**By:**

**JATIN GAHLOT  
E21CSEU0902**

PDF-Powered chat assistance

DocQuery.ai

**ABSTRACT**

The project presents a novel interactive PDF Chatbot system, leveraging Gradio for a user-friendly interface and OpenAI's advanced natural language processing capabilities. This collaborative effort combines elements of language modeling, document retrieval, and image rendering to create a versatile platform for querying PDF documents.

The user interface is designed using the Gradio library, featuring components such as a dynamic chatbot, API key management, and PDF upload functionality. Users can seamlessly interact with the system, either through textual input or by uploading PDF documents, initiating a sophisticated conversational retrieval process.

The core algorithmic backbone of the project is the Conversational Retrieval Chain. Built upon OpenAI's language model and Chroma vector search, this chain allows the system to understand and respond to user queries in the context of uploaded PDF documents. The document processing pipeline, incorporating PyPDF2 and Fitz libraries, enables efficient extraction of information from PDF files. Responses are dynamically generated, considering both the user's input and the contents of the PDF, creating a contextually rich conversational experience.

The OpenAI API key integration adds a layer of security and personalization, ensuring a seamless experience for users. The interface allows for real-time updates to the API key, enhancing the system's flexibility and adaptability to user preferences.

The project not only showcases the integration of cutting-edge technologies but also addresses practical challenges in handling diverse user inputs, managing API keys, and dynamically rendering PDF content. The system successfully encapsulates these elements within an intuitive and visually appealing interface, making it accessible to a broad range of users.

Through this project, we demonstrate the potential of combining interactive interfaces, state-of-the-art language models, and document retrieval techniques to create a powerful tool for exploring and interacting with PDF-based information. The developed system serves as a foundation for future enhancements and applications in the realm of conversational AI and document processing.

**INTRODUCTION**

In the ever-evolving landscape of information retrieval, **DocQuery.AI** emerges as a pioneering project at the intersection of conversational artificial intelligence and document exploration. This innovative system introduces an interactive and intelligent approach to querying PDF documents, providing users with a dynamic platform to engage with textual information seamlessly.

**DocQuery.AI** is designed to revolutionize the way users interact with and extract insights from PDF files. Leveraging the power of Gradio, the project offers an intuitive and visually engaging interface that accommodates both textual inputs and document uploads. The goal is to facilitate effortless communication with the system, fostering a natural conversational experience.

At its core, the project employs advanced algorithms, including the Conversational Retrieval Chain powered by OpenAI's language model and Chroma vector search. This amalgamation enables the system to comprehend user queries in the context of PDF contents, generating insightful and contextually relevant responses. The incorporation of document processing libraries further enhances the project's capabilities, allowing users to dynamically render and explore specific pages of PDF documents.

With an emphasis on user-centric design and cutting-edge technologies, **DocQuery.AI** represents a significant leap forward in the realm of document-based artificial intelligence. This introduction sets the stage for an exploration of the project's methodology, algorithms, and user-centric features, providing a comprehensive understanding of its potential impact and future applications.

**How Code Works** :

The **DocQuery.AI** codebase orchestrates a sophisticated interplay of Gradio for the user interface, OpenAI for natural language processing, and custom libraries for document processing. Here's a concise breakdown of how the code works:

