Real-time Speech to Text to Speech : Building Your AI-Based Alexa

Khoi Duong
Prof. Chang
CS589
3/13/2024

Table of contents

requirements.txt

The file contains more than enough modules to run for the project. If any module is missing during the execution, we can install more using pip install.

TextSpeech > requirements.txt

Click here to ask Blackbox to help you code faster

```
pydub
openai==0.28
python-dotenv
langchain
datetime
wikipedia
langchain_experimental
langchain_openai
numexpr
pypdf
yt_dlp
chromadb
SpeechRecognition
openai-whisper
torch
gTTS
pyaudio
```

```
Building wheels for collected packages: wikipedia, openai-whisper, pyaudio, pypika
  Building wheel for wikipedia (setup.py) ... done
  Created wheel for wikipedia: filename=wikipedia-1.4.0-py3-none-any.whl size=11680 sha256=ab5a315de23b4942691edc8e8dce547ce83633b0b7454906c7f160f0b71af623
  Stored in directory: /home/koiisme/.cache/pip/wheels/8f/ab/cb/45ccc40522d3a1c41e1d2ad53b8f33a62f394011ec38cd71c6
  Building wheel for openai-whisper (pyproject.toml) ... done
  Created wheel for openai-whisper: filename=openai_whisper-20231117-py3-none-any.whl size=801356 sha256=89f06418eaf44f02afda984745a852aa5bebe3999135206c784dac83e1065
  Stored in directory: /home/koiisme/.cache/pip/wheels/55/5d/42/c296ab046d52caa0adc0e3f159e98f011b3994a022d6282105
  Building wheel for pyaudio (pyproject.toml) ... error
  error: subprocess-exited-with-error

  × Building wheel for pyaudio (pyproject.toml) did not run successfully.
  │ exit code: 1
  ╰─> [18 lines of output]
      running bdist_wheel
      running build
      running build_py
      creating build
      creating build/lib.linux-x86_64-cpython-311
      creating build/lib.linux-x86_64-cpython-311/pyaudio
      copying src/pyaudio/__init__.py -> build/lib.linux-x86_64-cpython-311/pyaudio
      running build_ext
      building 'pyaudio._portaudio' extension
      creating build/temp.linux-x86_64-cpython-311
      creating build/temp.linux-x86_64-cpython-311/src
      creating build/temp.linux-x86_64-cpython-311/src/pyaudio
      x86_64-linux-gnu-gcc -Wsign-compare -DNDEBUG -g -fwrapv -O2 -Wall -g -fstack-protector-strong -Wformat -Werror=format-security -g -fwrapv -O2 -fPIC -I/usr/local/
ce_api.c -o build/temp.linux-x86_64-cpython-311/src/pyaudio/device_api.o
      src/pyaudio/device_api.c:9:10: fatal error: portaudio.h: No such file or directory
          9 | #include "portaudio.h"
            |          ^~~~~~~~~~~~~
      compilation terminated.
      error: command '/usr/bin/x86_64-linux-gnu-gcc' failed with exit code 1
      [end of output]

  note: This error originates from a subprocess, and is likely not a problem with pip.
  ERROR: Failed building wheel for pyaudio
  Building wheel for pypika (pyproject.toml) ... done
  Created wheel for pypika: filename=PyPika-0.48.9-py2.py3-none-any.whl size=53723 sha256=45b2e3f92954288802e591e10deac17a0f8f923603d89efdb841d21e8f5b1dfd
  Stored in directory: /home/koiisme/.cache/pip/wheels/a3/01/bd/4c40ceb9d5354160cb186dcc153360f4ab7eb23e2b24daf96d
Successfully built wikipedia openai-whisper pypika
Failed to build pyaudio
ERROR: Could not build wheels for pyaudio, which is required to install pyproject.toml-based projects
```

We got some problems when trying to install pyaudio.

