A Retrieval-Augmented Conversational Framework for Reliable Question Answering on PDF Documents

Rahul J. Teradal
Design Verification Trainee
SION semiconductors Pvt Ltd
Bengaluru, India

Nikita L. Soude
Design Verification Trainee
SION semiconductors Pvt Ltd
Bengaluru, India

Megharaj Hiremath
Master of Technology (VLSI)
KLE Technological University
Hubli, India

Misbah Mansoor Nabiwale
Department of ECE
Jain College of Engineering
Belagavi, India

Shivam
Department of CSE (AIML)
East West Institue of Technology
Bengalore, India

Padmashree J. Teradal
Dept of ECE
Hirasugar Institute of Technology
Nidasoshi, India

***Abstract*—PDFs are widely used for sharing academic, techni- cal, and professional content, but extracting relevant information from lengthy documents remains difficult. Traditional keyword searches miss the semantic depth of queries, resulting in irrel- evant information. This paper presents DocMind, a Retrieval- Augmented Generation (RAG) framework for conversational question answering on PDFs. The system integrates semantic document parsing and chunking, embedding generation, and vector database retrieval (FAISS/Chroma) for efficient semantic searching.For each user's query, we embed it within the same vector space as the user's query. The most relevant chunks of text are identified, and the information is supplied to a Large Language Model (LLM) to generate a coherent answer. Unlike traditional RAG systems which produce static answers, DocMind is capable of multi-turn conversational memory and provides fallback clarity by informing the user when an answer is provided without supporting PDF data. In our experimental evaluation, we showed that DocMind significantly increased the baseline models using BM25 and TF-IDF in recall, F1 score, and ROUGE-L metrics due to improvement in contextual accuracy and hallucination mitigation. By merging semantic retrieval with generative reasoning, DocMind offers dependable, interactive, and domain-agnostic QA on PDF documents, greatly improving information accessibility and reliability.**

***Index Terms*—Retrieval Augmented Generation, Vector Database, FAISS, Chroma, Conversational QA, PDF Processing, Large Language Model**

1. Introduction

PDFs are widely used for the dissemination of scholarly papers, reports, legal agreements, and business data. Never- theless, despite the availability of traditional keyword search functions like Ctrl+F, retrieval of proper information is still not an easy affair since the semantic sense and contextual appropriateness involved. For instance, keyword searching can get the correct word but not the proper sense behind it. This problem is further complicated when users are required to browse through long documents with hundreds of pages, which is both laborious and time-consuming.

With fast developments in Natural Language Processing (NLP) and Large Language Models (LLMs), conversational systems are being developed in the direction of intelligent question-answering functionality. NLP allows computers to read and process human language, whereas LLMs such as GPT or Gemini can produce responses similar to those written by humans. But LLMs are prone to hallucinations - a situation in which the model generates confident but wrong facts not indicated by the source text [1][2][3]. This usually occursbecause LLMs do not depend on ratified information but on patterns of data.

To overcome this limitation, Retrieval-Augmented Genera- tion (RAG) has risen as a hybrid model that mingles infor- mation retrieval with generative reasoning [4]. Simply put, RAG initially looks for pertinent facts from reliable documents (retrieval) and subsequently employs an LLM to create a natural, well-structured response (generation). This way, the responses become more precise and based on actual data.

In this work, we introduce DocMind, a conversational PDF Question Answering (QA) model based on the RAG frame- work [5]. The model reads uploaded PDFs, splits the text into semantically meaningful pieces, creates semantic embeddings (numerical forms of meaning), and indexes them in vector databases [6] like FAISS or Chroma to perform similarity- based retrieval. These embeddings enable the model to retrieve semantically equivalent information even if the same words are not used.When a query is submitted by a user, DocMind gets the most contextually relevant chunks and presents them as context to a Large Language Model (LLM), which produces coherent and factually backed responses.Responses are based on the content in the provided PDF instead of information found online. DocMind utilizes conversational memory. It remembers the prior questions in the active conversational thread to enable seamless multi-turn dialogues. With this functionality, PDFs that are static in nature become responsive, intelligent knowledge bases that can be interacted with conversation- ally.Improving access, productivity, and decision-making in fields that rely on precise, contextual information, such as research and education, law and medicine, will have a notable impact in those fields.

