

Whisper: AI-Powered Speech-to-Text

CyFuture Assignment

Presented by :-
Aashish Singune

NIT Agartala

25 march, 2025

Outline

Introduction
Technology Used
Functioning of the System
Snapshots
Conclusion

Introduction

Technology Used

Functioning of the System

Snapshots

Conclusion

Introduction

- ▶ **Introduction:** Whisper AI is a powerful automatic speech recognition (ASR) model developed by OpenAI.
- ▶ **High Accuracy:** It transcribes audio into text with precision, even in noisy environments or with diverse accents.
- ▶ **Multilingual Support:** The model supports multiple languages, making it ideal for global applications.
- ▶ **Versatile Applications:** Used for transcription, subtitles, voice assistants, and accessibility solutions.
- ▶ **Deep Learning-Based:** Leverages advanced AI techniques to process spoken language efficiently.
- ▶ **Flexible Integration:** Supports various audio formats and durations for real-time and offline processing.

Technology Used in Whisper

- ▶ **Speech Recognition:** Whisper AI for high-accuracy audio-to-text conversion.
- ▶ **Multi-Language Support:** Whisper AI supports multiple languages for diverse user needs.
- ▶ **Use Cases:** Transcription, accessibility, subtitles, and automation using AI.
- ▶ **Deep Learning-Based:** Utilizes neural networks for efficient speech processing.
- ▶ **Open-Source Integration:** Easily integrates with applications for AI-driven speech-to-text functionality.

Functioning of the System

Whisper AI converts audio files into text with high accuracy and multilingual support.

Audio Processing:

- ▶ Users upload an MP3 file for transcription.
- ▶ The model converts speech to text efficiently.

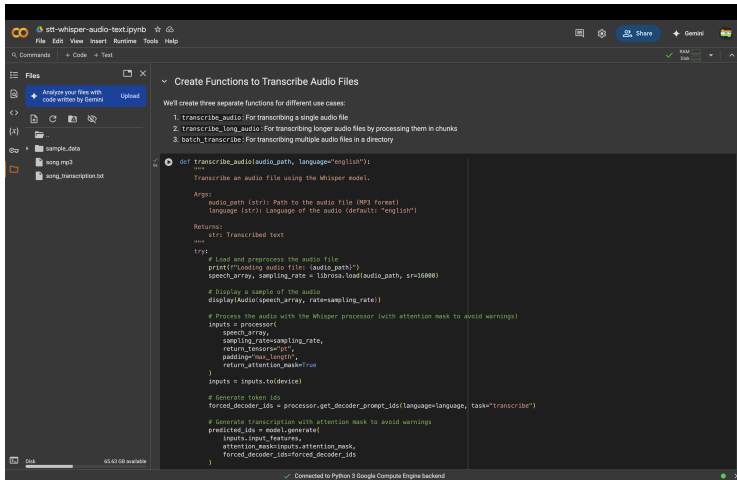
Key Features:

- ▶ Supports multiple languages and accents.
- ▶ Handles background noise effectively.
- ▶ Enables real-time transcription.

Use Cases:

- ▶ Automated note-taking and subtitles.
- ▶ Accessibility and voice command applications.

Function to Transcribe Audio files



```
def transcribe_audio(audio_path, language="english"):
    """
    Transcribe an audio file using the Whisper model.

    Args:
        audio_path (str): Path to the audio file (MP3 format)
        language (str): Language of the audio (default: "english")

    Returns:
        str: Transcribed text
    """
    try:
        # Load and preprocess the audio file
        print(f"Loading audio file: {audio_path}")
        speech_array, sampling_rate = librosa.load(audio_path, sr=16000)

        # Display a sample of the audio
        display(Audio(speech_array, rate=sampling_rate))

        # Process the audio with the Whisper processor (with attention mask to avoid warnings)
        inputs = processor(
            speech_array,
            sampling_rate=sampling_rate,
            return_tensors="pt",
            padding="max_length",
            return_attention_mask=True
        )
        inputs = inputs.to(device)

        # Generate token ids
        forced_decoder_ids = processor.get_decoder_prompt_ids(language=language, task="transcribe")

        # Generate transcription with attention mask to avoid warnings
        predicted_ids = model.generate(
            inputs.input_features,
            attention_mask=inputs.attention_mask,
            forced_decoder_ids=forced_decoder_ids
        )
```