To fix the problem, simply install portaudio19 with:

- For VSCode: ***pip install sudo***
- For Linux/Ubuntu: ***sudo apt-get install python3-dev portaudio19-dev***

```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/TextSpeech$ sudo apt-get install python3-dev portaudio19-
dev
[sudo] password for koiisme:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-dev is already the newest version (3.10.6-1~22.04).
python3-dev set to manually installed.
The following additional packages will be installed:
  libasound2-dev libjack-dev libjack0 libportaudiocpp0 pkg-config uuid-dev
Suggested packages:
  libasound2-doc jackd1 portaudio19-doc
The following packages will be REMOVED:
  libjack-jackd2-0
The following NEW packages will be installed:
  libasound2-dev libjack-dev libjack0 libportaudiocpp0 pkg-config portaudio19-dev uuid-dev
0 upgraded, 7 newly installed, 1 to remove and 30 not upgraded.
Need to get 612 kB of archives.
After this operation, 3266 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libjack0 amd64 1:0.125.0-3build2 [93
.3 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 libasound2-dev amd64 1.2.6.1-1ubuntu1 [1
10 kB]
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 pkg-config amd64 0.29.2-1ubuntu3 [48.2 k
B]
Get:4 http://archive.ubuntu.com/ubuntu jammy/main amd64 uuid-dev amd64 2.37.2-4ubuntu3 [33.1 kB]
Get:5 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libjack-dev amd64 1:0.125.0-3build2
[206 kB]
Get:6 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libportaudiocpp0 amd64 19.6.0-1.1 [1
6.1 kB]
Get:7 http://archive.ubuntu.com/ubuntu jammy/universe amd64 portaudio19-dev amd64 19.6.0-1.1 [10
```

We can then install pyaudio easily.

```
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/TextSpeech$ pip install pyaudio
Collecting pyaudio
  Using cached PyAudio-0.2.14.tar.gz (47 kB)
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Building wheels for collected packages: pyaudio
  Building wheel for pyaudio (pyproject.toml) ... done
  Created wheel for pyaudio: filename=PyAudio-0.2.14-cp310-cp310-linux_x86_64.whl size=63858 sha
256=e356a39ab0308e9d686b7e6bca78670b07c8ae1bdf154155ade60da970895177
  Stored in directory: /home/koiisme/.cache/pip/wheels/d6/21/f4/0b51d41ba79e51b16295cbb096ec49f3
34792814d545b508c5
Successfully built pyaudio
Installing collected packages: pyaudio
Successfully installed pyaudio-0.2.14
(venv) koiisme@DESKTOP-LVBMC2V:~/CS589/TextSpeech$
```

Import modules

For the first part, we need to import all modules needed, and using dotenv module to obtain the OpenAI API key from .env file

TextSpeech > app.py > reply

Click here to ask Blackbox to help you code faster

```python
from pydub import AudioSegment
from pydub.playback import play
import speech_recognition as sr
import whisper
import queue
import os
import threading
import torch
import numpy as np
import re
from gtts import gTTS
import openai
import click
import sys
sys.path.append('../..')
from dotenv import load_dotenv, find_dotenv
# read local .env file
_ = load_dotenv(find_dotenv())

openai.api_key  = os.environ['OPENAI_API_KEY']
```

@click.command()

We use “click” module to obtain the parameter for our program to run later. The default parameters are: --model, --english, --energy, --pause, --dynamic_energy, --wake_word, and --verbose

```
@click.command()
@click.option("--model", default="base", help="Model to use", type=click.Choice(["tiny", "base", "small", "medium", "large"]))
@click.option("--english", default=False, help="Whether to use the English model", is_flag=True, type=bool)
@click.option("--energy", default=300, help="Energy level for the mic to detect", type=int)
@click.option("--pause", default=0.8, help="Pause time before entry ends", type=float)
@click.option("--dynamic_energy", default=False, is_flag=True, help="Flag to enable dynamic energy", type=bool)
@click.option("--wake_word", default="hey computer", help="Wake word to listen for", type=str)
@click.option("--verbose", default=False, help="Whether to print verbose output", is_flag=True, type=bool)
```

Main function

Define the main() function that:

