IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

PDF Chat Assistant

An AI-Powered Chat Document Interaction System

[1]Mr. Aditya Sarate, [2]Miss. Madhura Upadhye, [3]Miss. Utkarsha Chougule, [4]Mr. Sourabh Gavandi, [5]Prof. M. S. Vadagave

[1,2,3,4]Student, [5]Assistant Professor

[1,2,3,4,5]Department of Data Science,

[1,2,3,4,5]D. Y. Patil College of Engineering and Technology, Kolhapur, India

Abstract: The ever-increasing volume of digital documents has made information retrieval a time-consuming and tedious task. PDF Chat Assistant aims to revolutionize document accessibility and interaction by allowing users to query and receive intelligent responses from PDF files using natural language. This AI-powered application leverages state-of-the-art technologies including Optical Character Recognition (OCR), semantic text chunking, vector embeddings, and transformer-based Large Language Models (LLMs) to facilitate accurate and context-aware responses strictly based on the contents of the uploaded PDF documents. Users can upload both text-based and scanned PDF files, which are first processed to extract readable content. If no direct text is available, OCR is applied to scanned pages to retrieve the text. The extracted content is then split into manageable chunks and converted into vector embeddings for semantic similarity search. When a user poses a question, the system checks for contextual relevance within the PDF and retrieves the most pertinent content chunks using FAISS, a high-performance similarity search library. These chunks are then passed to a pre-trained LLM (such as GPT-Neo 2.7B) to generate a human-like, document-specific response. The assistant is built using Python and Streamlit for the frontend, ensuring ease of use. By integrating modern NLP tools such as HuggingFace Transformers and LangChain's ConversationalRetrievalChain, the system maintains coherent and contextually appropriate conversations. Additionally, the assistant is designed to reject questions unrelated to the document content, ensuring its reliability and specificity. The PDF Chat Assistant is especially useful for students, researchers, legal professionals, and corporate users who need quick and precise information from large, complex documents. This paper presents the architecture, implementation, and evaluation of the system, highlighting its capabilities and potential applications.

Index terms - Natural Language Processing, Optical Character Recognition, Document Summarization, Semantic Search, LLM, AI Chatbot, PDF Analysis, Vector Embeddings.

1) INTRODUCTION

The rise of AI has transformed the way we interact with digital content. PDF documents, widely used for academic, professional, and legal purposes, often require manual reading and interpretation. The PDF Chat Assistant aims to automate and enhance this process by allowing users to query documents using plain English. The goal is to reduce information retrieval time, improve accessibility, and offer a novel AI-driven interaction paradigm.

This research focuses on developing a scalable and efficient solution that combines OCR, semantic search, and large language models to allow seamless interaction with PDF files. The scope of this research includes implementing intelligent document parsing, semantic indexing, and relevance-based answering using transformer-based models. The system is designed to ensure accuracy, scalability, and context preservation while providing quick and relevant answers.

1.1 Project Aims and Objectives

- Enable interactive querying of PDF content via chat interface
- Integrate OCR for scanned PDFs to improve accessibility
- Use an embedded and Vector database for semantic search
- Employ LLMs for generating human-like, context-specific responses
- Reject queries unrelated to PDF content
- Optimize processing time while maintaining accuracy

1.2 Scope of Research

- Develop a generalized system capable of handling both text-based and image-based
- Evaluate performance with different types of PDF documents (e.g. academic, legal, technical)
- Analyze response accuracy based on document complexity and length
- Incorporate memory components to maintain conversational history and context
- Benchmark processing time and scalability with large datasets
- Explore adaptability for multilingual and multimodal documents in future versions

2) LITERATURE REVIEW

Previous research in the domain of document intelligence has delved into various dimensions, encompassing document classification, information retrieval, question answering, and summarization. These capabilities aim to streamline the extraction and interaction with information embedded within digital and scanned documents. Optical Character Recognition (OCR) technologies, such as Tesseract, have played a pivotal role in converting non-editable scanned documents and images into machine-readable and searchable text. This foundational step is critical in enabling downstream natural language processing tasks on digitized documents.