2. Literature Survey

The capability of reading documents and carrying out conversation question answering (QA) with PDFs is

becoming a well-established, area of research. This proposed system relies on some of the most relevant research studies mentioned below:

Lala et al. (2023/24): The researchers proposed using retrieval- augmented the multi-agent frameworks for scientific research questions, when they studied those systems for PaperQA. Pa- perQA combines GPT-4/3.5 and Claude-2 with LangChain and Google Scholar APIs, and produces GPT-4/3.5-level outputswith low hallucination and high accuracy to create evidence- based responses. In PaperQA's architecture, there are 4 distinct but interrelated components involving reformulating the query, retrieving relevant papers and summarizing the results. While PaperQA produces superior responses than the baseline LLMs on scientific topics, it is constrained to access to scholarly papers for academic or research purposes on account of paywalls and APIs being costly to use. NeuSymRAG(2025): In the case of NeuSym RAG, Caoet al. [8] describes the introduction of the hybrid neural and symbolic RAG system, which incorporates PDF parsing, mul- timodal embeddings, and a SQL structured database. Qwen2.5- VL and LLaMA3.3 are used in conjunction with Milvus that works with vector indexing. This system shows enhancement of accuracy between 17 and 29 percent compared to traditional RAG, although the RAG system handles multimodal PDFs. However, this design is resource intensive and is fundamen- tally restricted to AI research papers which limits broader use cases. QA-RAG/A-RAG(2024): In the case of A-RAG, Royetal. [9] describes use of query decomposition and the refor- matting of QA pair which in turn helps in the response quality improvements. LLaMA3, Mixtral and Gemma LLMs are paired with the validation modules which are aimed to minimize hallucinations. Although the approach does improve the BLEU/ROUGE scores, the approach is resource intensive and relative to traditional RAG systems, there is a significant overhead. Next-GenQA (2024): In the case of Najwa et al. [10], a complete PDF QA pipeline is created for legal and tender documents. The system architecture includes PyPDFium2 for document parsing, ChromaDB for indexing, and OpenAI em- beddings along with ChatOpenAI for answer generation. The system accuracy is boosted with recursive splitting but there is a huge limitation to PDFs and no multilingual capabilities. RAG Experience Report from PDFs (2024): Khan et al. [11] provide a reproducible workflow for constructing RAG pipelines from PDFs in Python using PyMuPDF, pdfplumber, FAISS/Chroma methods along with either GPT or LLaMA models. The report provides best practices for RAG and situations it is more reliable than static LLM; however, the current implementation has limitation of OCR errors, and it cannot process multi-format documents (non-PDF). Conver- sational Text Extraction with LLMs (2024): Roy et al. [12] devised a pipeline which combined LangChain, FAISS, and Google GenAI to allow for a conversational style of text extraction from PDFs. By using PyPDF2 and Streamlit the system provides rapid reponse (1 second) while processing over 500 queries with accuracy. In this case, a limitation of the conversational text extraction is that it only supports PDFs with text (text-permitting) and cannot process multimodal documents. Chat with PDF using LangChain and AstraDB (2024): Pran- naveshwara & Kalaiselvan [13] presented the workflow com- bining LangChain with AstraDB for question-answering from PDF documents. Text was extracted with either PyMuPDFor PDFMiner and both models extracted prior to generating embeddings in AstraDB for scale. While potentially useful to enable a conversational mode within the query, there were in multiple areas of the workflow.Donut (ECCV2022): Kim et al. [14] presented Donut, a transformer for document understanding without using OCR. Unlike methods that extract text, Donut operates on document images and consists of a pair of Vision Transformer encoder and autoregressive text decoder that produces structured out- puts in JSON. This enables the method to bypass pitfalls associated with OCR techniques and process documents in different languages (i.e.mailing formats); however, the com- putational effectiveness of this methodology is far less effi- cient than OCR. The speed and accuracy for dense tables is debatable and entirely fails with handwritten documents.RAG-QAArena (2024): Han et al. [15] established the RAG QAArena benchmark that defines assessment of domain robustness of long-form RAG systems. the benchmark involves 26,000 Q&A pairs in 7 different domains for assessing other models including GPT-4, LLaMA, and Falcon. This work is an important start, but only scrapes the surface of challenges with RAG domain transfer, and provides no advancement of the system beyond providing benchmarks.Adaptive-RAG (2024): Jeonget et al. [16] proposed Adap- tive RAG, a system that dynamically selects retrieval ap- proaches based on computing complexity of the query. Queries that can be efficiently resolved by heuristics were sent to BM25 retrievers and hybrid or multi-hop retrieval methods were applied to queries that were more complex. The ac- curacy of the results were not sacrificed significantly, with computational costs of the system reduced by approximately 30%. However, reliability of the results can still suffer due to erroneous rates from classifiers in relation to more com- plex queries. To operate most effectively, Adaptive RAG will require an extensive amount of labeled training data.ChatDOC (2024): Lin[17] describes ChatDOC, a whitepa- per processing system that combines OCR, layout recognition, and structure-sensitive chunking. By preserving the layout of tables, sections, and multi-column layouts, ChatDOC improves accuracy of question-answering by 47% compared to basic parsing methods. However, the complex process only works in selected domains (finance and academia) and is unable to adapt to other industrial contexts.As presented in [Table 1], the comparative analysis of existing PDF QA systems reveals strengths, weaknesses, and potential future improvements.DocMind differs as it integrates a complete set of function- alities into a domain-agnostic, user-trust-centric RAG frame- work. Instead of being a specialized system like PaperQA (academic only), Next-Gen QA (legal/tender), or ChatDOC (finance/academia), DocMind is designed to handle general PDF documents (with flexible PDF parsing grip capabilities) .

