Project Documentation: OpenAI Whisper Model Implementation in Delphi

Introduction

What is OpenAI Whisper?

Whisper is an advanced Automatic Speech Recognition (ASR) system developed by OpenAI. This model is designed to transcribe spoken audio into written text with high accuracy and efficiency, even in challenging conditions like background noise, multiple speakers, and different accents and dialects. Whisper is multilingual and can handle various languages, making it highly versatile for global applications.

What is Whisper used for?

Whisper is used in various applications requiring audio transcription and processing, including:

- **Automatic Video Subtitling:** Facilitates accurate subtitle generation for video content, improving accessibility and comprehension.
- **Virtual Assistants and Chatbots:** Enables virtual assistants and chatbots to understand and respond to user voice commands.
- **Meeting and Conference Transcription:** Automates transcription of discussions and presentations, aiding in documentation and subsequent analysis.
- **Accessibility Enhancement:** Helps people with hearing disabilities by providing accurate audio transcriptions.
- **Audio Content Analysis:** Facilitates spoken content analysis for applications in research, marketing, and other fields.

Whisper Functionalities in this Project

In this project, we implement the following Whisper model functionalities using Delphi:

- **Audio Transcription:** Converts spoken audio to written text.
- **Audio Translation:** Translates audio content from one language to another.
- **Text-to-Speech:** Generates spoken audio from written text.

These functionalities enable the creation of powerful applications that can interact efficiently with users through speech.

TAiAudio Component Description

TAiAudio is a Delphi class designed to interact with OpenAI's Whisper model. It provides methods for transcription, translation, and text-to-speech from audio files. Below is a detailed description of its structure, properties, and methods.

Component Properties

TAiAudio component properties allow configuration of various parameters needed to interact with the OpenAI API:

- **ApiKey:** API key used to authenticate OpenAI requests.
- **Url:** OpenAI API base URL.
- **Model:** Transcription or text-to-speech model used.
- **Voice:** Voice used for text-to-speech synthesis.
- **Format:** Audio file output format (e.g., mp3, opus, aac, flac, pcm).
- **Language:** Audio language.
- **Speed:** Speech speed in voice synthesis.
- **Temperature:** Temperature used for text generation, influencing model creativity.
- **ResponseFormat:** Response format (e.g., json, text, srt, verbose_json, vtt).
- **Quality:** Text-to-speech model quality (e.g., tts-1, tts-1-hd).

Component Methods

TAiAudio provides several methods to interact with the OpenAI API:

- **Speech:** Generates an audio file from text.
- **Transcription:** Transcribes audio file content to text.
- **Translation:** Translates audio file content from one language to another.

Constructor and Destructor

- **Create:** Initializes a TAiAudio component instance, configuring default properties.
- **Destroy:** Releases resources used by the component instance.

Helper Methods

- **IsValidExtension:** Verifies if a file extension is valid for processing.
- **ConvertFileFormat:** Converts audio file format using ffmpeg.

Usage Example

```
var
  AiAudio: TAiAudio;
  TranscriptionText: String;
  AudioStream: TMemoryStream;
begin
  AiAudio := TAiAudio.Create(nil);
  try
```

```
    AiAudio.ApiKey := 'your_api_key';
    AudioStream := TMemoryStream.Create;
    try
      AudioStream.LoadFromFile('path/to/your/audio/file.mp3');
      TranscriptionText      :=      AiAudio.Transcription(AudioStream,
'file.mp3', 'This is an example prompt');
      ShowMessage(TranscriptionText);
    finally
      AudioStream.Free;
    end;
  finally
    AiAudio.Free;
  end;
end;
```

Detailed TAiAudio Component Properties

Properties

1. **ApiKey:**
   - Description: API key used to authenticate OpenAI requests
   - Type: String
   - Usage: `AiAudio.ApiKey := 'your_api_key';`
2. **Url:**
   - Description: OpenAI API base URL
   - Type: String
   - Usage: `AiAudio.Url := 'https://api.openai.com/v1/';`
3. **Model:**
   - Description: Transcription or text-to-speech model used
   - Type: String
   - Values: whisper-1 for transcriptions and tts for text-to-speech
   - Note: 'tts-1' for speed, 'tts-1-hd' for higher quality
   - Usage: `AiAudio.Model := 'tts-1';`
4. **Voice:**
   - Description: Voice used for text-to-speech synthesis
   - Type: String
   - Values: 'alloy', 'echo', 'fable', 'onyx', 'nova', 'shimmer'
   - Usage: `AiAudio.Voice := 'nova';`
5. **Format:**
   - Description: Audio file output format
   - Type: String
   - Values: 'mp3', 'opus', 'aac', 'flac', 'pcm'
   - Usage: `AiAudio.Format := 'mp3';`
6. **Language:**
   - Description: Audio language
   - Type: String
   - Usage: `AiAudio.Language := 'en';`

7. **Speed:**
    - Description: Speech speed in text-to-speech synthesis
    - Type: Single (range 0.25 to 4.0, default 1)
    - Usage: `AiAudio.Speed := 1.0;`
8. **Temperature:**
    - Description: Temperature for text generation
    - Type: Single (range 0 to 1, 0 more strict, 1 more random)
    - Usage: `AiAudio.Temperature := 0.7;`
9. **ResponseFormat:**
    - Description: Response format
    - Type: String
    - Values: 'json', 'text', 'srt', 'verbose_json', 'vtt'
    - Usage: `AiAudio.ResponseFormat := 'text';`
10. **Quality:**
    - Description: Text-to-speech model quality
    - Type: String
    - Values: 'tts-1', 'tts-1-hd'
    - Usage: `AiAudio.Quality := 'tts-1';`

Detailed TAiAudio Component Methods

Speech

- Description: Generates audio file from text
- Parameters:
    - aText: Text to convert to audio
    - aVoice (optional): Voice for synthesis
- Returns: TMemoryStream with generated audio file
- Example:

```
var
  AudioStream: TMemoryStream;
begin
  AudioStream := AiAudio.Speech('Hello, this is a voice synthesis
example.');
  try
    AudioStream.SaveToFile('output.mp3');
  finally
    AudioStream.Free;
  end;
end;
```

Transcription

- Description: Transcribes audio file content to text
- Parameters:
    - aStream: Memory stream containing audio file

    - aFileName: Audio file name
    - aPrompt: Transcription model prompt
- Returns: String with transcribed text
- Example:

```
var
  TranscriptionText: String;
begin
  TranscriptionText  :=  AiAudio.Transcription(AudioStream,  'file.mp3',
'This is an example prompt');
  ShowMessage(TranscriptionText);
end;
```

Translation

- Description: Translates audio file content between languages
- Parameters:
    - aStream: Memory stream containing audio file
    - aFileName: Audio file name
    - aPrompt: Translation model prompt
- Returns: String with translated text
- Example:

```
var
  TranslationText: String;
begin
  TranslationText := AiAudio.Translation(AudioStream, 'file.mp3', 'This
is an example prompt');
  ShowMessage(TranslationText);
end;
```

Pending Implementation

1. TTS streaming functionality is yet to be implemented.