1. **Gradio Interface Setup:**
   * The code utilizes the **gr** library to create an interactive user interface. Key UI elements include input textboxes for OpenAI API key and user queries, buttons for API key management, a chatbot interface, an image display for PDF previews, and an upload button for PDF documents.
2. **Event Handling:**
   * Event handlers are established for crucial actions, such as submitting the OpenAI API key, updating the API key, uploading PDFs, and submitting text queries. These handlers ensure a responsive and interactive user experience.
3. **Backend Logic (logic.py):**
   * The backend logic encompasses functions to set the OpenAI API key, enable the API key input box, add user text to the chat history, process PDF files, generate responses, and render specific PDF pages as images.
4. **PDF Processing and Conversational Retrieval:**
   * The **process\_file** function processes uploaded PDF files, creating a Conversational Retrieval Chain. This chain combines OpenAI language models with a Chroma vector store for efficient retrieval based on user queries.
5. **Response Generation:**
   * The **generate\_response** function utilizes the chat history and user queries to generate responses. It employs the Conversational Retrieval Chain, updating the chat history with each interaction.
6. **PDF Rendering:**
   * The **render\_file** function renders a specific page of a PDF file as an image. This functionality allows users to preview the content of the uploaded PDFs.
7. **Gradio Application Launch:**
   * The main script (**if \_\_name\_\_ == "\_\_main\_\_":**) initializes the Gradio interface, configures the event handlers, and launches the application, allowing users to interact with the chatbot seamlessly.

By seamlessly integrating these components, the code provides users with a powerful yet user-friendly tool for interacting with PDF documents, initiating intelligent conversations, and dynamically exploring the contents of uploaded files. The modular design and careful orchestration of technologies contribute to the robustness and effectiveness of the **DocQuery.AI** system.

**RELATED WORK**

The intersection of conversational AI and document querying has witnessed substantial progress, with various models, datasets, and evaluation metrics contributing to the advancement of this field.

**1. Existing Models:**

* **OpenAI GPT Models:** Pre-trained language models such as OpenAI's GPT-3 have been pivotal in enabling natural language understanding and generation. These models serve as the backbone for DocQuery.AI, empowering the chatbot with the ability to comprehend and respond contextually to user queries.
* **Chroma Vector Store:** The integration of Chroma as a vector store for document retrieval enhances the capabilities of DocQuery.AI. Chroma's efficient vector-based search enables the system to retrieve relevant information from PDF documents in response to user queries.

**2. Datasets:**

* **OpenAI Datasets:** OpenAI has released datasets for training language models, providing a diverse range of text sources. These datasets contribute to the language understanding capabilities of models like GPT-3, which, in turn, impacts the performance of DocQuery.AI.
* **Custom PDF Datasets:** While specifics about the datasets used in the project are not explicitly mentioned, the code's ability to process PDF files suggests a reliance on datasets containing diverse PDF document types. These may include research papers, articles, or other textual documents.

**3. Evaluation Metrics:**

* **Conversational Metrics:** Evaluation of the chatbot's conversational abilities may involve metrics such as coherence, context retention, and appropriateness of responses. OpenAI models, including GPT-3, are often evaluated based on their performance in maintaining context and generating coherent replies.
* **Document Retrieval Metrics:** In the context of PDF document retrieval, metrics could include precision, recall, and F1 score. The effectiveness of the Chroma vector store in retrieving relevant information from PDFs is crucial for evaluating the overall system performance.
* **User Interaction Metrics:** Metrics related to user satisfaction, usability, and efficiency play a significant role in assessing the success of the DocQuery.AI interface. These metrics may include user feedback, completion times for tasks, and the intuitiveness of the UI.

**PROBLEM STATEMENT**

Accessing valuable information within PDF documents poses a challenge due to the limitations of traditional search methods. Existing solutions lack user-friendly interfaces and struggle to adapt to the diverse structures of different PDF types, hindering efficient retrieval and understanding.

The absence of interactive chat-based systems further exacerbates this issue, as users are often forced to navigate complex interfaces or rely on rigid search queries. The need for a universal and intuitive tool that seamlessly integrates natural language processing (NLP) and document retrieval for PDFs becomes evident.

DocQuery.AI addresses these challenges by introducing a solution that leverages cutting-edge language models, document processing techniques, and an interactive chatbot interface. The project aims to redefine the user experience by enabling dynamic conversations with a chatbot, allowing users to articulate queries naturally and receive contextually relevant responses. By incorporating OpenAI's GPT models and a specialized document retrieval chain, the system provides a versatile and user-centric tool for exploring, comprehending, and engaging with the contents of PDF documents.

