Multimodal RAG API

Multimodal REST API for retrieval and response generation based on text, audio, and images using FastAPI.

Features

- Support for text and image (OCR) input questions.
- Audio transcription with Whisper.
- Semantic search over knowledge base using Sentence Transformers + Faiss.
- Response generation via language model (transformers).
- Speech synthesis with gTTS for audio responses.

Technologies Used

- FastAPI
- Whisper (OpenAI)
- Sentence Transformers
- Faiss
- Transformers
- PyPDF2
- Tesseract OCR
- gTTS
- Pillow

Steps

Receive questions via audio, text, or image

- Functions:
  - `support_endpoint` — receives text and image via HTTP POST form.
  - `support_audio_endpoint` — receives audio and image via HTTP POST.
- **Transcribe audio to text (mandatory Speech-to-Text)**
- Tool: **Whisper**
- Function:
  - `transcribe_audio(file_path)` — uses Whisper model to convert audio into text.
- **Search a knowledge base (RAG) for relevant information**
- Tools: **Sentence Transformers + Faiss**
- Functions:
  - `build_index_from_kb()` — builds the FAISS vector index from knowledge base documents.
  - `retrieve(query, top_k)` — performs semantic search on the index to get relevant text chunks.
- **Process screenshots or captures (V2V / OCR)**
- Tool: **Tesseract OCR**
- Function:
  - `ocr_image(image_path)` — extracts text from images using OCR.
- **Generate a technical response (LLM)**
- Tool: **Transformers (text generation pipeline)**
- Functions:
  - `load_llm()` — loads the language model for text generation.
  - `generate_answer(question, context_docs)` — generates a response based on the question and retrieved context.
- **Convert that response to voice (mandatory Text-to-Speech)**
- Tool: **gTTS (Google Text-to-Speech)**

- Function:
    - `text_to_speech(text, out_path)` — converts the generated answer text into an MP3 audio file.
- **Return the response in both text and audio**
- Function:
    - The FastAPI endpoints `/support` and `/support/audio` return a JSON with:
        - `transcription` (transcribed audio text)
        - `ocr_text` (extracted text from image)
        - `answer` (generated answer)
        - `audio_url` (link to the answer audio)
        - `source_documents` (list of source documents used)

Local Setup

1.Clone the repository:

```
git clone https://github.com/yourusername/yourrepository.git
cd yourrepository
```

2.(Optional but recommended) Create and activate a virtual environment:

```
python -m venv venv
source venv/bin/activate  # Linux/Mac
venv\Scripts\activate     # Windows
```

3.Install dependencies:

```
pip install -r requirements.txt
```

4.Configure ffmpeg in your system PATH (required for Whisper and TTS).

5.Run the API:

```
uvicorn main:app --reload --host 0.0.0.0 --port 8000
```

6.Test the endpoints `POST /support` and `POST /support/audio` as per the documentation.

Running with Docker

1.Build the Docker image:

```
docker build -t multimodal-rag-api .
```

2.Run the container:

```
docker run -p 8000:8000 multimodal-rag-api
```

3.The API will be available at `http://localhost:8000`

Examples

The `examples` folder contains `.mp3` files generated as sample responses for typical queries.

Notes

- Requires ffmpeg installed and configured in the system PATH.
- Audio input accepts `.mp3` or `.wav` files.
- Image input accepts `.jpg` or `.png` files.

Example API Response

```
{
  "transcription": "Mi pantalla se queda en negro al iniciar.",
  "ocr_text": null,
  "answer": "Por favor verifique si su tarjeta gráfica está bien conectada.",
  "audio_url": "/static/response_2a424c2a6a7445e5a9a0634b7dede4dd.mp3",
  "source_documents": [
    "Errores Comunes en SoftHelp.txt",
    "Preguntas Frecuentes (FAQ).txt",
    "AI Engineering Technical Challenge.pdf"
  ]
}
```