The image describes Task 2: Speech Recognition System with the following instructions:

Objective: Build a basic speech-to-text system using pre-trained models and libraries such as SpeechRecognition or Wav2Vec.

Deliverable: A functional system capable of transcribing short audio clips.

---

Python Script: Speech-to-Text System

Here’s a Python implementation for this task using the SpeechRecognition library:

---

Using SpeechRecognition Library

# Install necessary library (uncomment if not installed)

# !pip install SpeechRecognition pydub

import speech\_recognition as sr

# Initialize the recognizer

recognizer = sr.Recognizer()

# Function to transcribe audio

def transcribe\_audio(file\_path):

try:

# Load the audio file

with sr.AudioFile(file\_path) as source:

print("Processing audio...")

audio\_data = recognizer.record(source)

# Recognize speech using Google Web Speech API

text = recognizer.recognize\_google(audio\_data)

print("Transcription:")

print(text)

except Exception as e:

print("Error:", str(e))

# Input: Path to the audio file (replace 'audio.wav' with your file path)

audio\_file = "audio.wav"

# Call the function

transcribe\_audio(audio\_file)

---

Using Wav2Vec for Advanced Transcription

# Install necessary library (uncomment if not installed)

# !pip install torch transformers librosa

from transformers import Wav2Vec2ForCTC, Wav2Vec2Processor

import torch

import librosa

# Load pre-trained Wav2Vec model and processor

model = Wav2Vec2ForCTC.from\_pretrained("facebook/wav2vec2-base-960h")

processor = Wav2Vec2Processor.from\_pretrained("facebook/wav2vec2-base-960h")

# Function to transcribe audio using Wav2Vec

def wav2vec\_transcribe(file\_path):

try:

# Load and process audio file

audio, rate = librosa.load(file\_path, sr=16000)

input\_values = processor(audio, sampling\_rate=rate, return\_tensors="pt").input\_values

# Perform inference

logits = model(input\_values).logits

predicted\_ids = torch.argmax(logits, dim=-1)

# Decode transcription

transcription = processor.decode(predicted\_ids[0])

print("Transcription:")

print(transcription)

except Exception as e:

print("Error:", str(e))

# Input: Path to the audio file (replace 'audio.wav' with your file path)

audio\_file = "audio.wav"

# Call the function

wav2vec\_transcribe(audio\_file)

---

How It Works

1. SpeechRecognition Method:

Uses the recognize\_google() method from the SpeechRecognition library.

Processes short audio clips effectively.

2. Wav2Vec Method:

Utilizes Facebook’s Wav2Vec2 pre-trained model for advanced transcription.

Suitable for higher accuracy and handling more complex audio.

---

Deliverable

A functional script that:

Reads an audio file (e.g., audio.wav).

Outputs the transcribed text to the console.