

Documentation for the Flask Application

This documentation provides an overview of the Flask application that integrates with MongoDB for data storage, Hugging Face for natural language processing, and FAISS for vector storage. The application allows users to upload PDFs, extract text, and interact with a conversational AI model.

Table of Contents

1. Introduction
2. Dependencies
3. Configuration
4. Routes
 - Index Route
 - Update Knowledge Base Route
 - Delete Task Route
 - Update Task Route
5. Functions
 - PDF Text Extraction
 - Text Chunking
 - Vector Store Creation
 - Conversation Chain Creation
6. Running the Application

Introduction

The Flask application is designed to handle tasks related to text processing and conversational AI. It allows users to upload PDF documents, extract text, and interact with a conversational model powered by Hugging Face. The application also stores tasks and their responses in a MongoDB database.

Dependencies

The application relies on several Python libraries:

- **Flask**: A lightweight web framework for Python.
- **pymongo**: A Python driver for MongoDB.
- **dotenv**: A library to load environment variables from a `.env` file.
- **PyPDF2**: A library to read and extract text from PDF files.
- **langchain**: A framework for building applications with language models.
- **FAISS**: A library for efficient similarity search and clustering of dense vectors.
- **certifi**: A library to provide Mozilla's CA Bundle for SSL verification.

Configuration

The application uses environment variables for configuration, which are loaded from a `.env` file. The following environment variables are required:

- `HUGGINGFACEHUB_API_TOKEN`: API token for Hugging Face Hub.
- `MONGODB_URI`: URI for connecting to the MongoDB database.

Routes

Index Route

- **URL:** `/`
- **Methods:** `GET`, `POST`
- **Description:**
 - **GET:** Fetches the latest 5 tasks from the MongoDB database and renders them on the index page.
 - **POST:** Accepts a task content, processes it using the conversational AI model, and stores the task and response in the database.

Update Knowledge Base Route

- **URL:** `/admin/update_kb`
- **Methods:** `GET`, `POST`
- **Description:**
 - **GET:** Renders the upload page for updating the knowledge base.
 - **POST:** Accepts a PDF file, saves it to the `knowledge_base` folder, and logs the upload.

Delete Task Route

- **URL:** `/delete/<task_id>`
- **Methods:** `GET`
- **Description:** Deletes a task from the MongoDB database based on the provided `task_id`.

Update Task Route

- **URL:** `/update/<task_id>`
- **Methods:** `GET`, `POST`
- **Description:**
 - **GET:** Fetches the task to be updated and renders the update page.
 - **POST:** Updates the task content in the MongoDB database.

Functions

PDF Text Extraction

- **Function:** `get_pdf_text(pdf_docs)`

- **Description:** Extracts text from a list of PDF documents.
- **Parameters:**
 - `pdf_docs`: List of PDF file paths.
- **Returns:** Concatenated text from all PDF pages.

Text Chunking

- **Function:** `get_text_chunks(text)`
- **Description:** Splits the extracted text into smaller chunks.
- **Parameters:**
 - `text`: The text to be split.
- **Returns:** List of text chunks.

Vector Store Creation

- **Function:** `get_vectorstore(text_chunks)`
- **Description:** Creates a vector store from text chunks using Hugging Face embeddings.
- **Parameters:**
 - `text_chunks`: List of text chunks.
- **Returns:** FAISS vector store.

Conversation Chain Creation

- **Function:** `get_conversation_chain(vectorstore)`
- **Description:** Creates a conversational chain using the Hugging Face model and the vector store.
- **Parameters:**
 - `vectorstore`: FAISS vector store.
- **Returns:** `ConversationalRetrievalChain` object.

Running the Application

To run the application, ensure that all dependencies are installed and the `.env` file is properly configured. Then, execute the following command:

```
python app.py
```

The application will start on port 5001 by default. You can access it via `http://localhost:5001`.

Conclusion

This Flask application provides a robust platform for interacting with conversational AI models, managing tasks, and updating knowledge bases with PDF documents. The integration with MongoDB ensures that tasks and responses are persistently stored, while the use of Hugging Face and FAISS enables advanced text processing and retrieval capabilities.