TABLE 1
COMPARATIVE ANALYSIS OF EXISTING PDF QA SYSTEMS

Paper / System	LLMs / Transformers	Embeddings	Limitations	Possible Suggestions
PaperQA [7]	GPT-4, GPT-3.5, Claude-2	OpenAI embeddings	Academic-only, costly	Use open-source LLMs; expand to other domains
NeuSym RAG[8]	Qwen2.5-VL, LLaMA3.3	Multimodal + SQL	Compute-heavy, domain-limited	Optimize models; add general datasets
A-RAG[9]	LLaMA3, Mixtral, Gemma	QA-pair reformatted	High overhead	Automate QA-pair creation
Next-Gen QA[10]	ChatOpenAI	OpenAI embeddings	Only PDFs, no multilingual	Add multilingual and image support
Exp. Report[11]	GPT, LLaMA	FAISS / Chroma	OCR errors, PDF-only	Add OCR correction; support other formats
Conv. Text Extraction[12]	Google GenAI	FAISS	Text-only PDFs	Add OCR/image handling
Chat with PDF[13]	OpenAI / BERT	AstraDB	Weak summarization	Improve summarization
Donut [14]	ViT encoder + decoder	Visual patches	Compute-heavy	Use lightweight ViT
RAG-QA Arena[15]	GPT-4, LLaMA, Falcon	Dense + BM25	Benchmark only	Make deployable; real-user testing
Adaptive-RAG[16]	Hybrid	BM25 + dense	Needs labeled data	Use self-supervised training
ChatDOC[17]	GPT-4	OCR + layout-aware	Domain-limited, heavy	Modularize pipeline; smaller models

3. METHODOLOGY

The proposed system, DocMind, is a domain-agnostic, retrieval-augmented question-answering platform for PDF documents. Fig. 3.1 shows the overall system architecture, dividing it into a Client Side and an AI Engine. The client side handles user interactions and prepares the document by parsing and chunking it. The AI engine uses an embedding generator and retriever to find relevant information, then passes it to an LLM to generate a final, coherent response for the user. Eventually, the collected data is transferred to a Large Language Model (LLM) that creates a well-structured, context-sensitive, and correct reply. Eventually, the collected data is transferred to a Large Language Model (LLM) that creates a well-structured, context-sensitive, and correct reply. Integration of retrieval.

3.1 PDF Upload & Parsing

When the users desire to use a PDF file, they first upload the file into the system. The file is parsed through established libraries (such as, PyMuPDF, pdfminer) to read the raw text of the document. This is potentially the most significant step, as PDF files have architecture which differs from one document to another, hence making it more difficult to extract the text. The PDF parsing is predominantly based on the raw data cleanliness and consistency. Second, the PDF parsing module can also accommodate hybrid extraction techniques. That is for PDFs that are, essentially text-only, lightweight parsers are used, while for image-rich or scanned PDFs, Optical Character Recognition (OCR), with either Tesseract or EasyOCR, is utilized to extract the text found within images as text files. For maximum consistency and reliability, the financial statements, tables, or datasets are extracted in a structured form using pdfplumber or Camelot that are libraries natively coupled with the module. The output of this parsing step is the standardization of text data devoid of unnecessary noise such as headers or footers to produce semantic consistency for the downstream process.

