấy bạn {} ạ!! Hihi...".format(name)]
    speak(choice(name_ban))
    time.sleep(6)

# Khi người dùng biểu lộ cảm xúc nó đáp
def Chan_qua_AI():
    chan_qua_noi = ["Tưởng gì tôi sẽ kể cho bạn một câu chuyện cười nhé, đảm bảo bạn sẽ vui",
                    "Chuyện nhỏ, để tôi, Su sẽ cố gắng làm cho bạn cười, hoặc biết đâu tôi sẽ khiến bạn ngạc nhiên đó,
bạn có muốn nghe tôi kể chuyện không?",
                    "Để tôi kể cho bạn nghe một câu chuyện nhé đảm bảo bạn sẽ cười đó."]
    speak(choice(chan_qua_noi))
    time.sleep(10)

# truyện cười
dap_chan_qua_noi = ["""Bạn nghe nhé:
        - Này Con! Anh cả con học kinh tế, anh hai thì học tài chính.
        Sao con không theo gương các anh mà học luật?
        - Bố nghĩ xem, nếu con không học làm luật sư thì sao này ai sẽ giúp anh hai con đây""",
                    """"Bạn nghe nhé:
        - Câu chuyện được mang tên là : CƯỜI!....!
        - Trong cuộc thi vấn đáp, ban giám khảo hỏi thí sinh:
        - Em tên gì?

```

- Em tên là Hà.
- Cô gái nói xong thì cười rất rạng rỡ.

Ban giám khảo hỏi:

- Tại sao em lại cười.

Cô gái trả lời:

- Dạ, tại vì đề của câu 1 dễ quá!

Ban giám khảo: ...!!! """"",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : XỬ NHẦM...!
- Sau khi có phán quyết ly dị vợ, cậu thấy thế nào?
- Bị đát! Chiếc xe hơi mua bằng tiền tố kiếm được, toà lại xử cho cô ấy.
- Còn lũ trẻ, mà tờ đình ninh là của người khác, toà lại xử cho sống chung với tờ.""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : ÔNG NỘI VÀ CHÁU...!
- Ông nội và người cháu đích tôn 3 tuổi đang ngồi chơi trò bán hàng.
- Cháu: - Đây tôi đưa bác 5.000 đồng, nhưng với một điều kiện.
- Ông: - Điều kiện gì cũng được.
- Cháu: - Thật không?
- Ông: - Thật. Bác cứ nói đi.
- Cháu: - Bác phải về dạy lại con bác đi nhé, con bác hay đánh tôi lắm đấy.""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : AN ỦI...!
- Cô: Nếu sau này em làm y tá, chuẩn bị tiêm thuốc cho 1 em nhỏ, em nhỏ sợ quá khóc òa lên,

vậy em có tiêm không?

- Bé: Không ạ!
- Cô: Vậy em có an ủi bé không ?
- Bé: Thưa cô, em sẽ an ủi là : thôi, đừng khóc nữa, nếu không cô chích cho 1 mũi bây giờ!""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : CHẾT... GIÀ...!
- Một người bị kết án tử hình khẩn cầu tòa giảm án.

Quan tòa bảo:

- Anh đã phạm tội tày trời làm sao chúng tôi tha được? Nhưng có thể chấp thuận cho anh được quyền chọn lựa cách chết.

Tử tù vội nói:

- Xin đội ơn ngài. Xin cho tôi được chết... già!""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : HỢP TÁC...!

- Ông chồng trò chuyện với vợ:

- Này em, từ ngày chúng ta dùng tiền để thưởng, con trai mình học khá hẳn lên, nhiều điểm 10 lắm, em thấy vui chứ?

- Theo em thì hẳn là nó đã đem tiền chia cho thầy giáo một nửa thì có.""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : KHEN KHÉO...!

- Mắm : Nè, ông thấy tôi mặc cái áo mới này như thế nào?

- Quỳnh: Ồ, tuyệt cú mèo!

- Mắm (hớn hờ): Thật hả? Ông ko nịnh tôi đó chứ?

- Quỳnh: Thật mà! Cái áo thì "tuyệt", còn bà là "cú mèo" đó

- Mắm: (suy sụp)!?"?""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : CÒN PHẢI XEM XÉT...!

- Chàng trai trở về nhà sau cuộc thi sát hạch lấy bằng lái xe với vẻ mặt hoang mang:

- Thật là rắc rối , Anh ta nói với ông bố : Chiếc xe tải đó có vấn đề.

- Nghĩa là con bị đánh trượt?

- Điều đó chưa rõ. Cả Ban giám khảo có còn ai chấm được điểm đâu ạ!"""]

