Audio Transcription with Whisper ASR

This code demonstrates how to transcribe audio segments using the OpenAI Whisper ASR model. It provides a step-by-step guide on how to download audio, split it into segments, transcribe using the Whisper ASR model, generate a VTT file for captions, and store conversation data.

Prerequisites

Before running the code, ensure that you have the following packages installed:
- pydub
- requests
- openai
- whisper

You can use pip to install these packages:
```bash
pip install pydub requests openai whisper
```

Now, let's explore the steps involved in transcribing audio segments using the Whisper ASR model.

Step 1: Audio Download

The code starts by downloading an audio file from Google Drive using its file ID. This allows you to retrieve the audio file you want to transcribe.

Step 2: Audio Playback

After downloading the audio file, the code uses IPython to play the audio and verify its content. This step ensures that the correct audio file has been downloaded.

Step 3: Audio Splitting

The audio file is then split into two segments: caller and callee. Each segment contains half of the conversation. This splitting allows for more accurate transcription of each speaker's dialogue.

Step 4: Whisper ASR Model

The Whisper ASR model is loaded, and the audio segments are transcribed using the model. Whisper is a state-of-the-art automatic speech recognition (ASR) system developed by OpenAI. It provides accurate and reliable transcriptions for various applications.

Step 5: VTT File Generation

The transcriptions obtained from the Whisper ASR model are saved in WebVTT (VTT) format. VTT files are commonly used for captioning and subtitles. This step enables you to generate caption files for your audio segments.

Step 6: Conversation Data Storage

Finally, the transcriptions and paths to the audio segments are stored in a JSON file. This allows for further analysis or use of the conversation data. Storing the data in a structured format facilitates easy retrieval and manipulation.

By following these steps, you can effectively transcribe audio segments using the Whisper ASR model, generate VTT files for captions, and store conversation data for further analysis or use.

Usage:

1.To set up the Whisper ASR model, you will need to provide the appropriate model path. This model will be used to transcribe audio segments. Ensure that you have the caller and callee audio segments ready and set their paths in the code.

2.Once you have set up the model path and provided the audio segment paths, you can run the code. Running the code will transcribe the audio segments using the Whisper ASR model and save the transcriptions in VTT format.

3.The conversation data, including the transcriptions and audio segment paths, will be stored in a JSON file. This JSON file can be used for various applications, such as generating subtitles or analyzing the content of the conversation.

4.By using the transcriptions and data from the JSON file, you can perform tasks like generating subtitles for the conversation or analyzing the content of the conversation for further insights. This can be useful in applications like transcription services, call center analysis, or speech recognition research.

License:

This code is made available to you under an open-source license. You have the freedom to modify and utilize it as per your requirements. Feel free to make any changes or enhancements to suit your needs.

We hope this code proves useful to you in your coding endeavors. It is our intention to promote collaboration and innovation by providing this open-source solution.

Remember to adhere to the terms and conditions of the license while using this code. If you have any questions or need assistance, don't hesitate to reach out.

Make sure to customize the README file to include any additional details or usage instructions specific to your project.

Happy coding!