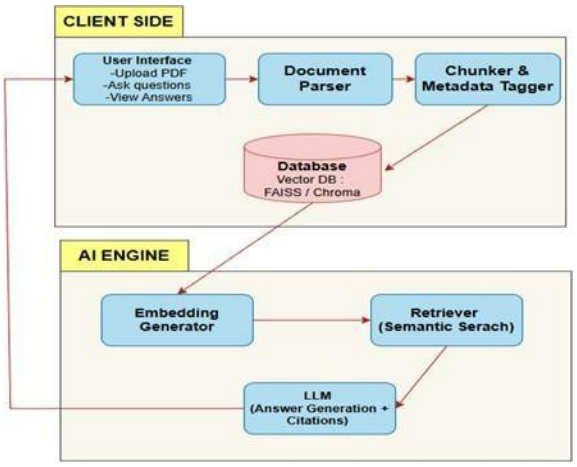

Fig. 3.1. System Architecture

3.2 Text Chunking

Documents typically exceed the point of straightforward processing given their length. In order to take a document-length text document and enhance the chances of success- fully retrieving relevant spans of text from it, the extracted

text is divided into smaller sections, or ”chunks,” which are semantically coherent and in a range of what is considered 200—500 tokens, ideally. Where chunks are split, an overlap of approximately 20% is utilized to avoid losing context across chunks. This process, enhances the quality of retrieval because, when querying, the model will likely retrieve more relevant spans of text fragments based on the original document text (in this context, the original document is the relevant document).

3.3 Embedding Generation

Each text chunk is transformed into vector embeddings, which represent the text's semantic meaning. These embeddings are dense, numerical representations of the text, and are used by the system to assess the similarity between queries and portions of a document. Some commonly used models include sentence-transformers (e.g., all-MiniLM-L6- v2, all-mpnet-base-v2) or proprietary APIs (e.g., OpenAI ada embeddings). The choice of embedding model has a large impact on performance, as smaller models (MiniLM, 384 dimensions) are faster and have less resource constraints, which is appropriate for lightweight deployments. Larger models (mpnet, 768 dimensions) encode deeper semantics, but require a more powerful GPU.

This affords DocMind the ability to be configurable, based on the application use case - for use case applications that focus on speed (e.g., chatbots, quick lookups), selecting a smaller and faster model is appropriate, while research or enterprise deployments may wish to choose larger embed- dings, which deliver a more accurate result, but require more computational power to run. Figure 3.2 shows the progression of how text is eventually transformed into numbers. Each text chunk is sent into an Embedding Model, which generates the embedding, or numerical representation, to develop a vector. These vectors are then organized and stored into a database for rapid and efficient searching and retrieval.

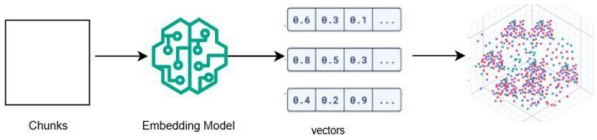

Fig. 3.2. Embedding Generation

3.4 Vector Database Storage & Retrieval

The generated embeddings are stored in a vector database (e.g., FAISS, Chroma, or Weaviate) that supports fast similarity search. During query time, the user's input is also transformed into an embedding and compared against the stored document vectors using nearest-neighbor search. The system retrieves the top-k most relevant chunks based on similarity scores.

The indexing strategy plays a critical role in system responsiveness. DocMind leverages HNSW (Hierarchical Navigable Small World) graphs for sub-linear search efficiency, enabling millisecond-level retrieval even with millions of vectors. Cosine similarity is the chosen metric as it measures semantic closeness effectively in high-dimensional spaces. By default, the system retrieves the top 5 results (k = 5), though this parameter is configurable depending on user needs - higher k improves recall but increases token usage when feeding context into the LLM. This ensures that only the most contextually aligned parts of the document are passed forward. Fig. 3.3 explains the retrieval process. A user's query is converted into an embedding, which is then used to search a Vector Database. The system finds and retrieves the most relevant document chunks by comparing their vectors to the query's vector.

