AI-POWERED INTERACTIVE LEARNING ASSISTANT FOR CLASSROOMS

INTRODUCTION

Objective:

To enhance student engagement using an intelligent assistant that:

- Accepts voice, text, and document inputs.
- Runs locally (no need for high-end GPU or constant internet).
- Leverages OpenVINO for optimized performance.

MODEL ARCHITECTURE

Main Components:

- Text Model: Phi-3 Mini 4k Instruct (Microsoft)
- Speech-to-Text: Whisper Base (OpenAI)
- PDF Processing: PyPDF2
- UI/UX Layer: CustomTkinter

Flow:

1. User Input (Text/Voice/PDF)
2. Preprocessing
3. Prompt construction (contextualized)
4. Phi-3 generates response
5. Display via GUI

LIBRARIES USED

Primary Libraries:

- transformers, optimum-intel: for model inference
- torch, numpy: backend tensor & math ops
- sounddevice: audio recording
- PyPDF2: document text extraction
- customtkinter: GUI interface
- openai-whisper: voice transcription model

Custom Implementation:

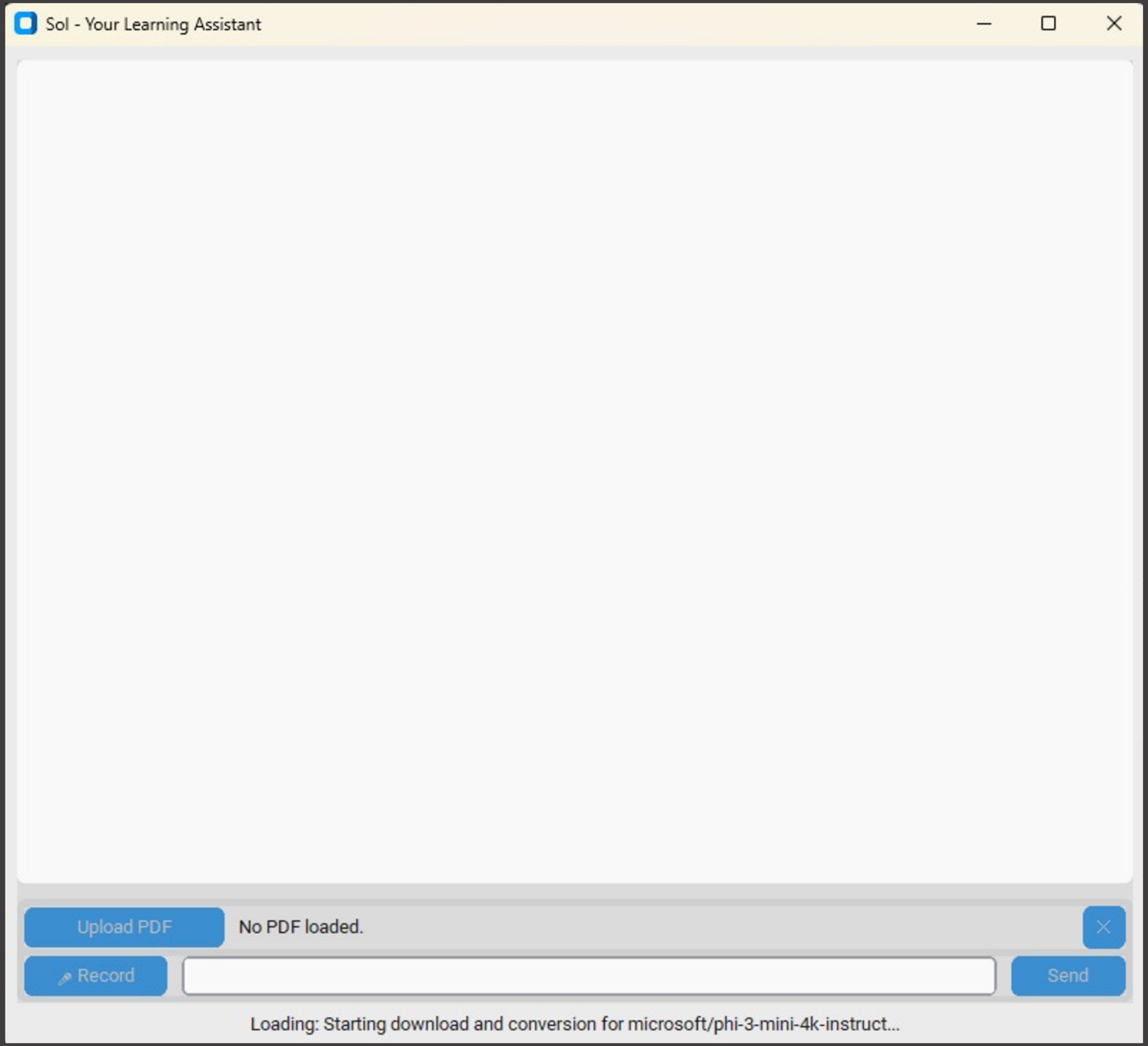
- PDF-to-context pipeline
- Audio transcription thread management
- GUI state updates and prompt orchestration
- Dynamic multi-modal input handling

How the AI Assistant Works

1. **User Input**
    - ○ The user can type a question, speak into the mic, or upload a PDF.
2. **Input Processing**
    - ○ ✍️ Text goes directly to the language model
    - ○ 🎤 Voice is converted to text using the Whisper model
    - ○ 📄 PDF content is extracted using PyPDF2
3. **Context Building**
    - ○ The system combines the current question with past conversation and PDF content.
4. **Response Generation**
    - ○ The processed input is sent to Phi-3 Mini, which generates a relevant, friendly answer.
5. **Output Display**
    - ○ The response appears in a chat window inside the CustomTkinter GUI.

(All of this happens locally, optimized using Intel OpenVINO, without needing the internet.)

Sample Response Generated by “Sol”



Loading: Starting download and conversion for microsoft/phi-3-mini-4k-instruct...

THANK YOU!

BY,

GOVIND KRISHNA C
FARHANA SALAM