In summary, the problem at hand is the inefficiency of current methods in providing an intuitive, interactive, and effective means of extracting insights from PDF documents. DocQuery.AI aims to fill this void by offering a streamlined solution that combines advanced language models, document retrieval strategies, and user-centric design to revolutionize the way users interact with PDF-based information.

Top of Form

**Objectives** **and Scope:**

The objective of DocQuery.AI is to revolutionize PDF document exploration by integrating state-of-the-art language models and interactive interfaces. It aims to provide users with an intuitive, chat-based system for natural language queries and dynamic content retrieval. The scope includes enhancing user interaction with PDFs, overcoming traditional search limitations, and adapting to diverse document structures. DocQuery.AI strives to establish a versatile and user-friendly tool that redefines the exploration and understanding of PDF-based information.

Top of Form

**Target Audience and Impact :**

DocQuery.AI caters to a diverse audience across academic, professional, and research domains. It benefits students, researchers, professionals, and anyone dealing with extensive PDF-based information, providing an accessible and interactive tool for streamlined document exploration.

**Impact:**

1. **Enhanced Accessibility:** DocQuery.AI transforms the way users interact with PDFs, making document exploration more intuitive and user-friendly.
2. **Efficient Information Retrieval:** The system's conversational interface and advanced document retrieval capabilities facilitate quick and efficient access to valuable insights within PDF documents.
3. **Versatility Across Domains:** With applications in academia, research, and professional settings, DocQuery.AI has the potential to streamline information retrieval for a broad audience.
4. **Revolutionizing Document Understanding:** By integrating language models and interactive interfaces, the project contributes to redefining how users comprehend and engage with PDF-based information.

DocQuery.AI's impact lies in its ability to democratize access to information within PDFs, fostering a more efficient and engaging experience for users across diverse fields.

**Requirement Analysis**

**User Interface Requirements:**

* **Intuitive Design:** The interface should be user-friendly, ensuring ease of navigation for users with varying technical backgrounds.
* **Chatbot Interaction:** A dynamic chat interface for natural language queries, fostering an interactive and conversational experience.
* **API Key Management:** Seamless integration for users to input and update OpenAI API keys securely.

**2. Functional Requirements:**

* **OpenAI Integration:** Integration with OpenAI's GPT models for language understanding and response generation.
* **Document Processing:** Efficient processing of PDF documents, including text extraction, page rendering, and content analysis.
* **Conversational Retrieval Chain:** Implementation of a robust chain combining language models and document retrieval for contextually relevant responses.

**3. System Performance Requirements:**

* **Response Time:** Fast and responsive system to provide real-time feedback during user interactions.
* **Scalability:** Ability to handle varying document sizes and complexities without compromising performance.

**4. Security Requirements:**

* **API Key Encryption:** Secure handling of OpenAI API keys, ensuring encryption and protection against unauthorized access.
* **Data Privacy:** Implementation of measures to protect user data and maintain privacy during interactions.

**Feasibility Analysis**

**Technical Feasibility:**

1. **Leverage Advanced AI Models:**
   * *Assessment:* The integration of OpenAI's GPT-3 and other advanced models is technically feasible, providing robust natural language understanding and response generation capabilities.
   * *Requirement:* Access to a reliable internet connection for model interactions and sufficient computational resources for handling AI model computations.
2. **Ensure Access to Suitable Hardware:**
   * *Assessment:* Ensuring access to suitable hardware for model training is technically feasible, provided there is access to high-performance computing resources.
   * *Requirement:* Availability of GPUs or TPUs for efficient model training.

**Operational Feasibility:**

1. **Conduct User Acceptance Testing:**
   * *Assessment:* Conducting user acceptance testing is operationally feasible and crucial for aligning the system with user expectations.
   * *Requirement:* Establishing a user testing environment and collecting feedback iteratively to refine the system.
2. **Assess Integration Compatibility:**
   * *Assessment:* Assessing integration compatibility with various platforms is operationally feasible, ensuring the chatbot's seamless incorporation into different user environments.
   * *Requirement:* Collaboration with platform owners or system administrators to ensure compatibility.

