# Milestone 2: Speech to Text

In this milestone, we developed a Speech-to-Text (STT) system using a pre-trained transformer model. The main objective of this milestone was to convert spoken audio into text accurately using modern deep learning models.

## Objective

To build a functional automatic speech recognition (ASR) system capable of transcribing speech into text using the OpenAI Whisper pre-trained model. This milestone follows the audio cleaning phase from Milestone 1 and focuses on leveraging a reliable ASR model to produce accurate text transcriptions.

## Tools and Libraries Used

- Google Colab (for implementation)  
- Transformers library from Hugging Face  
- Torch (PyTorch backend)  
- JiWER (for computing Word Error Rate)  
- OpenAI Whisper pre-trained model (for ASR pipeline)

## Implementation Steps

1. \*\*Import Required Libraries:\*\*

We imported necessary Python libraries such as transformers, torch, and jiwer. These libraries support model loading, GPU utilization, and performance evaluation.

2. \*\*Load Pre-trained Model:\*\*

We used the OpenAI Whisper model (‘openai/whisper-small’) through the Hugging Face `pipeline()` function. The model was loaded as follows:  
  
```python  
from transformers import pipeline  
import torch  
asr\_pipeline = pipeline(  
 'automatic-speech-recognition',  
 model='openai/whisper-small',  
 device=0 if torch.cuda.is\_available() else -1  
)  
```  
This checks for GPU availability and assigns the model to it for faster inference.

3. \*\*Load and Process Audio File:\*\*

The audio file was loaded from Google Drive or local storage. The model automatically handled resampling and decoding, eliminating the need for manual preprocessing.

4. \*\*Speech to Text Conversion:\*\*

The Whisper ASR pipeline was used to transcribe the input audio file into text:  
  
```python  
result = asr\_pipeline('/content/drive/MyDrive/auido\_mono.wav')  
transcript = result['text']  
print('Transcript:', transcript)  
```  
This produced a text transcript corresponding to the spoken words in the audio.

5. \*\*Evaluation using WER (Word Error Rate):\*\*

The transcription was compared to a ground truth sentence to measure accuracy using the Word Error Rate (WER):  
  
```python  
from jiwer import wer  
ground\_truth\_text = 'this is the ground truth sentence for my audio'  
error = wer(ground\_truth\_text.lower(), transcript.lower())  
print(f'Word Error Rate: {error \* 100:.2f}%')  
```  
A lower WER indicates a more accurate transcription.

## Results

The Whisper model successfully transcribed the input audio into text. The transcribed output was compared with a reference sentence to evaluate accuracy. Even though minor variations were observed depending on accent, pronunciation, or background noise, the Whisper model provided robust and reliable transcription performance.

## Conclusion

In this milestone, a complete end-to-end speech-to-text system was implemented using the OpenAI Whisper pre-trained model. This model demonstrated strong performance in transcribing English audio samples. The next milestone will focus on integrating this system into a frontend interface using Streamlit, allowing users to upload audio and receive transcribed text interactively.