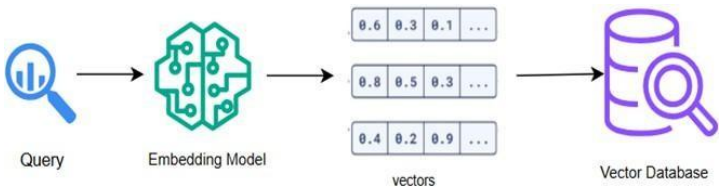

Fig. 3.3. Vector Database Retrieval

3.5 Query Processing & RAG Pipeline

When a user submits a question, it is embedded in the same vector space and used to query the database. The retrieved chunks are then combined with the user's query and sent to a large language model (LLM) for answer generation. The LLM (e.g., Gemini, ChatGPT, LLaMA, or Claude) uses the retrieved content as grounding context to reduce hallucinations and provide factually consistent answers. To improve precision, techniques for prompt engineering are utilized. The query alongside the retrieved segments is organized within structured templates that include clear instructions such as:

- "Answer strictly based on the provided content."
- "If context is insufficient, state it clearly instead of guessing."

This version of prompting makes it more likely that the answers will be factually accurate and less likely to generate another incorrect response. In this aspect, the entire pipeline, also utilizes contextual concatenation, where multiple top-k segments are all combined together into one context block so that we take an extended pass at finding relevant information. In doing this, we ensure that the answers are still primarily based on the document that was uploaded but that the fluency and coherence of the language model are still contributing. Figure 3.4 shows the complete Retrieval-Augmented Generation (RAG) Pipeline, where the user's input or question leads to get all the relevant document segments . Then the question and retrieved segments are entered into a language model that utilizes the sourced content to produce a response that is accurate and relevant.

3.6 Conversational Memory

To support ongoing conversations in the system, conversational memory is used. This helps follow-up questions get updated about previous conversations. If the question is not addressed in the uploaded PDF, the system uses a fallback

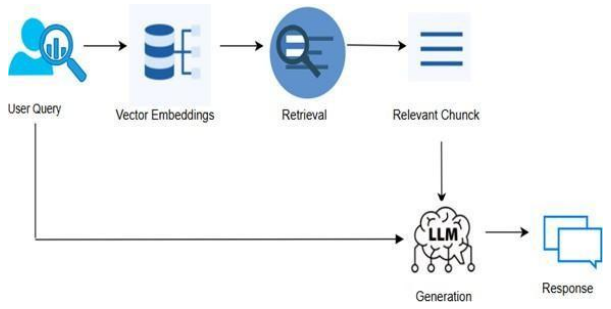

Fig. 3.4. RAG Pipeline

method to activate the LLM. The LLM will simply generate an answer based on its general knowledge, but the LLM adds a disclaimer saying "This answer is not from the uploaded PDF." Doing this transparently builds trust and makes it easier to avoid misleading the user into thinking an unsupported statement is from one of the PDFs.

3.7 Implementation Details and Toolkit

DocMind relies heavily on important libraries for document parsing: fitz (PyMuPDF) and PyPDF2. The text is taken and chunked and vectorized with the sentence-transformers (all MiniLM-L6-v2) and stored inchromadb. The main RAG engine retrieves the top5 results (k=5) for context and uses the Gemini2.0FlashLLMfor generation. The user interface is provided via the gradio framework.

4. Results and Discussion

To evaluate DocMind, a benchmark dataset made up of 40 multi-page PDFs, each an average of about 20 pages in length, was used. This dataset was created in-house, containing complex domain-specific PDFs and used for thorough evaluation, containing an average of 25 unique questions per document. Subsequently, the system's effectiveness was measured by assessing several quantitative metrics[Table 2]. These criteria have been selected to specifically evaluate the quality of RAG output metrics. Faithfulness - Evaluates factual constancy; does the generated answer have data from the retrieved documents context. Relevance - Evaluates relevance of retrieved context/full answer in response to the user query. Citation Correctness - Evaluates the accuracy of citation and verifies that any type of cited reference can be traced back to the PDF if presented.

