Installation Guide

Hybrid Semantic Search Installation Guide

Prerequisites

1. **Python 3.8+** (recommended for compatibility with all dependencies)
2. **Google API Key** (for Gemini embeddings)
   Sign up for Google's Generative AI API
   here and obtain your API key.
3. **System Libraries** (for FAISS):
   - **Linux**: `sudo apt-get install libopenblas-dev libomp-dev`
   - **macOS**: `brew install libomp openblas`
   - **Windows:** `pip install faiss-cpu`

Step 1: Open the Project

```
unzip hackathon
cd hackathon
```

Step 2: Install Dependencies

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

Install the dependencies:

```
pip install streamlit==1.32.0 PyPDF2==3.0.1 pandas==2.1.4 numpy==1.26.0 faiss-cpu==1.7.4 pdfplumber==0.10.4 nltk==3.8.1 scikit-learn==1.4.0 google-generativeai==0.3.2 google-api-core==2.15.0 python-dotenv==1.0.0
```

Install NLTK Data:

```
python -c "import nltk; nltk.download('punkt')"
python -c "import nltk; nltk.download('punkt_tab')"
```

Step 3: Configure API Key

1. Create a `.env` file in the project root:

   ```
   touch .env
   ```

2. Add your Google API key:

   ```
   GEMINI_API_KEY=your-api-key-here
   ```

3. Update `search_logic.py` to use the environment variable:
   Replace the hardcoded API key line:

   ```
   genai.configure(api_key=os.getenv("GEMINI_API_KEY"))  # Update this line
   ```

Step 4: Set Up Directories (Optional)

The app will auto-create these folders on first run, but you can create them manually:

```
mkdir -p data processed embeddings
```

Step 5: Run the Application

```
streamlit run app.py
```

Step 6: Usage Instructions

1. **Upload PDFs**: Drag and drop files into the sidebar.
2. **Search**: Enter a query (e.g., "AI ethics") and select a search mode (Hybrid recommended).

3. **Results**: View highlighted snippets with relevance scores.

Troubleshooting

1. **FAISS Installation Issues**:
   - Use `conda install -c conda-forge faiss-cpu` if pip fails.
   - Ensure system libraries (Step 1 prerequisites) are installed.
2. **API Errors**:
   - Ensure your `.env` file is correctly formatted.
   - Check usage limits for Google's Gemini API.
3. **PDF Extraction Failures**:
   - Image-based PDFs will show `<IMAGE PAGE>` placeholders.
   - Use OCR tools for scanned documents.

Project Structure

```
├── app.py                # Streamlit UI and logic
├── search_logic.py       # Document processing and search e
ngine
├── data/                 # Uploaded PDFs
├── processed/            # Processed text chunks (CSV)
├── embeddings/           # FAISS/TF-IDF indices
└── .env                  # API key configuration
```