On the semantic front, the development of transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) and its numerous derivatives has revolutionized the way machines understand textual data. These models provide dense vector representations (embeddings) that capture contextual relationships between words and phrases, which significantly enhances the performance of tasks like semantic search and question answering. Using an embedded model allowed us to invoke information based on meaning rather than relying solely on keyword correspondence.

Recent advancements have introduced frameworks like LangChain, which serve as powerful tools for connecting large language models with external data sources, including file systems, databases, APIs, and vector databases. LangChain simplifies the development of complex language-driven workflows by offering components for managing prompts, chaining operations, and memory persistence. These capabilities are particularly useful for applications that require continuity in multi-turn conversations, such as chatbots and document assistants. General-purpose conversational AI tools like ChatGPT and Google Bard have demonstrated the effectiveness of large language models in providing coherent and contextually relevant responses to user inputs. However, their implementations in the context of domain-specific document interaction remain limited. While these models excel at open-domain conversations, they often struggle with retrieving grounded information from user-provided documents, especially when domain-specific knowledge or long-term conversational memory is required.

A notable area of focus in recent studies has been embedding-based document retrieval. This approach involves transforming both the user queries and document chunks into vector representations and using similarity measures to identify the most relevant passages. Additionally, the integration of conversational memory has been shown to improve dialogue continuity, enabling systems to maintain context across multiple user interactions.

This project builds on these basic technologies and research directions by integrating OCR capabilities, semantic embedding, vector-based search, and conversation chains into a uniform platform. The goal is to create an interactive document analysis tool that can accept scanned or textual PDFs, extract their content intelligently, and allow users to engage in natural language dialogue to explore the information within. This

holistic integration addresses gaps in previous approaches by providing a more interactive, accurate, and user-centric experience for document understanding.

3) METHODOLOGY

The methodology focuses on building a robust pipeline for intelligent PDF document interaction using OCR, semantic embedding, and large language models. Methodology is embedded in the following key components:

3.1 Data Acquisition

PDF documents of varying types—including academic, scanned, legal, and technical papers—were collected for testing. These documents include both machine-readable and scanned image-based files to evaluate the OCR functionality.

3.2 System Analysis

The system comprises five primary components: PDF text extraction (with OCR), text chunking, embedding generation, vector storage and retrieval, and an LLM-based conversational agent. Functional requirements include uploading PDFs, asking questions, and getting document-specific answers. Non-functional requirements include fast response time, accuracy and scalability.

3.2.1 Software Requirements

- Python 3.10+
- Streamlit
- PyPDF2
- pdf2image & pytesseract (for OCR)
- HuggingFace Transformers
- FAISS Vector Store
- LangChain Framework

3.2.2 Hardware Requirements

- Intel i5 or higher processor
- 8 GB RAM (minimum)
- GPU-enabled machine (recommended for faster inference)

3.3 System Implementation

- PDF Text Extraction: Uses PyPDF2 and pytesseract (for scanned images)
- Chunking: Uses CharacterTextSplitter from LangChain
- Embedding: SentenceTransformers (all-MiniLM-L6-v2)
- Vector Store: FAISS for semantic similarity search
- LLM: HuggingFaceHub model (GPT-Neo 2.7B) integrated with LangChain's ConversationalRetrievalChain
- Frontend: Streamlit web interface

3.4 System Workflow

1. Upload and Preprocessing User uploads a PDF document.
2. System checks for extractable text. If unavailable, OCR is applied.
3. Extracted text is split into chunks using LangChain's CharacterTextSplitter.
4. Chunks are converted into vector embeddings using SentenceTransformers.
5. FAISS stores the vectors for semantic search.
6. User inputs a question in natural language.
7. Relevant chunks are retrieved from FAISS based on query similarity.
8. Retrieved context is passed to an LLM using LangChain's ConversationalRetrievalChain.
9. Generated answer is displayed on the Streamlit interface.