Transcribe Audio in a Different language

The screenshot shows a Jupyter Notebook interface with a dark theme. The title bar indicates the file is 'stt-whisper-audio-text.ipynb'. The left sidebar shows a file explorer with a folder named 'sample_data' containing 'esorg.mp3' and 'esorg_transcription.txt'. The main area displays the notebook content for 'Example 4: Transcribe Audio in a Different Language'. It includes a description: 'Use this example to transcribe audio in a language other than English.' and a Python code block for transcribing audio in Spanish. The code handles both short and long audio files by chunking. Below the code is a 'Conclusion' section summarizing the notebook's purpose and listing key features and potential improvements.

```
[ ] # Path to your non-English audio file
non_english_audio_path = 'path_to_your_non_english_audio.mp3' # Replace with your actual file path

# Check if the file exists
if os.path.exists(non_english_audio_path):
    # Choose the appropriate function based on audio length
    # For shorter audio files:
    transcription = transcribe_audio(non_english_audio_path, language="spanish") # Change language as needed

    # For longer audio files:
    transcription = transcribe_long_audio(non_english_audio_path, chunk_length_sec=30, language="spanish")

# Print the transcription
print("\nTranscription:")
print(transcription)

# Save the transcription to a text file
output_file = os.path.splitext(non_english_audio_path)[0] + "_transcription.txt"
with open(output_file, "w", encoding="utf-8") as f:
    f.write(transcription)

print(f"\nTranscription saved to: {output_file}")
else:
    print(f"File not found: {non_english_audio_path}")
```

Conclusion

This notebook demonstrates how to use the Whisper model from Hugging Face to transcribe audio files in MP3 format to text. You can use this as a starting point for your audio transcription projects.

Key features implemented:

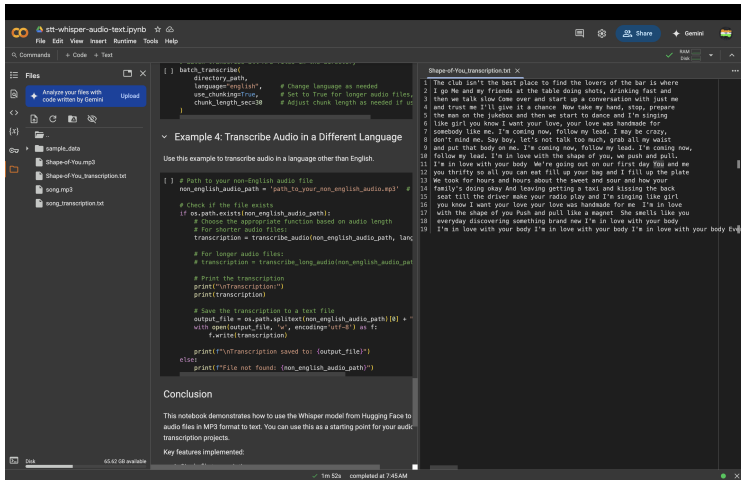
1. Single file transcription
2. Batch processing of multiple files
3. Support for different languages
4. Handling of longer audio files through chunking
5. Proper attention mask handling to avoid warnings

Additional improvements you could make:

1. Add support for more audio formats (WAV, FLAC, etc.)

66.43 GB available
Connected to Python 3 Google Compute Engine backend

Whisper AI - Output



```
batch_transcribe(
    directory_path,
    language="english",  # Change language as needed
    use_chunking=True,    # Set to True for longer audio files,
    chunk_length_secs=30  # Adjust chunk length as needed if use
```

Example 4: Transcribe Audio in a Different Language

Use this example to transcribe audio in a language other than English.

```
1 # Path to your non-English audio file
2 non_english_audio_path = 'path_to_your_non_english_audio.mp3' #
3
4 # Check if the file exists
5 if os.path.exists(non_english_audio_path):
6     # Choose the appropriate function based on audio length
7     # For shorter audio files:
8     transcription = transcribe_audio(non_english_audio_path, lang
9
10    # For longer audio files:
11    transcription = transcribe_long_audio(non_english_audio_pat
12
13    # Print the transcription
14    print("\nTranscription:")
15    print(transcription)
16
17    # Save the transcription to a text file
18    output_file = os.path.splitext(non_english_audio_path)[0] + ".txt"
19    with open(output_file, "w", encoding="utf-8") as f:
20        f.write(transcription)
21
22    print(f"\nTranscription saved to: {output_file}")
23 else:
24    print(f"File not found: {non_english_audio_path}")
```

Conclusion

This notebook demonstrates how to use the Whisper model from Hugging Face to audio files in MP3 format to text. You can use this as a starting point for your audio transcription projects.

Key features implemented:

1m 52s completed at 7:45 AM

Conclusion

- ▶ **Accurate Transcription:** Converts speech to text with high precision.
- ▶ **Multilingual Support:** Handles multiple languages and accents effectively.
- ▶ **Real-Time Processing:** Enables fast and efficient speech-to-text conversion.
- ▶ **Wide Applications:** Useful for accessibility, subtitles, and automated note-taking.

Thank You!