```

ke_AI = get_text()
if "có" in ke_AI or "ok" in ke_AI or "kể" in ke_AI or "nghe" in ke_AI:
    speak(choice(dap_chan_qua_noi))
    time.sleep(30)
    #playsound.playsound('Tieng_cuoi.mp3')
    speak('Su Su cảm ơn bạn đã nghe SU kể chuyện!!!')
    time.sleep(3)
    for yess in dap_chan_qua_noi[1:]:
        nghe_tiepp = ["bạn có muốn nghe thêm không?",
                      "Bạn có muốn nghe Su kể nữa không",
                      "Để Su kể tiếp nhé"]
        speak(choice(nghe_tiepp))
        time.sleep(3)
        yess = get_text()
        if "có" not in yess or "ok" not in yess :
            break
        speak(choice(dap_chan_qua_noi))
        time.sleep(28)
        #playsound.playsound('Tieng_cuoi.mp3')
        speak('Su cảm ơn bạn đã lắng nghe!!!')

```

```

time.sleep(3)
elif "không" in ke_AI:
    nghe_ke_chuyen = ["Bạn không muốn nghe tôi kể chuyện sao, buồn quá",
                        "Xin lỗi bạn nhé, nhưng tôi chỉ muốn bạn vui thôi mà",
                        "bạn muốn làm gì khác sao?, hãy nói với tôi nhé"]
    speak(choice nghe_ke_chuyen))
time.sleep(4)
else:
    speak("Bạn vừa nói gì SU không hiểu. Xin hãy nói lại ạ")
time.sleep(5)

def Ke_chuyen_AI(name):
    #open(r"trolyao\Part\truyencuoi.py")
    truyen_cuoi_AI = ["""Bạn nghe nhé:
        - Đây Con! Anh cả con học kinh tế, anh hai thì học tài chính.
        Sao con không theo gương các anh mà học luật?
        - Bố nghĩ xem, nếu con không học làm luật sư thì sao này ai sẽ giúp anh hai con đây""",
        """Bạn nghe nhé:
        - Câu chuyện được mang tên là : CƯỜI....!
        - Trong cuộc thi vấn đáp, ban giám khảo hỏi thí sinh:
        - Em tên gì?
        - Em tên là Hà.
        - Cô gái nói xong thì cười rất rạng rỡ.
        Ban giám khảo hỏi:
        - Tại sao em lại cười.
        Cô gái trả lời:
        - Dạ, tại vì đề của câu 1 dễ quá!
        Ban giám khảo: ....!!! """,
        """Bạn nghe nhé:
        - Câu chuyện được mang tên là : XỬ NHẪM...!
        - Sau khi có phán quyết ly dị vợ, cậu thấy thế nào?
        - Bí đấy! Chiếc xe hơi mua bằng tiền tố kiếm được, toà lại xử cho cô ấy.
        Còn lũ trẻ, mà tố đình ninh là của người khác, toà lại xử cho sống chung với tố.""",
        """Bạn nghe nhé:
        - Câu chuyện được mang tên là : ÔNG NỘI VÀ CHÁU...!
        - Ông nội và người cháu đích tôn 3 tuổi đang ngồi chơi trò bán hàng.
        - Cháu: - Đây tôi đưa bác 5.000 đồng, nhưng với một điều kiện.

```

- Ông: - Điều kiện gì cũng được.

- Cháu: - Thật không?

- Ông: - Thật. Bác cứ nói đi.

- Cháu: - Bác phải về dạy lại con bác đi nhé, con bác hay đánh tôi lắm đấy.""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : AN ỦI...!

- Cô: Nếu sau này em làm y tá, chuẩn bị tiêm thuốc cho 1 em nhỏ, em nhỏ sợ quá khóc òa lên, vậy em có tiêm không?

- Bé: Không ạ!

- Cô: Vậy em có an ủi bé không ?

- Bé: Thưa cô, em sẽ an ủi là : thôi, đừng khóc nữa, nếu không cô chích cho 1 mũi bây giờ!""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : CHẾT... GIÀ...!

- Một người bị kết án tử hình khẩn cầu tòa giảm án.

Quan tòa bảo:

- Anh đã phạm tội tày trời làm sao chúng tôi tha được? Nhưng có thể chấp thuận cho anh được quyền chọn lựa cách chết.

Tử tù vội nói:

- Xin đội ơn ngài. Xin cho tôi được chết... già!""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : HỢP TÁC...!

- Ông chồng trò chuyện với vợ:

- Này em, từ ngày chúng ta dùng tiền để thưởng, con trai mình học khá hẳn lên, nhiều điểm 10 lắm, em thấy vui chứ?

- Theo em thì hẳn là nó đã đem tiền chia cho thầy giáo một nửa thì có.""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : KHEN KHÉO...!

- Mắm : Nè, ông thấy tôi mặc cái áo mới này như thế nào?

- Quỳnh: Ồ, tuyệt cú mèo!

- Mắm (hớn hờ): Thật hả? Ông ko nịnh tôi đó chứ?

- Quỳnh: Thật mà! Cái áo thì "tuyệt", còn bà là "cú mèo" đó

- Mắm: (suy sụp)!??""",

""""Bạn nghe nhé:

- Câu chuyện được mang tên là : CÒN PHẢI XEM XÉT...!

- Chàng trai trở về nhà sau cuộc thi sát hạch lấy bằng lái xe với vẻ mặt hoang mang:

- Thật là rắc rối , Anh ta nói với ông bố : Chiếc xe tải đó có vấn đề.

- Nghĩa là con bị đánh trượt?

