DOC-X: Quick Insight Solutions

Overview

Quick Insight Solutions is an advanced AI-powered system designed to extract and retrieve information from unstructured DOCX documents efficiently. Utilizing the RAG (Retrieval-Augmented Generation) architecture, this tool allows users to upload DOCX files and query specific information, providing accurate and relevant answers. It combines document processing, natural language understanding, and information retrieval to transform raw text into actionable insights. Ideal for businesses, researchers, and professionals, Quick Insight Solutions enhances productivity by simplifying data extraction and ensuring quick access to critical information from extensive and complex documents.

Features

- Unstructured Data Processing: Efficiently handles and processes unstructured DOCX files.
- AI-Powered Insights: Leverages advanced AI models for natural language understanding and information retrieval.
- User-Friendly Interface: Simple and intuitive UI for uploading documents and querying information.
- Secure and Scalable: Ensures data security and scalability to handle large volumes of documents.

Architecture

The system is built using a combination of modern web technologies and AI frameworks:

- Frontend: HTML, CSS, JavaScript, and Flask for the web interface.
- Backend: Flask for handling requests, file uploads, and processing.
- AI Models: Hugging Face models for embeddings and question-answering.
- Vector Store: Chroma for storing and retrieving document chunks.

Installation and Setup

1. Clone the Repository: Clone the repository from the provided source.

2. Install Dependencies: Use pip to install the necessary dependencies listed in the requirements.txt file.

3. Set Up Environment Variables: Ensure the Hugging Face API token is set in the environment.

4. Run the Application: Start the Flask server to run the application locally.

Usage

1. Upload DOCX File: Use the provided interface to upload a DOCX file.

2. Enter Query: Input your query in the provided text box.

3. Process and Retrieve Insights: Submit the query to process the document and retrieve the relevant insights.

Code Structure

- app.py: The main application file containing routes and logic for handling requests.

- templates/: Directory containing HTML templates for the web interface.

- static/: Directory containing static files such as CSS and JavaScript.

- uploads/: Directory where uploaded DOCX files are stored temporarily.

Future Enhancements

- Enhanced Query Capabilities: Improve the natural language processing capabilities to handle more complex queries.

- Support for More File Types: Extend support to other file formats like PDF and TXT.

- Real-time Collaboration: Enable multiple users to collaborate and query documents in real-time.