

ASA Chat Bot
With Speech Recognition
for Luna Cat

- [1. Luna_Cat_Chat_Main](#)
- [2. Play_VT](#)
- [3. Record_Speech](#)
- [4. SpeechToText](#)
- [5. Text_to_Speech](#)
- [6. GPT_Chat](#)
- [7. Further Development Direction](#)

1. Luna_Cat_Chat_Main

A programme created to become the central system of the Luna Cat Chat Bot

When called, the programme will wait for the users' input and execute further with the users' input.

This programme will play some audio in return and keep looping.

Code Explain:

```
# Function trigger by keyword

def keyword_trigger():

    keyword = False

    try:

        with sr.Microphone() as source:

            print('Listening... ')
            voice = listener.listen(source)
            command = listener.recognize_google(voice)

            play_Wav(listen_path)

            command = command.lower()

            print(command)

            if 'hey luna' or 'hello luna' or 'hi luna' in command:

                keyword = True
                command = None

            else:

                keyword = False

    except:

        keyword = False
        pass

    return keyword
```

This function uses the python package - speech_recognition to check the users' audio input. If the keyword is in the audio file, the function will trigger and return True.

```

# call the keyword trigger function

trigger = keyword_trigger()

if trigger == True:
    """play_Wav(answer_path)

    time.sleep(1)

    play_Wav(notice_path)"""

    result = Record_Audio()

if result == True:
    # Load the Audio File

    audio_file = 'temp.wav'

    text, lang_code ,trans2text = wavToText(audio_file)

    print(lang_code)

if trans2text == True:
    if 'play' in text:
        song = text.replace('play', '')

        play_music(song)

    elif '播放' in text:
        song = text.replace('播放', '')

        play_music(song)

    else:
        answer, ansGPT = chating_bot(text , lang_code)

```

With the keyword trigger, the programme will start to record the users' input and turn the audio data to text with the help of the "wavToText" function and return the text to the programme.

The programme then will analyze the text return and decide which function it needs to execute.

If the users' input has the keyword play, the programme will execute the "play_music" function, Else, the programme will execute the "chating_bot" function.

2. Play_VT

This function is created for playing the video or music from Youtube with the users' input.

This function required the Google Cloud Api key and a Youtube Premium Subscription to execute.

Code Explain:

```
def play_music(song):  
    # Set up API client  
    DEVELOPER_KEY = "AIzaSyAJUHcPmM7quJNaW_qotSii52V83WY6zn4"  
    YOUTUBE_API_SERVICE_NAME = "youtube"  
    YOUTUBE_API_VERSION = "v3"  
    youtube = build(YOUTUBE_API_SERVICE_NAME, YOUTUBE_API_VERSION, developerKey=DEVELOPER_KEY)  
  
    # Search for videos  
    search_response = youtube.search().list(  
        q = song,  
        part = "id",  
        type = "video",  
        fields = "items/id"  
    ).execute()  
  
    # Get the first video ID  
    video_id = search_response["items"][0]["id"]["videoId"]  
  
    # Play the video in a web browser  
    webbrowser.open(f"https://www.youtube.com/watch?v={video_id}")
```

Using Youtube Data API from Google Cloud to search the url link with the users' input.
The programme will pop up a web browser to play the searched video.

3. Record_Speech

This programme is created to record the audio from the user.

This programme will keep recording the sound until the user keeps silent for the whole 3 seconds.

This program saves and reads the file in Wav format.

Code Explain:

```
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100
CHUNK = 1024
SILENCE_THRESHOLD = 1500 # adjust this as needed
SILENCE_DURATION = 3 # adjust this as needed

p = pyaudio.PyAudio()

stream = p.open(
    format = FORMAT,
    channels=CHANNELS,
    rate=RATE,
    input=True,
    frames_per_buffer=CHUNK
)

frames = []

print("Recording started...")

silence_counter = 0
max_silence_duration = int(RATE / CHUNK * SILENCE_DURATION)
```

The above image is the setting of the recording environment.

Parameters `silence_threshold` and `silence_duration` can be customized for different needs. Parameter "`SILENCE_THRESHOLD`" represents the minimum hz the recording system will accept. And parameter "`SILENCE_DURATION`" represents the maximum duration that the recording sound is below the threshold.

```

while True:

    data = stream.read(CHUNK)
    frames.append(data)
    rms = audioop.rms(data, 2)
    if rms < SILENCE_THRESHOLD:
        silence_counter += 1
        #print(silence_counter)
    else:
        silence_counter = 0

    if silence_counter >= max_silence_duration:
        break

print("Recording stopped...")

stream.stop_stream()
stream.close()
p.terminate()

wf = wave.open("temp.wav", "wb")
wf.setnchannels(CHANNELS)
wf.setsampwidth(p.get_sample_size(FORMAT))
wf.setframerate(RATE)
wf.writeframes(b"".join(frames))
wf.close()

print("Audio file saved as temp.wav")

return True

```

The program will record the users' input and after the counter hits the maximum silence duration. The program will save the audio file as temp.wav for further instruction.