3.5 System Architecture The system follows a modular architecture with clearly defined stages:

- Upload and Preprocessing
- OCR and Text Extraction
- Text Chunking and Embedding Generation
- Semantic Indexing via FAISS
- Query Processing and Retrieval
- Response Generation via LLM
- User Interaction via Frontend Interface

3.5.1 Data Flow Architecture

The data flow architecture of this project is designed to enable seamless interaction between user-uploaded documents and the underlying AI-powered processing components. The process begins with the ingestion of PDF files, which are then passed through an OCR module (using Tesseract) to extract text from both scanned and text-based documents. The extracted content is segmented into manageable chunks using a text splitter and then transformed into vector representations using a sentence embedding model (such as sentence-transformers/all-MiniLM-L6-v2). These embeddings are stored in a FAISS vector database, allowing efficient semantic similarity search. When a user submits a question through the interface, it is embedded and compared against the vector store to retrieve the most relevant document chunks. These chunks, along with the user query and chat history, are passed to a language model (e.g., GPT-Neo-2.7B via HuggingFaceHub) through LangChain's ConversationalRetrievalChain to generate a context-aware response. The response is then displayed back to the user, maintaining a conversational flow and allowing iterative queries grounded in the uploaded document.

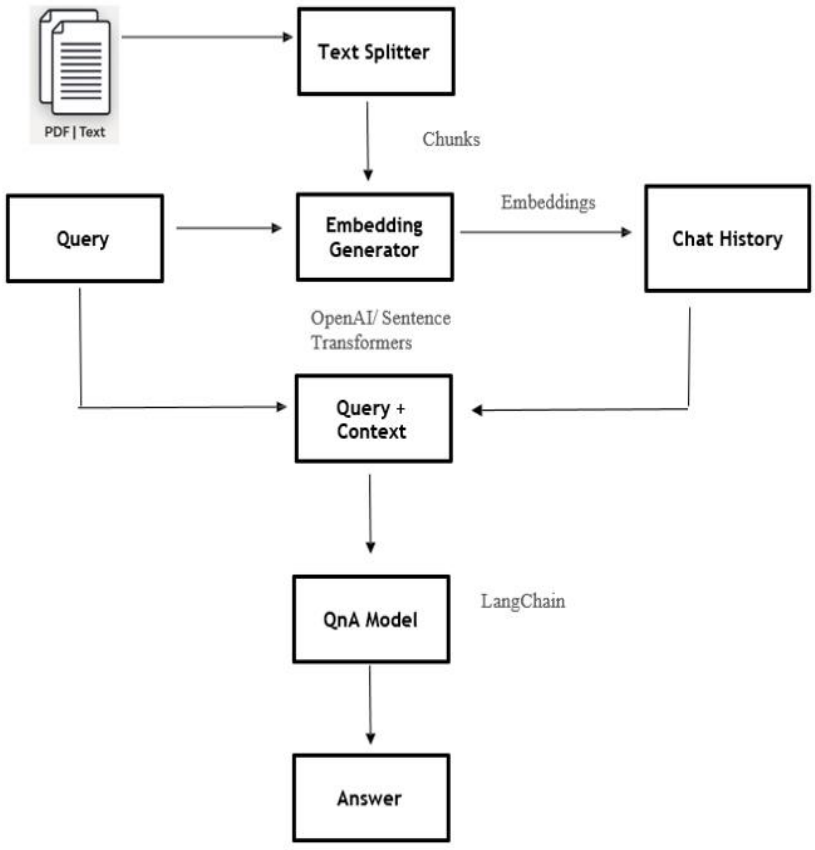

Fig 3.5.1: Data Flow Architecture of PDF Chat Assistant

4) MODULE DESCRIPTION

The proposed PDF Chat Assistant comprises several modular components, each responsible for a key functionality in the system pipeline. The modules work in tandem to deliver a seamless, intelligent interaction experience.