The effectiveness of the system was evaluated using various quantitative criteria. Effectiveness of the system was measured using several quantitative metrics or criteria, such as recall, semantic similarity, faithfulness, and ranking-based metrics.

The findings for DocMind suggest that combining Retrieval-Augmented Generation (RAG) with semantic embeddings significantly improves contextual accuracy and reduces hallucinations in question answering about PDFs. Relative to standard retrieval methods (BM25 and TF-IDF), the proposed model has improved Average F1 by approximately 6×, ROUGE-L by approximately 7×, and BERT F1 by approximately 2×, indicating it has greater responsiveness in producing answers that are contextually true or semantically valid. Average Faithfulness (0.647) and Average Relevance (0.745) indicate that a DocMind-generated answer is reasonably likely to be grounded and answered according to the citation from the source text.

A key point of failure arises when PDFs are complex or filled with images, leading to a lack of retrieval due to imperfect parsing, and increased response time. Additionally, retrieval accuracy suffered when a straightforward baseline configuration of retrieval was upgraded with a longer and longer conversational context, or with noise when cognitively irrelevant chunks were included; the collateral implications of the noise has a negative impact both on retrieval quality and downstream grounded generation within the LLM. Failure of factual consistency (Faithfulness) indicates that the LLM failed to understand or even accurately generate based on the relevant retrieved context. Performance can also vary depending on configurable Component parameters, including embedding model type and retrieval depth (k), suggesting that minimum accuracy settings also represent at least some future source of retrieval error.

Variability in performance was observed as a function of document size, page length, complexity, and the number of user queries. The reported values are based on baseline con- dition. Performance will vary based on LLM type, embedding model, chunk size, and retrieval (k). These factors may im- prove accuracy but will be computationally more demanding. In the future, DocMind will move toward adaptive tuning parameters, changes for specific domains, and multilingual capability, which should produce more robust performance across the types of documents and queries encountered in the real world.

5. CONCLUSION AND FUTURE SCOPE

This paper outlines DocMind, an artificial intelligence (AI) powered document analysis tool that uses Retrieval-Augmented Generation (RAG) methods to perform document query systems using PDF files. Example integrated systems that provide semantic searching capabilities, embedding retrieval and generative models provide answers to queries directly, and are intelligent. Unlike systems that rely solely on keyword searching, DocMind retrieval precision has enhanced user focus. The personalized adaptive systems provide an effective and intelligent method of browsing multi-PDF collections in excess of 500 pages in an efficient manner.

Envisioned in this study are broader system integration beyond current capabilities of DocMind for each standalone functionality to aid students, researchers, and practitioners interacting with PDFs for specific disciplinary purposes. PDF document applications are intended to facilitate productivity and streamline decision-making within the respective context through functionality such as contextual PDF searching, summarization and dialogic interfaces.

Opportunities in future development include expanding the multimodal document understanding framework (including

TABLE 2
QUANTITATIVE EVALUATION METRICS

Model	Recall@5	Avg ROUGE-L	Avg BERT F1	Avg Cosine Similarity	Avg Recall	Avg Faithfulness	Avg Relevance	Citation Correctness	MRR	Avg Accuracy	nDCG@5
BM25	0.733	0.073	0.383	0.383	0.599	0.228	0.383	0.733	0.533	0.080	0.423
TF-IDF	0.733	0.093	0.462	0.462	0.799	0.278	0.462	0.733	0.633	0.086	0.478
Gemini-2.0-Flash LLM (raw)	0.000	0.112	0.552	0.552	0.408	0.332	0.552	0.000	0.000	0.151	0.000
DocMind (Gemini-2.0-Flash + RAG)	0.733	0.549	0.745	0.745	0.869	0.647	0.745	0.733	0.700	0.633	0.527

tables, images, and scanned PDF files), contextualizing the tool for the purposes of discipline (for example, law, health- care, or scientific publishing), and expanding support for multilingual users that an international audience can engage with. An additional potential is to support real-time collaborative query designing and reasoning capacity through integration with knowledge graphs. Implementing improved summarization techniques can also provide meaningful insights from extensive data sets. In this prospective application, the overarching function of DocMind could be to create a fully-complete artificial intelligence assistant for document intelligence. Expanding on limitations of existing products for the analysis of PDF files, DocMind could present a fully-complete solution for enhancing accuracy, grounding replies to original documents, and supporting informed decisions in many contexts. Overall, the reality of such a complete capacity would radically change how individuals and organizations interacted with unstructured information, found in PDF files, supporting better access, access that is more trustworthy in the digital age.