4. SpeechToText

This program is created to translate users' recorded audio to text for further instruction.

This program can detect English input and also Cantonese input.

Code Explain:

```
client_file = 'audio-bot-389910-9ef67fc68a47.json'
credit = service_account.Credentials.from_service_account_file(client_file)
client = speech.SpeechClient(credentials = credit)

first_lang = "en-US"
second_lang = "yue-Hant-HK"

with open(path, "rb") as audio_file:
    content = audio_file.read()

audio = speech.RecognitionAudio(content=content)

config = speech.RecognitionConfig(
    encoding=speech.RecognitionConfig.AudioEncoding.LINEAR16,
    language_code=first_lang,
    alternative_language_codes=[second_lang],
)

print("Waiting for operation to complete...")
response = client.recognize(config=config, audio=audio)

for result in response.results:
    text = result.alternatives[0].transcript

for result in response.results:
    lang_code = result.language_code

return text, lang_code , True
```

Remember to check the location of the Api client json file.

5. Text_to_Speech

This program was created to turn the text input to audio file.

The Audio file will be in .WAV format.

With the language code given by the speech to text, this program will generate an audio file with the help of google cloud texttospeech api.

Code Explain:

```
def text_to_wav(txt, lang_code):

    client_file = 'audio-bot-389910-9ef67fc68a47.json'
    credit = service_account.Credentials.from_service_account_file(client_file)

    text_input = tts.SynthesisInput(text=txt)

    Voice_params_en = tts.VoiceSelectionParams(
        language_code = 'en-US',
        name = 'en-US-Neural2-H'
    )

    Voice_params_cn = tts.VoiceSelectionParams(
        language_code = 'yue-Hant-HK',
        name = 'yue-HK-Standard-A'
    )

    if lang_code == "en-us":
        Voice_params = Voice_params_en

    if lang_code == "yue-hant-hk":
        Voice_params = Voice_params_cn

    audio_config = tts.AudioConfig(audio_encoding=tts.AudioEncoding.LINEAR16)

    client = tts.TextToSpeechClient(credentials = credit)

    response = client.synthesize_speech(
        input=text_input,
        voice=Voice_params,
        audio_config=audio_config,
    )

    filename = "audio.wav"

    with open(filename, "wb") as out:
        out.write(response.audio_content)

    return True
```

6. GPT_Chat

This program was created to provide a communication function with the users' input.

This program uses the ChatGPT to answer the users' questions.

Code Explain:

```
def chatting_bot(question, lang_code):  
  
    # Set up the GPT-3 model  
    model_engine = "text-davinci-002"  
    model_prompt_cn = (  
        "Q: 你是誰?\n"  
        "A: 我叫作「陸奧娜貓」.\n"  
        "Q: 你是做什麼的?\n"  
        "A: 我是一位專業護士, 可以回答你的問題並和你聊天.\n"  
        "Q: 誰創造了你?\n"  
        "A: 我是由Asa Robotics Limited開發的.\n"  
        "Q: 請你用粵語口語嚟答我問題\n"  
        "A: 冇問題, 我可以用粵語口語回答你嘅問題.\n"  
    )  
  
    model_prompt_en = (  
        "Q: Who are you?\n"  
        "A: My name is Luna Cat.\n"  
        "Q: What are you?\n"  
        "A: I am a professional nurse, that can answer your question and chat with you.\n"  
        "Q: Who created you\n"  
        "A: I am developed by Asa Robotics Limited.\n"  
    )  
  
    if lang_code == "en-us":  
        model_prompt = model_prompt_en  
  
    if lang_code == "yue-hant-hk":  
        model_prompt = model_prompt_cn
```

Input the configuration for the ChatGPT to answer the question with certain roles.

With the given language code by the program "SpeechToText", the program will answer users' questions in a certain language.

7. Further Development Direction

- Play the video from Youtube with a external media player

To execute the function on Luna Cat, the program "PlayYT" has to use a media player to play the video instead of using a web browser pop up window.

Required a google account with google workspace to use the google cloud api service and youtube premium to provide youtube data v3 authority token

<https://www.geeksforgeeks.org/playing-youtube-video-using-python/>

- Using Vits text to speech for more humanize voice

To avoid the audio answer return too robotics, using a better texttospeech module to produce a more humanize audio file is the solution

https://www.youtube.com/watch?v=riYOD_EFKDE

<https://github.com/svc-develop-team/so-vits-svc>

- ChatGpt Configuration

Update the ChatGPT configuration for the AI robot for a better and humanize answer return

https://github.com/zixiu/Digital_Life_Server

- Keyword detection improve

For Cantonese keyword detection, the accuracy is not great enough. For improvement, maybe we need to apply AI training in the detection to improve the performance.