```

        - Điều đó chưa rõ. Cả Ban giám khảo có còn ai chấm được điểm đâu ạ!"""]

speak(choice(truyen_cuoi_AI))
time.sleep(28)
#playsound.playsound('Tiemg_cuoi.mp3')
speak('Su cảm ơn bạn đã lắng nghe!!!')
time.sleep(3)
for chuyen_chay in truyen_cuoi_AI[1:]:
    nghe_tiep = ["bạn có muốn nghe thêm không?",
                  "Bạn có muốn nghe Su kể nữa không",
                  "Để su kể tiếp nhé"]
    speak(choice(nghe_tiep))
    time.sleep(3)
    chuyen_chay = get_text()
    if "có" not in chuyen_chay or "ok" not in chuyen_chay :
        break
    speak(choice(truyen_cuoi_AI))
    #playsound.playsound('Tiemg_cuoi.mp3')
    time.sleep(10)
    speak('SU cảm ơn bạn đã lắng nghe!!!')
    time.sleep(3)
# phần điều khiển video, trình phát nhạc
def tatungdung():
    pyautogui.hotkey('cmd','q')
    pyautogui.hotkey('enter')
    speak("chương trình đã được tắt")
    time.sleep(3)
def tat():
    pyautogui.hotkey('m')
    speak("OK cậu chủ")
    time.sleep(3)
def bat():
    pyautogui.hotkey('m')
    speak("OK cậu chủ")
    time.sleep(3)
def chuyenbai():
    pyautogui.hotkey('shift','N')
    speak("đã chuyển bài")

```

```

time.sleep(3)
def phongto():
    pyautogui.hotkey('f')
    speak("đã phóng to")
    time.sleep(3)
def tua():
    pyautogui.hotkey('right')
    speak("OK cậu chủ")
    time.sleep(3)
def lui():
    pyautogui.hotkey('left')
    speak("OK cậu chủ")
    time.sleep(3)
def pause():
    pyautogui.hotkey('space')
    speak("OK cậu chủ")
    time.sleep(3)
def play():
    pyautogui.hotkey('space')
    speak("OK cậu chủ")
    time.sleep(3)
def volumedown():
    pyautogui.hotkey('down')
    pyautogui.hotkey('down')
    pyautogui.hotkey('down')
    pyautogui.hotkey('down')
    pyautogui.hotkey('down')
def volumeup():
    pyautogui.hotkey('up')
    pyautogui.hotkey('up')
    pyautogui.hotkey('up')
    pyautogui.hotkey('up')
    pyautogui.hotkey('up')

```

Ở đây mình làm các tương tác vui vẻ cho trợ lý ảo AI.

2.17 Xây dựng form giao diện cho trợ lý ảo AI