6. REFERENCES

[1] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. ArXiv preprint .2023

[2] Nuno Miguel Guerreiro, Duarte M. Alves, Jonas Waldendorf, Barry Haddow, Alexandra Birch, Pierre Colombo, and Andre´ F. T. Martins. 2023. Hallucinations in Large Multilingual Translation Models. ArXiv preprint. 2023.

[3] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiao Cheng Feng, Bing Qin, and Ting Liu. 2023. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. arXiv preprint. 2023.

[4] Mohammed-Khalil Ghali, Abdelrahman Farrag, Daehan Won, and Yu Jin. 2024. Enhancing Knowledge Retrieval with In-Context Learning and Semantic Search through Generative AI. arXiv preprint. 2024.

[5] A. Rai, "Retrieval-Augmented Generation (RAG): Trends, Architectures, and Use Cases — A Comprehensive Study," Journal Article, International Journal of Novel Research and Development (IJNRD), vol. 10, no. 6, June 2025.

[6] S. Kukreja, T. Kumar, V. Bharate, A. Purohit, A. Dasgupta, and D. Guha, "Vector Databases and Vector Embeddings—Review," in Proceedings of the 2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP), IEEE, Dec. 2023

[7] J. Lala, O. O'Donoghue, A. Shtedritski, S. Cox, S. G. Rodriques, and A. D. White, "PaperQA: Retrieval-Augmented Generative Agent for Scientific Research," arXiv preprint, Dec. 2023.

[8] R. Cao, H. Zhang, T. Huang, Z. Kang, Y. Zhang, L. Sun, H. LLY, M. Miao, S. Fan, L. Chen, and K. Yu, "NeuSym RAG: Hybrid Neural Symbolic Retrieval with Multiview Structuring for PDF Question Answering," arXiv preprint, May 2025.

[9] K. Roy, Y. Zi, C. Shyalika, R. Prasad, S. Murali, V. Palit, and A. Sheth, "QA-RAG: Leveraging Question and Answer-based Retrieved Chunk Reformatting for Improving Response Quality During Retrieval-augmented Generation," Journal article, 2024.

[10] Z. Najwa, M. Ghazouani, and N. Chafiq, "Revolutionizing Information Retrieval: Unveiling a Next Generation AI-powered Question-Answer System for Comprehensive Document Analysis," Journal of Theoretical and Applied Information Technology, vol. 102, no. 24, pp. 1–10, Dec. 2024.

[11] A. A. Khan, M. T. Hasan, K. K. Komell, J. Rasku, and P. Abrahamsson, "Developing Retrieval Augmented Generation (RAG) based LLM Systems from PDFs: An Experience Report," arXiv preprint, Oct. 2024.

[12] S. Roy, M. Goswami, N. Nargund, S. Mohanty, and P. K. Pattnaik, "Conversational Text Extraction with Large Language Models Using Retrieval-Augmented Systems," Conference paper, 2024.

[13] P. Prannaveshwaaran and B. Kalaiselvan, "Chat with PDF Using LangChain and AstraDB," International Research Journal of Modernization in Engineering, Technology and Science, vol. 6, no. 11, Nov. 2024.

[14] N. Kim, T. You, S. Oh, S. Kim, and H. Kim, "Donut: Document Understanding Transformer without OCR," in European Conference on Computer Vision (ECCV), 2022.

[15] R. Han, Y. Zhang, P. Qi, Y. Xu, J. Wang, L. Liu, W. Y. Wang, B. Min, and V. Castelli, "RAG-QA Arena: Evaluating Domain Robustness for Long-form Retrieval Augmented Question Answering," arXiv preprint, Jul. 2024.

[16] S. Jeong, J. Baek, S. Choi, S. J. Hwang, and J. C. Park, "Adaptive-RAG: Learning to Adapt Retrieval-Augmented Large Language Models through Question Complexity," Conference paper, 2024.

[17] D. Lin, "Revolutionizing Retrieval-Augmented Generation with Enhanced PDF Structure Recognition," Whitepaper,Jan.2024.