- Adjusts the model name if English is selected and the model is not "large".
- Loads the audio model using whisper.load_model().
- Creates queues for audio and result data.
- Starts threads to run record_audio, transcribe_forever, and reply functions concurrently.
- Prints results from the result queue indefinitely.

```python
def main(model, english, energy, pause, dynamic_energy, wake_word, verbose):
    if model != "large" and english:
        model = model + ".en"
    audio_model = whisper.load_model(model)
    audio_queue = queue.Queue()
    result_queue = queue.Queue()

    threading.Thread(target=record_audio, args=(audio_queue, energy, pause, dynamic_energy,)).start()
    threading.Thread(target=transcribe_forever, args=(audio_queue, result_queue, audio_model, english, wake_word, verbose,)).start()
    threading.Thread(target=reply, args=(result_queue,)).start()

    while True:
        print(result_queue.get())
```

record_audio()

The function above record_audio records audio from a microphone and saves it to a queue for further processing.

```python
def record_audio(audio_queue, energy, pause, dynamic_energy):
    r = sr.Recognizer()
    # print("List microphone")                      # use to debug, if microphones not found
    # print(sr.Microphone.list_microphone_names())  # then we need to change the env since it doesn't support audio
    r.energy_threshold = energy
    r.pause_threshold = pause
    r.dynamic_energy_threshold = dynamic_energy

    with sr.Microphone(sample_rate=16000) as source:
        print("Listening...")
        i = 0
        while True:
            audio = r.listen(source)
            torch_audio = torch.from_numpy(np.frombuffer(audio.get_raw_data(), np.int16).flatten().astype(np.float32) / 32768.0)
            audio_data = torch_audio
            audio_queue.put_nowait(audio_data)
            i += 1
```

transcribe_forever()

The transcribe_forever function receives two queues:

- audio_queue, which contains the audio data to be transcribed, and
- result_queue, which is used to store the transcribed text.

If the predicted_text string starts with the wake_word, the function processes the text by removing the wake_word from the start of the string using regular expressions.

Finally, the predicted_text string is added to the result_queue using the result_queue.put_nowait() method.

```python
def transcribe_forever(audio_queue, result_queue, audio_model, english, wake_word, verbose):
    while True:
        audio_data = audio_queue.get()
        if english:
            result = audio_model.transcribe(audio_data, language='english')
        else:
            result = audio_model.transcribe(audio_data)

        predicted_text = result["text"]

        if predicted_text.strip().lower().startswith(wake_word.strip().lower()):
            pattern = re.compile(re.escape(wake_word), re.IGNORECASE)
            predicted_text = pattern.sub("", predicted_text).strip()
            punc = '''!()-[]{};:'"\,<>./?@#$%^&*_~'''
            predicted_text = predicted_text.translate({ord(i): None for i in punc})
            if verbose:
                print("You said the wake word.. Processing ...")
                print("You said:" + predicted_text)

                result_queue.put_nowait(predicted_text)
        else:
            if verbose:
                print("You did not say the wake word.. Ignoring")
```

reply()

The function loops continuously to wait for results from the result_queue passed as an argument.

```python
def reply(result_queue):
    while True:
        result = result_queue.get()
        data = openai.Completion.create(
            model="gpt-3.5-turbo-instruct",
            prompt=result,
            temperature=0,
            max_tokens=150,
        )
        answer = data["choices"][0]["text"]
        print("The answer content:" + answer)
        print("Transform the answer to mp3 ... Result will be in reply.mp3!")
        mp3_obj = gTTS(text=answer, lang="en", slow=False)
        mp3_obj.save("reply.mp3")
        reply_audio = AudioSegment.from_mp3("D:\VS CODE\Python\CS589\TextSpeech\reply.mp3")
        # change the path based on the location of reply.mp3
        play(reply_audio)
```