```

class TrolyaoTV(App):
    def build(self):
        #returns a window object with all it's widgets
        self.window = GridLayout()
        self.window.cols = 1
        self.window.size_hint = (0.6, 0.7)
        self.window.pos_hint = {"center_x": 0.5, "center_y":0.5}

        # image widget
        self.window.add_widget(Image(source="logo.png"))

        # label widget
        self.greeting = Label(
            text= "Trợ Lý Ảo Tiếng Việt",
            font_size= 30,
            color= '#00FFCE'
        )
        self.window.add_widget(self.greeting)

        self.button = Button(
            text= "VOICE",
            size_hint= (1,0.5),
            bold= True,
            background_color = '#00FFCE',
        )
        self.button.bind(on_press=self.callback)
        self.window.add_widget(self.button)
        return self.window

    def callback(self, instance):
        assistant()
        # change label text to "Hello + user name!"
        #self.greeting.text = "Hello " + boot+ "!"

# run TrolyaoTV App Calss
if __name__ == "__main__":
    TrolyaoTV().run()

```

Mình sử dụng thư viện kivy để tạo giao diện cho trợ lý thì cần cài đặt các packet

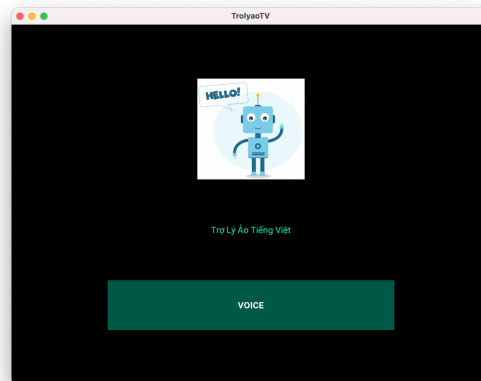
```
from kivy.app import App
from kivy.uix.gridlayout import GridLayout
from kivy.uix.label import Label
from kivy.uix.image import Image
from kivy.uix.button import Button
```

3. Kết Quả đạt được

Đã xây dựng được 1 ứng dụng chạy trên macos có tên Trợ lý ảo TV

- Task 1: Giao tiếp chào hỏi (1 ngày)
- Task 2: Xem thời gian hiện tại (1 ngày)
- Task 3: Mở ứng dụng và bất kỳ trang web nào trên Chrome (1 ngày)
- Task 4: Tìm kiếm trên Google (1 ngày)
- Task 5: Dự Báo thời tiết (1 ngày)
- Task 6: Phát cho bạn 1 bản nhạc trên youtube (1 ngày)
- Task 7: Đọc báo hằng ngày (1 ngày)
- Task 8: Gửi email tự động
- Task 9: Cho tôi biết về cả thế giới này (Wikipedia)
- Task 10: Kể chuyện (tương tác với AI)

Các chức năng hoạt động tốt, và mình có thêm 1 số chức năng khác như trò chuyện với AI, chức năng tìm định nghĩa trên từ điển wikipedia, chức năng kể chuyện



4. Tài liệu tham khảo

- Xây dựng trợ lý ảo Tiếng Việt bằng ngôn ngữ Python (cơ bản): <https://quantraai.com/xay-dung-tro-ly-ao-tieng-viet-bang-ngon-ngu-python-co-ban/>
- Xây dựng một robot trợ lý ảo với python thật đơn giản: <https://viblo.asia/p/xay-dung-mot-robot-tro-ly-ao-voi-python-that-don-gian-OeVKBWLQZkW>
- Xây dựng giao diện bằng kivy: <https://kivy.org/doc/stable/tutorials-index.html>
- Đóng gói ứng dụng MacOS bằng py2app: <https://py2app.readthedocs.io/en/latest/>
- Đóng gói ứng dụng cho window bằng Pyinstaller: <https://laptrinhcanban.com/python/nhap-mon-lap-trinh-python/dong-goi-chuong-trinh-python/dong-goi-chuong-trinh-python-pyinstaller/>

5. Soucer code

Github: <https://github.com/SonDt123/trolyaotiengviet.git>