DocuQuery: Intelligent Document Q&A

# Overview

DocuQuery is a powerful tool designed to facilitate intelligent question-answering from various document formats. Whether you have a PDF, TXT file, or even a URL link, DocuQuery allows you to upload the content and seamlessly ask questions related to the document's context.

# Features

## Document Upload

- Supported Formats: PDF, TXT files, or URL links.  
- Easy Access: Upload documents directly through the Streamlit user interface.

## Question-Answering

- Advanced NLP: Utilizes advanced natural language processing to provide accurate answers based on the document's content.

## OpenAI Integration

- Llama 3: Employs Llama 3, an open-source model with OpenAI embeddings for enhanced language understanding.

## Faiss Database

- Efficient Retrieval: Implements Faiss for efficient storage and retrieval of document embeddings.

## Streamlit UI

- User-Friendly Interface: Powered by Streamlit for smooth navigation and interaction.

## Tracking and Monitoring

- LangSmith Integration: Integration with LangSmith for comprehensive tracking and monitoring functionalities.

## LangChain Integration

- Enhanced Processing: Utilizes LangChain document loader, text splitter chains, etc., for improved document processing.

# Usage

1. Start the Streamlit App:  
 ```  
 streamlit run app.py  
 ```  
2. Access the Application:  
 - Open the provided URL in your browser.  
3. Upload Your Document:  
 - Upload a PDF, TXT file, or provide a URL link.  
4. Ask Questions:  
 - Ask questions related to the content of the document.  
5. Receive Answers:  
 - Receive intelligent answers generated by DocuQuery.

# Detailed Components

1. Document Upload  
DocuQuery allows users to upload various document formats (PDF, TXT, or URL) through a simple and intuitive interface. The documents are then processed and prepared for question-answering.  
  
2. Question-Answering  
Leveraging advanced natural language processing techniques, DocuQuery can understand and answer questions based on the uploaded document's content. The integration of Llama 3 and OpenAI embeddings ensures high accuracy and contextual understanding.  
  
3. OpenAI Integration  
By employing Llama 3 with OpenAI embeddings, DocuQuery enhances its language understanding capabilities, making it more effective in comprehending and responding to user queries.  
  
4. Faiss Database  
Faiss, a library for efficient similarity search and clustering of dense vectors, is utilized to store and retrieve document embeddings efficiently. This ensures fast and accurate retrieval of information.  
  
5. Streamlit UI  
The application features a user-friendly interface powered by Streamlit. This interface allows for easy document uploads, question submissions, and answer displays, providing a smooth user experience.  
  
6. Tracking and Monitoring  
DocuQuery integrates with LangSmith for robust tracking and monitoring functionalities. This helps in keeping track of document processing and query handling, ensuring reliability and efficiency.  
  
7. LangChain Integration  
LangChain is employed for enhanced document processing. It includes features like document loader and text splitter chains which facilitate better handling and processing of uploaded documents.

# Improvement Areas

1. Upgrade to Llama 3  
- Initially used Llama 2 but upgraded to Llama 3 for improved response accuracy and contextual understanding due to better training on more extensive datasets.  
  
2. Integration with OpenAI API  
- Consider utilizing OpenAI API key for additional functionalities and to leverage the latest advancements in natural language processing.  
  
3. Expand Llama Model Parameters  
- Increase usage of Llama models trained on more parameters for enhanced performance in understanding complex queries and documents.  
  
4. Implement Vector Database (Pinecone or DataStax)  
- Integrate specialized vector database for efficient similarity search and retrieval of document embeddings.  
- Enhance search capabilities, scalability, and accuracy of DocuQuery's query responses.  
  
5. LangChain Conversation Memory  
- Develop functionality to remember past interactions (conversation memory) for improved contextual understanding and personalized responses, transforming DocuQuery into a more interactive tool.

# Project Flow:

# 

### Explanation of the Flowchart

The flowchart depicts the process flow of DocuQuery, an intelligent document question-answering system. Here's a step-by-step explanation:

#### 1. User Actions

* **User Uploads Document (URL, PDF, TXT):**
  + The process begins when a user uploads a document in one of the supported formats: URL, PDF, or TXT.

#### 2. Document Processing

* The uploaded document undergoes initial processing to prepare it for embedding generation.

#### 3. Embedding Generation with OpenAI GPT-3

* **Embedding Generation:**
  + The processed document is then passed to OpenAI GPT-3 to generate embeddings. Embeddings are numerical representations of the document's content, capturing its semantic meaning.

#### 4. Embedding Storage in Faiss

* **Storage:**
  + The generated embeddings are stored in Faiss, a library for efficient similarity search and clustering of dense vectors. This allows for quick retrieval of relevant parts of the document.

#### 5. User Submits Query

* **Query Submission:**
  + The user submits a query related to the content of the uploaded document.

#### 6. Query Processing with NLP

* **Natural Language Processing:**
  + The system processes the user’s query using advanced NLP techniques to understand the context and intent.

#### 7. Retrieve Relevant Embeddings from Faiss

* **Retrieval:**
  + Based on the processed query, the system retrieves the most relevant embeddings from the Faiss database that correspond to parts of the document likely to contain the answer.

#### 8. Answer Generation with LLaMA 3

* **Answer Generation:**
  + Using the retrieved embeddings, LLaMA 3 generates a coherent and accurate answer to the user’s query.

#### 9. Display Answer to User

* **Display:**
  + The generated answer is displayed to the user through the interface.

#### 10. Tracking and Monitoring with LangSmith

* **Tracking and Monitoring:**
  + The entire process, including document processing, query handling, and answer generation, is tracked and monitored using LangSmith to ensure reliability and efficiency.

# Deployment Strategies

1. Amazon EC2 Deployment  
- Deploying DocuQuery on Amazon EC2 instances utilizing Streamlit could provide a robust infrastructure for seamless user interaction. The scalability of EC2 instances ensures that DocuQuery can handle increasing document volumes and user traffic without compromising performance.  
  
2. API Integration with LangServe  
- Developing an API using LangServe for DocuQuery could open up possibilities for integration with various applications and services. This could make DocuQuery more versatile and accessible, serving a broader range of user needs.  
  
3. Enhanced Llama Model Usage  
- Leveraging a well-configured EC2 instance could enable us to utilize more parameter-rich Llama models. This could lead to significant improvements in document understanding and question-answering capabilities, enhancing the overall effectiveness of DocuQuery.

# Conclusion

DocuQuery is a comprehensive tool designed to streamline the process of intelligent question-answering from various document formats. With its advanced features and integrations, it provides an efficient and user-friendly solution for accessing information from documents. For more details and to get started, visit the [DocuQuery GitHub repository](https://github.com/Vansh3503/DocuQuery-Intelligent-Document-Q-A).