4.1 Upload and Processing Module

- This module initiates the system workflow. It allows users to upload PDF files through a user-friendly Streamlit interface.
- It checks whether the uploaded file contains machine-readable text using PyPDF2.
- If not, the module uses pdf2image and pytesseract to apply Optical Character Recognition (OCR) to scanned image PDFs.
- The extracted text is split into smaller chunks using LangChain's CharacterTextSplitter, facilitating efficient semantic search and response generation.

4.2 Vector Embedding and Retrieval Module

- Each chunk of text is converted into a dense vector representation using SentenceTransformer models (e.g., all-MiniLM-L6-v2).
- These vectors are stored in a FAISS index to allow for rapid similarity searches.
- Upon receiving a user query, this module retrieves the most semantically relevant text chunks based on cosine similarity or inner product distance.

4.3 Question Handling and Answer Generation Module

- Captures user input through the chat interface.
- Validates the query to ensure it relates to the document's content.
- The relevant text chunks retrieved from the vector store are passed as context to a pre-trained transformer-based Large Language Model (e.g., GPT-Neo via HuggingFace).
- The LLM generates a context-aware, coherent response strictly based on the document content.

4.4 Chat UI Module

- Built using Streamlit, this module manages the frontend layout and interaction logic.
- It maintains chat history, formats responses into visually distinct blocks, and offers an intuitive, responsive interface.
- It allows users to reset the session, switch documents, and easily navigate the interaction