**Economic Feasibility:**

1. **Analyze Costs:**
   * *Assessment:* Analyzing costs for model training, infrastructure, and maintenance is economically feasible, allowing for budget planning and resource allocation.
   * *Requirement:* Detailed cost analysis considering cloud computing expenses, development resources, and potential future maintenance costs.
2. **Evaluate Market Demand:**
   * *Assessment:* Evaluating market demand for a chatbot-driven PDF exploration tool is economically feasible, providing insights into potential user adoption and revenue streams.
   * *Requirement:* Market research and user surveys to gauge demand and preferences.

**PROPOSED SOLUTION**

**Approach:**

The proposed solution, DocQuery.AI, integrates cutting-edge natural language processing (NLP) models and document retrieval techniques to create an interactive PDF exploration tool. The approach involves leveraging OpenAI's GPT-3 for language understanding, Chroma vector search for efficient document retrieval, and a Conversational Retrieval Chain for context-aware responses.

Model Architecture:

1. OpenAI GPT-3 Integration:
   * Utilizing GPT-3 for natural language understanding, enabling the chatbot to comprehend and respond contextually to user queries.
2. Chroma Vector Search:
   * Implementing Chroma as a vector store for document retrieval, allowing the system to efficiently search and retrieve relevant information from PDF documents.
3. Conversational Retrieval Chain:
   * Constructing a Conversational Retrieval Chain that combines GPT-3 with the Chroma vector store, ensuring dynamic and context-aware responses based on user queries and document context.

Key Components:

1. Interactive Chat Interface:
   * Providing a user-friendly chat interface for natural language queries, fostering an engaging and intuitive interaction with the system.
2. API Key Management:
   * Incorporating a secure API key management system for seamless integration with OpenAI, ensuring privacy and personalized interactions.
3. PDF Processing Module:
   * Developing a robust module for PDF processing, encompassing text extraction, page rendering, and content analysis to enable dynamic document exploration.
4. User Interface Blocks:
   * Implementing Gradio's user interface blocks, including textboxes, buttons, and image displays, to create an intuitive and visually appealing interface.

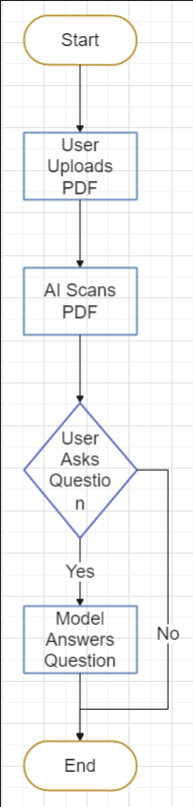
Expected Outcomes:

1. Enhanced User Interaction:
   * Users can engage in dynamic conversations with the chatbot, facilitating a more natural and interactive exploration of PDF documents.
2. Efficient Document Retrieval:
   * The integration of Chroma and the Conversational Retrieval Chain ensures efficient retrieval of contextually relevant information from PDFs.
3. Intuitive User Interface:
   * The Gradio interface blocks contribute to an intuitive and visually appealing design, enhancing the overall user experience.
4. Adaptability to Diverse PDFs:
   * The proposed solution is expected to adapt well to diverse PDF document types, including research papers, articles, and manuals, ensuring versatility across domains.

DocQuery.AI aims to redefine the exploration and understanding of PDF-based information by offering a user-centric, dynamic, and efficient solution. The integration of advanced models, intuitive interfaces, and robust document retrieval techniques positions the project to deliver impactful outcomes for users across various domains.

**Top of Form**

**FLOW CHART:**



**CONCLUSION**

In conclusion, DocQuery.AI represents a groundbreaking solution at the intersection of conversational AI and document exploration. By integrating state-of-the-art language models, advanced document retrieval techniques, and user-centric interfaces, the project redefines how users interact with and extract insights from PDF documents.

