

1.2.5 Speech Synthesis

1. What is Speech Synthesis?

Speech synthesis (SS) is a technology that converts written text into intelligible spoken audio. It enables computers to generate natural, human-like speech for communication or information delivery.

In this course, we will run a program that processes text using a large language model and generates corresponding audio.

2. How It Works

The program first sends the text to the OpenAI TTS (Text-to-Speech) model. The model returns the generated audio data, which is saved as a file named `tts_audio.wav` for playback or storage.

3. Preparation Before the Experiment

Refer to the course “1 Large Models Basic Courses\ 1.1 Large Language Models Courses\ 1.1.2 Large Language Models Deployment” to obtain your API key, and update the configuration file accordingly.

4. Experiment Steps

- 1) Power on the device and connect to it using MobaXterm (refer to the appendix "5.1 Remote Connection Tools and Instructions" for detailed guidance).
- 2) Navigate to the program directory by entering the following command:

```
cd large_models/
```

```
cd large_models/
```

- 3) Open the configuration file to enter your API Key. After editing, press Esc, type :wq, and hit Enter to save and exit:

```
vim config.py
```

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

- 4) Finally, run the program with the following command:

```
python3 openai_tts_demo.py
```

```
python3 openai_tts_demo.py
```

5. Function Realization

Upon running the program, it will play an audio message saying "Hello, Can I Help You", and simultaneously save the audio file with the same content to the following directory:

/home/ubuntu/large_models/resources/audio/

```
8 tts = speech.RealTimeOpenAITTS()  
9 tts.tts("Hello, Can I help you?") # https://platform.openai.com/docs/guides/text-to-speech  
10 tts.tts("Hello, Can I help you?", model="tts-1", voice="onyx", speed=1.0, instructions="Speak in a cheerful and positive tone.")  
11 tts.save_audio("Hello, Can I help you?", model="gpt-4o-mini-tts", voice="onyx", speed=1.0, instructions="Speak in a cheerful and positive tone.", audio_format='wav',  
12
```

6. Brief Program Analysis

This program is a speech synthesis system based on OpenAI's Text-to-Speech (TTS) API, capable of converting text into audio files. It supports input text and outputs audio in formats like PCM, WAV, FLAC, AAC, Opus, and MP3. By specifying the desired text, the program sends the request to the API and returns the synthesized audio, which can be played or saved locally.

The source code for this program is located at:
/home/ubuntu/large_models/openai_tts_demo.py

6.1 Module Import

```
5 from config import *
6 from speech import speech
```

speech: This module encapsulates the TTS functionalities.

6.2 Definition for TTS Class

```
tts = speech.RealTimeOpenAITTS()
tts.tts("Hello, Can I help you?") # https://platform.openai.com/docs/guides/text-to-speech
tts.tts("Hello, Can I help you?", model="tts-1", voice="onyx", speed=1.0, instructions='Speak in a cheerful and positive tone.')
tts.save_audio("Hello, Can I help you?", model="gpt-4o-mini-tts", voice="onyx", speed=1.0, save_path="./resources/audio/tts_audio.wav")
```

speed: Specifies the playback speed; the default value is 1.

For intelligent real-time applications, it is recommended to use the gpt-4o-mini-tts model.

Other available models include tts-1 and tts-1-hd. tts-1 offers lower latency but with slightly reduced quality compared to tts-1-hd.

Voice Options: nova, shimmer, echo, onyx, fable, alloy, ash, sage, coral.

For more details, you can refer to the OpenAI documentation:

<https://platform.openai.com/docs/guides/text-to-speech>

6.3 Function Extension

To change the voice, follow these steps:

- 1) Open the program by entering the command:

```
vim openai_tts_demo.py
```

```
vim openai_tts_demo.py
```

- 2) Press i on your keyboard to enter INSERT mode. Locate the line voice="onyx" and modify it to voice="nova".

```
i
```

```
~
~
~
-- INSERT --                               32,1      AL
```

```
10 tts.tts("Hello, Can I help you?", model="tts-1", voice="nova", speed=1.0, instructions='Speak in a cheerful and positive tone.')
11 tts.save_audio("Hello, Can I help you?", model="gpt-4o-mini-tts", voice="nova", speed=1.0, save_path="./resources/audio/tts_audio.wav")
12
```

- 3) Press Esc, then type :wq and hit Enter to save and exit.

```
:wq
```

```
~
:wq
```

- 4) Execute the program with the following command:

```
python3 openai_tts_demo.py
```

```
www.hiwonder.com
Time to first byte: 2.25 seconds
[Playback Thread] Playback stream stopped and closed.
[TTS] Playback task complete.
Time to first byte: 1.85 seconds
ALSA lib pcm.c:8568:(snd_pcm_recover) underrun occurred
[Playback Thread] Playback stream stopped and closed.
[TTS] Playback task complete.
```

Once the program starts, the speaker will play the synthesized audio using the newly selected voice style.