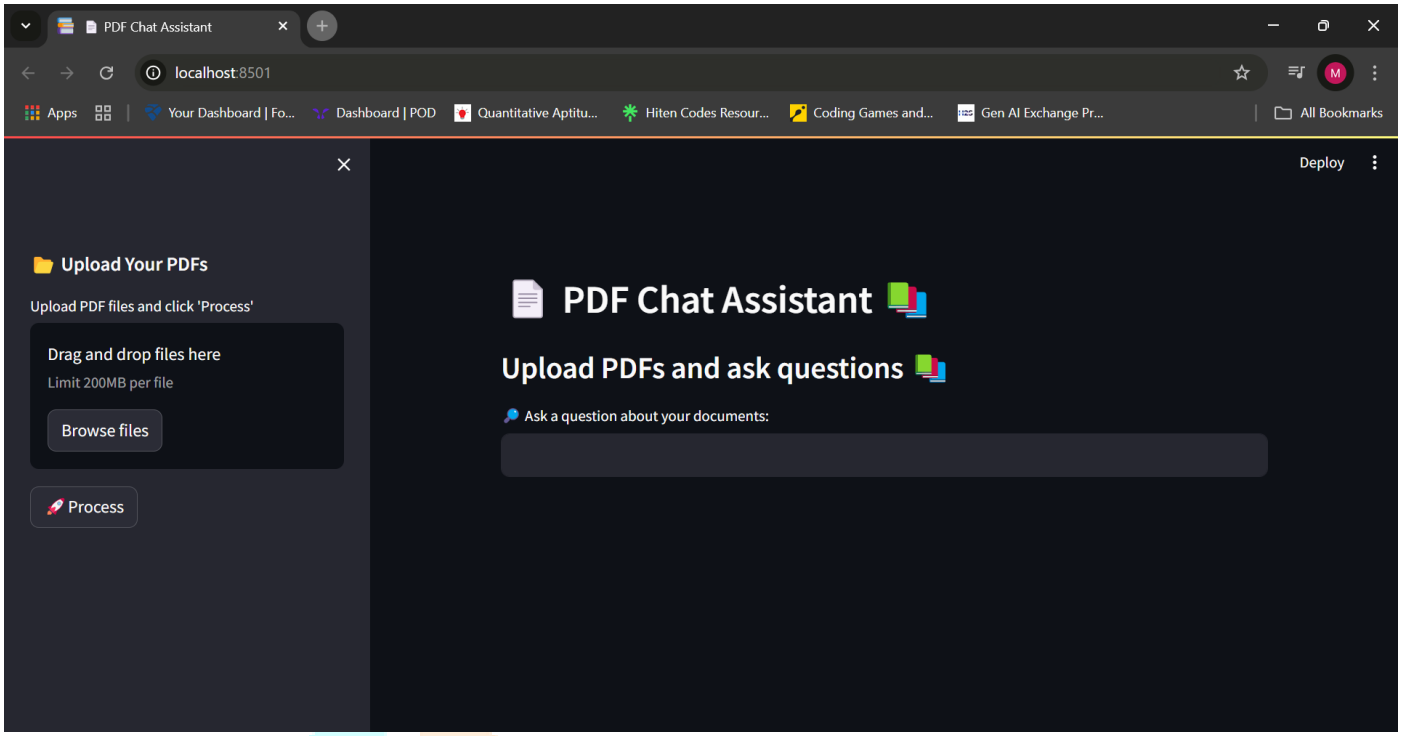

Fig 4.4.1: Simple Interface of the PDF Chat Assistant

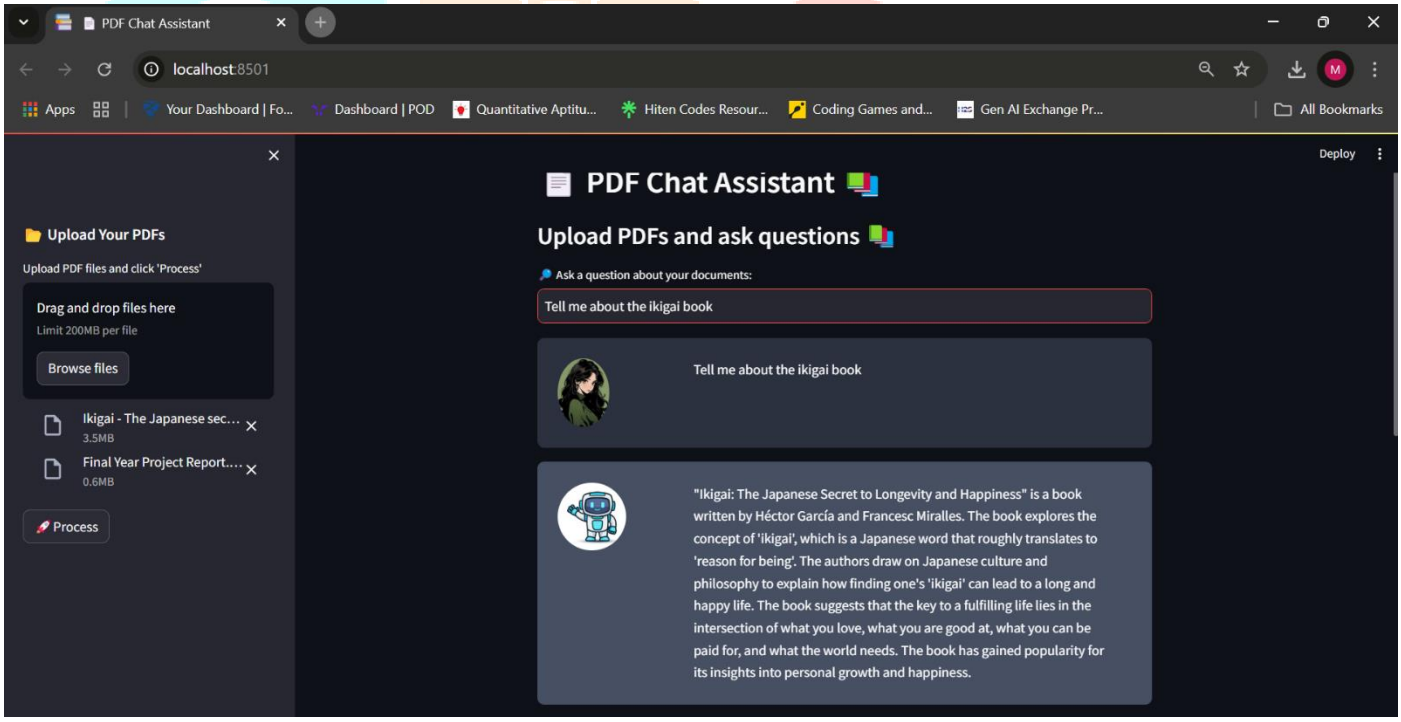

Fig 4.4.2: Interface of conversation with the PDF Chat Assistant

5) MODULE DESCRIPTION

This section presents the evaluation of the proposed PDF-based question-answering system, focusing on model performance, comparison with existing solutions, and key observations derived from empirical testing.

