DistilWhisperModel Documentation

Overview

The `DistilWhisperModel` is a Python class designed to handle English speech recognition tasks. It leverages the capabilities of the Whisper model, which is fine-tuned for speech-to-text processes. It is designed for both synchronous and asynchronous transcription of audio inputs, offering flexibility for real-time applications or batch processing.

Installation

Before you can use `DistilWhisperModel`, ensure you have the required libraries installed:

```sh
pip3 install --upgrade swarms
```

Initialization

The `DistilWhisperModel` class is initialized with the following parameters:

Parameter	Type	Description	Default
`model_id`	`str`	The identifier for the pre-trained Whisper model	`"distil-whisper/distil-large-v2"`

Example of initialization:

```python
from swarm_models import DistilWhisperModel

# Initialize with default model
model_wrapper = DistilWhisperModel()

# Initialize with a specific model ID
model_wrapper = DistilWhisperModel(model_id="distil-whisper/distil-large-v2")
```

Attributes

After initialization, the `DistilWhisperModel` has several attributes:

Attribute	Type	Description
`device`	`str`	The device used for computation (`"cuda:0"` for GPU or `"cpu"`).
`torch_dtype`	`torch.dtype`	The data type used for the Torch tensors.
`model_id`	`str`	The model identifier string.
`model`	`torch.nn.Module`	The actual Whisper model loaded from the identifier.
`processor`	`transformers.AutoProcessor`	The processor for handling input data.

Methods

`transcribe`

Transcribes audio input synchronously.

Arguments:

Argument	Type	Description
`inputs`	`Union[str, dict]`	File path or audio data dictionary.

Returns: `str` - The transcribed text.

Usage Example:

```python
# Synchronous transcription
transcription = model_wrapper.transcribe("path/to/audio.mp3")
print(transcription)
```

`async_transcribe`

Transcribes audio input asynchronously.

Arguments:

| Argument | Type | Description |

|----------|------|-------------|

| `inputs` | `Union[str, dict]` | File path or audio data dictionary. |

Returns: `Coroutine` - A coroutine that when awaited, returns the transcribed text.

Usage Example:

```python
import asyncio

# Asynchronous transcription
transcription = asyncio.run(model_wrapper.async_transcribe("path/to/audio.mp3"))
print(transcription)
```

`real_time_transcribe`

Simulates real-time transcription of an audio file.

Arguments:

Argument	Type	Description
`audio_file_path`	`str`	Path to the audio file.
`chunk_duration`	`int`	Duration of audio chunks in seconds.

Usage Example:

```python
# Real-time transcription simulation
model_wrapper.real_time_transcribe("path/to/audio.mp3", chunk_duration=5)
```

Error Handling

The `DistilWhisperModel` class incorporates error handling for file not found errors and generic exceptions during the transcription process. If a non-recoverable exception is raised, it is printed to the console in red to indicate failure.

Conclusion

The `DistilWhisperModel` offers a convenient interface to the powerful Whisper model for speech recognition. Its design supports both batch and real-time transcription, catering to different application needs. The class's error handling and retry logic make it robust for real-world applications.

Additional Notes

- Ensure you have appropriate permissions to read audio files when using file paths.
- Transcription quality depends on the audio quality and the Whisper model's performance on your dataset.
- Adjust `chunk_duration` according to the processing power of your system for real-time

transcription.

For a full list of models supported by `transformers.AutoModelForSpeechSeq2Seq`, visit the [Hugging Face Model Hub](https://huggingface.co/models).