

Forum:[Hack Forums](#)

Thread: [Speech to text chatgpt - speech synthesis.](#)

Mr. Robot - May 3, 2023, 10:19 am

Was just messing around with a fun background program to run when you want questions answered. Also wanted to use speech synthesis so in the future I can do peter griffin or some shit.

If I missed a setup step to make this as streamlined for windows as possible just let me know and I will add a fix.

Also note: I added in to make it always respond with a mafia accent, the only reason for that is for you guys to see how it works so you can prefix whatever you want to your prompt via voice.

Get your API key

<https://platform.openai.com/account/api-keys>

Basically just download anaconda

<https://www.anaconda.com/>

Then setup an environment:

Code `conda create --name my_tts_env`

`conda activate my_tts_env`

Then install these:

Code `pip install pyaudio`

`pip install SpeechRecognition`

`pip install winsound`

`pip install openai`

`pip install tts`

Also I added in `fetching.wav` which just tells you that it is fetching the data. I would recommend you make one as well with something like:

Code `tts --text "Fetching that for you now" --model_name tts_models/en/ljspeech/vits--neon`

Then rename that output to `fetching.wav`

Doing this is also a good test to make sure the model is downloaded and it's working.

Code `import os`

`import re`

`import time`

`import pyaudio`

`import speech_recognition as sr`

`import winsound`

`import openai`

`api_key = "APIKEY GOES HERE"`

`os.system('title AI Voice Recognition using chatGPT')`

`lang = 'en'`

`openai.api_key = api_key`

`guy = ""`

```
stop_flag = False
```

```
question_styles = ["how many", "what are", "i'm trying", "i am trying", "hey", "what is", "what's the",  
"who was", "how does", "when do", "why is", "who is", "what are", "how can", "what should", "how do  
I", "what's the difference", "what happens if", "how do I troubleshoot", "what's the syntax", "how do I  
install", "what are the best practices", "what are the limitations of", "what's the typical use case",  
"what's the performance of", "what's the scalability of", "what's the security of", "how do I  
implement", "what are the parameters for", "how do I use", "what are the options for", "what is the  
output of", "what is the input of", "what are the different types of", "what is the relationship between",  
"what are the advantages of", "what is the role of", "how does it relate to", "what is the difference  
between", "what is the impact of", "what is the meaning of the term", "what is the algorithm for"]
```

```
def sanitize_text(text):  
    clean_text = re.sub(r'[^a-zA-Z0-9\s.],', "", text)  
    return clean_text
```

```
while not stop_flag:  
    def get_audio():  
        global stop_flag  
        os.system('cls' if os.name == 'nt' else 'clear')  
        print(f"AI Voice chat bot Using ChatGPT")  
        r = sr.Recognizer()  
        r.energy_threshold = 4000  
  
        with sr.Microphone(device_index=1) as source:  
            audio = r.listen(source)  
            said = ""  
  
            try:  
                said = r.recognize_google(audio)  
                said = "Respond with a mafia boss accent " + said  
                print(said)  
                global guy  
                guy = said
```

```
            if "stop" in said.lower() or "shut up" in said.lower():  
                winsound.PlaySound(None, winsound.SND_PURGE) # stop current sound  
                return
```

```
            question_detected = any(style in said.lower() for style in question_styles)
```

```
            if question_detected:  
                try:  
                    winsound.PlaySound("fetching.wav", winsound.SND_ASYNC)  
                    completion = openai.ChatCompletion.create(model="gpt-3.5-turbo",  
messages=[{"role": "user", "content": said}])  
                    except openai.OpenAIError as e:  
                        print("OpenAI error: {}".format(e))  
                        return  
  
                    text = completion.choices[0].message.content  
                    clean_text = sanitize_text(text)  
                    print(f'{clean_text}')
```

```

        # Use os.system to generate TTS audio and save it as an MP3 file
        os.system(f'tts --text "{clean_text}" --model_name tts_models/en/ljspeech/vits--neon >
{"nul" if os.name == "nt" else "/dev/null"} 2>&1')

        # Play the saved MP3 file using winsound
        winsound.PlaySound("tts_output.wav", winsound.SND_ASYNC)

        if "tell me again" in said.lower() or "please repeat" in said.lower():
            winsound.PlaySound("tts_output.wav", winsound.SND_ASYNC)

    except sr.UnknownValueError:
        print("Speech Recognition could not understand audio")
        os.system('cls' if os.name == 'nt' else 'clear')
        print(f'AI Voice chat bot Using ChatGPT')
    except sr.RequestError as e:
        os.system('cls' if os.name == 'nt' else 'clear')
        print(f'AI Voice chat bot Using ChatGPT')
        print("Could not request results from Speech Recognition service: {0}".format(e))
    return said

get_audio()

```

ALEDIO - May 3, 2023, 11:13 am

It seems incredibly fun to use chatGPT in this way, it could also be applied as a communication assistant by both sides, we could interact with it exclusively via voice and I think it would be very fun. Great idea, really good idea.

javierpena - May 3, 2023, 1:07 pm

Half the things you do I don't understand but still enjoy almost every thread of yours!