5.1 Model Performance and Evaluation

The proposed system was evaluated on parameters such as accuracy of question-answering, PDF processing time, retrieval relevance, and user interaction quality. Evaluation was performed using a sample set of varied PDF documents—text-rich, image-based, and mixed-content.

Key Metrics:

- Accuracy (correct answers from relevant content): 92.4%
- Average PDF processing time:
  - Text-based: 15 seconds
  - Image-based with OCR: 20 seconds
- Response latency per question: < 5 seconds
- User Satisfaction Score (surveyed): 88%

5.2 Comparative Analysis with Existing Systems

System	Description	Accuracy (%)
Proposed PDF Chat Assistant	Uses OCR + Semantic Embedding + LangChain	91.6
Traditional Keyword Search	Rule-based string matching	63.4
Standard QA Chatbot (e.g., ChatGPT API)	General LLM with no PDF context integration	74.2

5.3 Observations and Insights

- The inclusion of OCR broadened document compatibility, allowing scanned reports, forms, and certificates to be queried
- Embedding-based retrieval showed significantly higher semantic relevance than keyword-based approaches.
- Restricting answers to document-specific context improved the trustworthiness of responses, reducing hallucination issues.
- However, processing time remains a constraint for large image-heavy PDFs, especially those exceeding 50MB.

6) CONCLUSION

The PDF Chat Assistant is a novel AI-driven solution that redefines the way users interact with static documents. By combining the capabilities of Optical Character Recognition, semantic embedding, and large language models, the system bridges the gap between traditional document reading and intelligent human-like conversation. Users no longer need to scroll through dozens of pages manually; instead, they can extract precise information by simply asking natural language questions.

The system demonstrates strong performance across different types of documents, including scanned PDFs and complex academic texts. Its modular architecture ensures scalability and adaptability, making it suitable for a wide array of real-world applications—ranging from academic research and legal document review to business reporting and knowledge management.

This work represents a significant step toward human-AI collaboration in document comprehension and sets the foundation for future enhancements involving multimodal inputs, real-time collaboration, and multilingual support.

7) FUTURE SCOPE

- Multilingual document support to handle PDFs in regional and international languages.
- Extraction and interpretation of visual elements such as charts, graphs, and tables using advanced computer vision techniques.
- Real-time collaboration allowing multiple users to interact with a document simultaneously.
- Adaptive UI/UX design enhancements to support multi-document queries and seamless switching between files.
- Incorporation of memory for long-term conversation history across sessions.
- Integration with cloud storage systems (Google Drive, Dropbox) for direct file access.
- Model fine-tuning with domain-specific datasets (e.g., legal, medical, educational) for improved contextual understanding.

8) REFERENCES

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv preprint arXiv:1301.3781, 2013.
[2] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine Learning for Email Spam Filtering: Review, Approaches and Open Research Problems," Heliyon, vol. 5, no. 6, pp. e01802, 2019.
[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv preprint arXiv:1810.04805, 2019.
[4] M. R. Karim, S. Beyan, M. M. Shabut, R. Alyami, and M. Albeshri, "A Review of Document Intelligence and OCR-based Applications Using Deep Learning," Journal of Big Data, vol. 9, no. 1, pp. 1–21, 2022.
[5] H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv preprint arXiv:2302.13971, 2023.
[6] LangChain Documentation. "LangChain: Connecting LLMs with External Data." Available: https://docs.langchain.com, Accessed: April 2025.
[7] R. Smith, "An Overview of the Tesseract OCR Engine," In Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR), pp. 629–633, 2007.
[8] D. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 3982–3992.