1.2.4 Speech Recognition

1. What is Speech Recognition?

Speech Recognition (Speech-to-Text, STT) is a technology that converts human speech signals into text or executable commands. In this course, we will implement speech recognition functionality using Alibaba OpenAI's Speech Recognition API.

2. How It Works

The wave library is used to extract audio data. The extracted audio is then sent to OpenAI's ASR (Automatic Speech Recognition) model. The recognized text returned by the ASR model is stored in speech_result for use in subsequent processes.

3. Preparation Before the Experiment

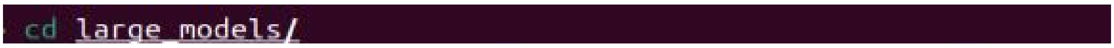
Before proceeding, refer to the course “1.2 Large Language Models Deployment” to obtain your API key, and make sure to add it into the configuration file (config).

4. Experiment Steps

1) Power on the device and connect to it using MobaXterm.
   (For detailed instructions, please refer to Appendix 5.5: Remote Connection Tools and Instructions.)

2) Navigate to the program directory by entering the following command:

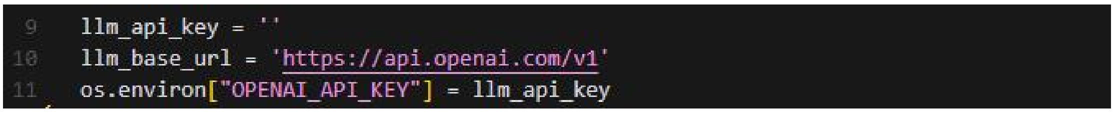
```
cd large_models/
```



```
cd large_models/
```

3) Open the configuration file to input your API Key by entering the command below. Press i to enter INSERT mode and enter your API Key. Once finished, press Esc, type :wq, and hit Enter to save and exit.

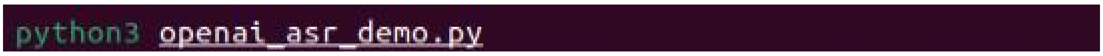
```
vim config.py
```



```
llm_api_key = ''
llm_base_url = 'https://api.openai.com/v1'
os.environ["OPENAI_API_KEY"] = llm_api_key
```

4) Run the speech recognition program with:

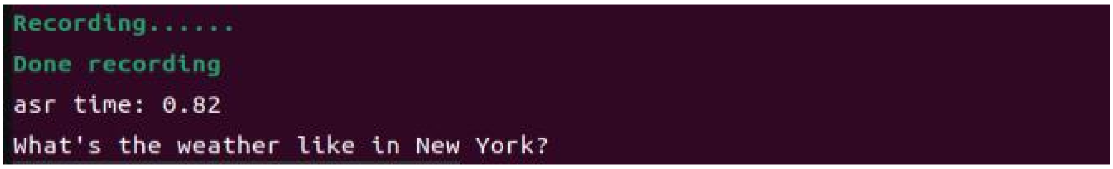
```
python3 openai_asr_demo.py
```



```
python3 openai_asr_demo.py
```

5. Function Realization

After the program starts, the microphone will recognize the recorded audio content from the user and print the converted text output.



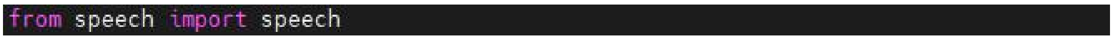
```
Recording......
Done recording
asr time: 0.82
What's the weather like in New York?
```

6. Brief Program Analysis

This program implements a speech recognition system by calling OpenAI's Speech-to-Text API to convert audio files into text.

The program source code is located at: /home/ubuntu/large_models/openai_asr_demo.py

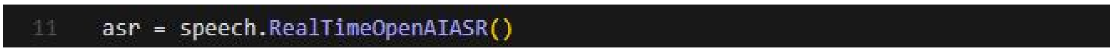
6.1 Module Import



```
from speech import speech
```

The speech module encapsulates ASR (Automatic Speech Recognition) functionalities, such as connecting to an external ASR service.

6.2 Define ASR Class

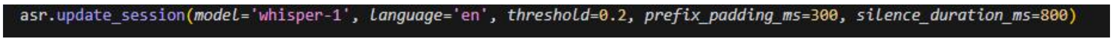


```
asr = speech.RealTimeOpenAIASR()
```

asr = speech.RealTimeOpenAIASR()

This line creates a real-time speech recognition object named asr. The RealTimeOpenAIASR class is used to interact with the speech recognition service.

6.3 Speech Recognition Functionality



```
asr.update_session(model='whisper-1', language='en', threshold=0.2, prefix_padding_ms=300, silence_duration_ms=800)
```

An ASR client object is created to prepare for invoking the speech recognition service.

The asr.asr() method is called to send the audio file (wav) to the ASR service for recognition.

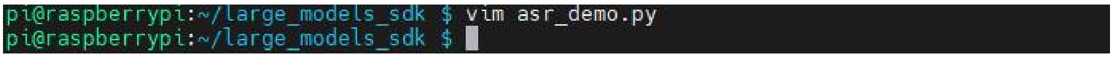
The recognized result (typically text) is printed to the console.

7. Function Extension

You can modify the model name to enable speech recognition in various languages, such as Chinese, English, Japanese, and Korean.

1) Enter the following command to edit the script:

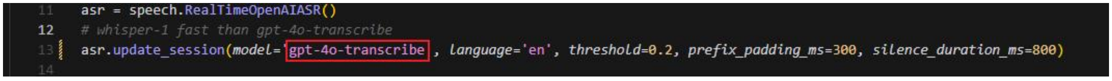
```
vim  openai_asr_demo.py
```



```
pi@raspberrypi:~/large_models_sdk $ vim asr_demo.py
pi@raspberrypi:~/large_models_sdk $ 
```

2) Press the i key to enter INSERT mode, and update the model setting. For example, modify it to use the gpt-4o-transcribe model.

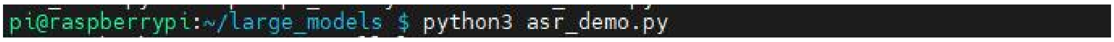
```
i
```



```
asr = speech.RealTimeOpenAIASR()
# whisper-1 fast than gpt-4o-transcribe
asr.update_session(model='gpt-4o-transcribe , language='en', threshold=0.2, prefix_padding_ms=300, silence_duration_ms=800)
```

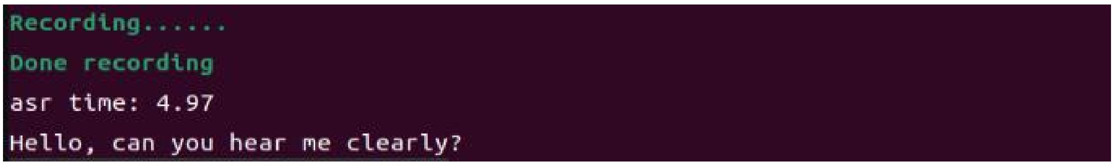
3) Then, run the program with the command:

```
python3  asr_demo.py
```



```
pi@raspberrypi:~/large_models $ python3 asr_demo.py
```

4) Record a sample sentence such as "Hello, can you hear me clearly?", and the recognized text will be printed on the console.



```
Recording......
Done recording
asr time: 4.97
Hello, can you hear me clearly?
```