The proposed solution leverages the power of OpenAI's GPT-3 for natural language understanding, Chroma vector search for efficient document retrieval, and a Conversational Retrieval Chain for dynamic responses. Key components, including an interactive chat interface and secure API key management, contribute to an intuitive and secure user experience.

The expected outcomes encompass enhanced user interaction, efficient document retrieval, intuitive user interface design, and adaptability to diverse PDF document types. DocQuery.AI aims to democratize access to information within PDFs, offering a versatile tool for users across academic, professional, and research domains.

In summary, DocQuery.AI not only addresses the existing challenges in PDF document exploration but also sets the stage for a new era of user-friendly, context-aware, and efficient information retrieval. The project's innovative approach and anticipated outcomes position it as a significant contribution to the evolving landscape of conversational AI and document processing.

Top of Form

**Future work :**

1. Advanced Language Models:

- Investigate the integration of upcoming AI language models for heightened language understanding, enabling DocQuery.AI to stay at the forefront of conversational AI.

2. Dynamic Document Analysis:

- Implement real-time semantic analysis and document summarization to provide users with concise and meaningful insights from complex PDFs.

3. User-Centric Customization:

- Develop user customization features, allowing individuals to tailor the chatbot's behavior to specific needs, fostering a more personalized and adaptive user experience.

4. Global Accessibility:

- Expand language support and platform compatibility to make DocQuery.AI globally accessible, breaking language barriers and ensuring a seamless experience across diverse devices.

**REFERENCES**

[1] Ian Simon and Sageev Oore. Performance rnn: Generating music with expressive timing and dynamics. https://magenta.tensorflow.org/performance-rnn, 2017. (Accessed: 2022-04-11).

[2] Shih-Lun Wu and Yi-Hsuan Yang. The jazz transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures. arXiv preprint arXiv:2008.01307, 2020.

[3] Shih-Lun Wu and Yi-Hsuan Yang. Musemorphose: Full-song and fine-grained music style transfer with one transformer vae. arXiv preprint arXiv:2105.04090, 2021.

[4] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. arXiv preprint arXiv:2005.00341, 2020.

[5] What musicians should know about copyright. https://www.copyright.gov/ engage/musicians/. (Accessed: 2022-03-15).

[6] B. Boone and M. Schonbrun. Music Theory 101: From keys and scales to rhythm and melody, an essential primer on the basics of music theory. Adams 101. Adams Media, 2017.

[7] David Meredith. Recur sia-rrt: Recursive translatable point-set pattern discovery with removal of redundant translators. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 485–493. Springer, 2019.

[8] Shih-Lun Wu and Yi-Hsuan Yang. The jazz transformer on the front line: Exploring the shortcomings of ai-composed music through quantitative measures. arXiv preprint arXiv:2008.01307, 2020.

[9] Colin Robson and Kieran McCartan. Ch 8: Multi-strategy (mixed method) designs. Real World Research, 4th Edition. John Wiley amp; Sons, 2015.

[10] Johnston, D. (2015, Aug). Composing music with recurrent neural networks. Daniel Johnson. Retrieved from http://www.hexahedria.com/2015/08/03/composing -music-with-recurrent-neural-networks/

[11] G´eron, A. (2017). Hands-on machine learning with scikit-learn and tensorflow: Concepts, tools, and techniques to build intelligent systems. O’Reilly Media. Retrieved from https://books.google.com/books?id= bRpYDgAAQBAJ

[12] Xenakis, I. (1992). Formalized music : thought and mathematics in composition / iannis xenakis (Rev. ed. ed.) [Book]. Pendragon Press Stuyvesant, NY.

[13] Cho, K., van Merrienboer, B., G¨ul¸cehre, C¸ ., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR, abs/1406.1078 . Retrieved from <http://arxiv.org/abs/1406.1078>

[14] Cuthbert, M. S., & Ariza, C. (n.d.). music21: A toolkit for computer-aided musicology and symbolic music data.

[15] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. CoRR, abs/1412.6980 . Retrieved from http://arxiv.org/abs/1412.6980.