AI Audio Correction with Dynamic Sentence Alignment - Documentation

Overview

This Streamlit application processes video files to extract audio, transcribe it using Google Cloud Speech-to-Text, correct the transcription using OpenAI, and generate new audio that aligns dynamically with the original video's timing. The final output is a video with the corrected audio.

Features

- Upload video files in formats like MP4, MOV, and AVI.
- Extract audio from video and convert it to mono.
- Transcribe audio with word-level timestamps.
- Correct transcriptions for grammar and filler words.
- Generate AI audio from corrected transcriptions with dynamic timing.
- Replace the original audio in the video with the generated audio.

Requirements

To run this application, ensure you have the following Python packages installed:

- `streamlit`
- `openai`
- `wave`
- `google-cloud-speech`
- `google-cloud-texttospeech`
- `moviepy`
- `pydub`
- `python-dotenv`
- `requests`

You can install these packages using pip:

```
pip install -r requirements.txt
```

Google Cloud Setup

1. **Service Account Credentials**: Create a Google Cloud project and enable the Speech-to-Text and Text-to-Speech APIs. Generate a service account key and save the credentials in a secure location.

Environment Variables: Set the following environment variables in a `.env` file:

```
GOOGLE_CREDENTIALS_FILE=<path_to_your_credentials_file.json>
CREDS=<your_google_service_account_json_content>
OPENAI_KEY=<your_openai_api_key>
OPENAI_ENDPOINT=<your_openai_api_endpoint>
```

Usage Instructions

Run the Application:
Start the Streamlit application by executing the following command in

```
streamlit run main.py
```

1. **Upload a Video File**: Use the file uploader in the application interface to upload a video file (MP4, MOV, or AVI).
2. **Processing**: The application will extract the audio from the video, transcribe it, correct the transcription, generate new audio, and replace the original audio in the video.
3. **Output**: After processing, the application displays the output video with the corrected and dynamically synchronized audio.

Function Descriptions

1. `convert_to_mono(audio_path)`

Converts the input audio file to mono format and saves it as `mono_audio.wav`.

2. `transcribe_audio(audio_path)`

Transcribes the audio file using Google Cloud Speech-to-Text and returns the transcription text.

3. `generate_audio(text)`

Generates audio from the provided text using Google Cloud Text-to-Speech and saves it as `generated_audio.wav`.

4. `adjust_audio_length(audio_path, video_path)`

Adjusts the length of the audio file to match the duration of the video file by adding silence or trimming.

5. `replace_audio_in_video(video_path, audio_path, output_path)`

Replaces the original audio in the video with the new audio file and saves it.

6. `transcribe_audio_with_word_timestamps(audio_path)`

Transcribes the audio while capturing word-level timestamps, returning a list of sentences with their respective start and end times.

7. `correct_transcription(transcription_text)`

Sends the transcription text to OpenAI for grammar correction and returns the corrected text.

8. `generate_audio_with_dynamic_ssml(corrected_sentences, original_sentences)`

Generates audio using SSML that dynamically matches the timing of the original sentences based on their timestamps.

Overview:-

The basic logic involves getting the timestamps of the original video's sentences (start and end), and then getting the correct transcription for the respective sentences. Then, mapping the corrected transcripts to the original video using certain algorithms(this part can be optimized) to the final video.