Executing the program

```
(venv) PS D:\VS CODE\Python\CS589\TextSpeech> python app.py --model base --english --energy 300 --pau
se 0.8 --dynamic_energy --wake_word "hey computer" --verbose
D:\VS CODE\Python\CS589\TextSpeech\venv\lib\site-packages\pydub\utils.py:170: RuntimeWarning: Couldn'
t find ffmpeg or avconv - defaulting to ffmpeg, but may not work
  warn("Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work", RuntimeWarning)
Listening...
D:\VS CODE\Python\CS589\TextSpeech\venv\lib\site-packages\whisper\transcribe.py:115: UserWarning: FP1
6 is not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
You said the wake word.. Processing ...
You said: who is the founder of OpenAI
The answer content:, a research institute that is dedicated to developing friendly AI. He is also a c
o-founder of Tesla and Neuralink, companies that are focused on developing advanced technologies in t
he fields of electric cars and brain-computer interfaces, respectively.

Musk was born in South Africa in 1971 and showed an early interest in computers and technology. He mo
ved to Canada at the age of 17 to attend Queen's University and then transferred to the University of
 Pennsylvania, where he received dual bachelor's degrees in economics and physics. He then went on to
 pursue a PhD in energy physics at Stanford University, but dropped out after two days to pursue entr
epreneurial ventures.

In 1995, Musk co-founded Zip2, a company that provided online content publishing software for newspap
ers
Transform the answer to mp3 ... Result will be in reply.mp3!
```

```
(venv) PS D:\VS CODE\Python\CS589\TextSpeech> python app.py --model base --english --energy 300 --pause 0.
8 --dynamic_energy --wake_word "hey computer" --verbose
D:\VS CODE\Python\CS589\TextSpeech\venv\lib\site-packages\pydub\utils.py:170: RuntimeWarning: Couldn't fin
d ffmpeg or avconv - defaulting to ffmpeg, but may not work
  warn("Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work", RuntimeWarning)
Listening...
D:\VS CODE\Python\CS589\TextSpeech\venv\lib\site-packages\whisper\transcribe.py:115: UserWarning: FP16 is
not supported on CPU; using FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
You did not say the wake word.. Ignoring
You did not say the wake word.. Ignoring
You said the wake word.. Processing ...
You said: wheres the location of San Francisco Bay University
The answer content:

San Francisco Bay University is located in San Francisco, California, United States.
Transform the answer to mp3 ... Result will be in reply.mp3!
```

Enhance with a better reply()

```python
def reply(result_queue, verbose):
    while True:
        question = result_queue.get()
        # We use the following format for the prompt: "Q: ?\nA:"
        prompt = "Q: {}?\nA:".format(question)

        data = openai.Completion.create(
            model="text-davinci-002",
            prompt=prompt,
            temperature=0.5,
            max_tokens=100,
            n=1,
            stop=["\n"]
        )

        # We catch the exception in case there is no answer
        try:
            answer = data["choices"][0]["text"]
            mp3_obj = gTTS(text=answer, lang="en", slow=False)
        except Exception as e:
            choices = [
                "I'm sorry, I don't know the answer to that",
                "I'm not sure I understand",
                "I'm not sure I can answer that",
                "Please repeat the question in a different way"
            ]
            mp3_obj = gTTS(text=choices[np.random.randint(0, len(choices))], lang="en", slow=False)
            if verbose:
                print(e)

        # In both cases, we play the audio
        mp3_obj.save("reply.mp3")
        reply_audio = AudioSegment.from_mp3("reply.mp3")
        play(reply_audio)
```

Result

```
use 0.8 --dynamic_energy --wake_word "hey computer" --verbose
D:\VS CODE\Python\CS589\TextSpeech\venv\lib\site-packages\pydub\utils.py:170: RuntimeWarning: Couldn't find ffmpeg or avconv - de
faulting to ffmpeg, but may not work
  warn("Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work", RuntimeWarning)
Listening...
D:\VS CODE\Python\CS589\TextSpeech\venv\lib\site-packages\whisper\transcribe.py:115: UserWarning: FP16 is not supported on CPU; u
sing FP32 instead
  warnings.warn("FP16 is not supported on CPU; using FP32 instead")
You did not say the wake word.. Ignoring
You did not say the wake word.. Ignoring
You did not say the wake word.. Ignoring
You said the wake word.. Processing ...
You said: splab splab splab splab splab splab splab
The answer content: I'm sorry, I do not understand what you are asking. Can you please rephrase your question?
Transform the answer to mp3... Result will be in reply.mp3!
```

References

- Real-time Speech to Text to Speech : Building Your AI-Based Alexa
- Text to speech - OpenAI API

Source code: https://github.com/MynameisKoi/CS589/tree/main/TextSpeech