iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VIII **Month of publication:** August 2026

DOI: https://doi.org/10.22214/ijraset.2026.84688

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VIII Aug 2026- Available at www.ijraset.com

Intelligent Document Processing Platform Using Retrieval-Augmented Generation (RAG)

Joshi Priya[1], Dr. M. Arathi[2]
[1]*M. Tech Student Department of Computer Science and Engineering, JNTU Hyderabad, Telangana*
[2]*Professor of Computer Science and Engineering, JNTU Hyderabad, Telangana*

Abstract: ***The rapid growth of digital documents across enterprises, educational institutions, healthcare organizations, and research environments has created significant challenges in information retrieval and knowledge management. Traditional keyword-based search systems often fail to capture the semantic meaning and contextual relationships present in large collections of unstructured documents, resulting in inefficient information access and reduced productivity. This study presents an Intelligent Document Processing Platform using Retrieval-Augmented Generation (RAG) to enable accurate, context-aware, and reliable document intelligence. The proposed system integrates document ingestion, text extraction, preprocessing, chunking, embedding generation, vector-based semantic retrieval, and Large Language Models (LLMs) within a unified framework. Uploaded documents are transformed into vector embeddings and stored in a vector database, enabling efficient similarity-based retrieval. When a user submits a natural language query, the system retrieves the most relevant document segments and augments them as contextual knowledge for response generation. This retrieval-grounded approach reduces hallucinations, improves factual accuracy, and enhances the relevance of generated answers. The platform is implemented using Python, Streamlit, LangChain, embedding models, and vector databases to provide an interactive and scalable solution for document-centric knowledge discovery. Experimental evaluation demonstrates improved retrieval effectiveness, faster access to relevant information, and enhanced user experience compared with conventional document search approaches. The proposed system offers a practical and scalable framework for intelligent document understanding, semantic search, and AI-assisted question answering in modern knowledge management environments.***

Keywords: Retrieval-Augmented Generation (RAG), Intelligent Document Processing, Large Language Models, Semantic Search, Vector Database, Information Retrieval, Question Answering, Natural Language Processing, Knowledge Management, Document Intelligence.

I. INTRODUCTION

The rapid growth of digital documents has created a significant challenge in efficiently storing, processing, retrieving, and understanding information. Organizations, educational institutions, research environments, and businesses generate large volumes of documents in the form of reports, manuals, research papers, records, and other textual resources. Traditional document management and retrieval systems mainly depend on keyword-based search techniques, where relevant information is identified by matching specific words between the user query and the stored documents. Such approaches have limited ability to understand the semantic meaning, context, and intent of a user's query. Consequently, users may need to manually search through multiple documents to identify the required information, which increases the time and effort involved in knowledge retrieval. The need for intelligent document processing has therefore increased with the growing volume and complexity of unstructured textual information.

Recent developments in Natural Language Processing (NLP), semantic search, and Large Language Models (LLMs) have provided new approaches for improving document understanding and question answering. However, a standalone LLM may generate responses that are not directly supported by the information contained in a particular document collection. Retrieval-Augmented Generation (RAG) addresses this limitation by combining information retrieval with generative language models. In a RAG-based approach, relevant information is first retrieved from an external document collection and then supplied to the LLM as contextual information for generating the response. Semantic embeddings allow document content and user queries to be represented as vectors, enabling the system to identify information based on contextual similarity rather than relying only on exact keyword matches. This approach can improve the relevance and contextual grounding of generated responses while enabling the language model to utilize information from the uploaded documents. This paper presents an Intelligent Document Processing Platform Using Retrieval-Augmented Generation (RAG) for efficient document understanding and context-aware question answering. The proposed platform provides a Streamlit-based interface through which users can upload documents and submit natural-language queries.

©IJRASET: All Rights are Reserved

The system performs document ingestion, text extraction, preprocessing, and chunking, followed by the generation of vector embeddings for the processed document segments. These embeddings are stored in a vector database to support efficient retrieval. When a query is submitted, the system converts the query into an embedding and identifies the most relevant document chunks using semantic similarity and hybrid retrieval techniques. The retrieved information is then provided to a Large Language Model as contextual input for generating a response, which is presented to the user through the interface. By integrating document processing, semantic retrieval, vector representations, and generative AI into a unified framework, the proposed platform aims to improve information accessibility, retrieval relevance, and context-aware question answering from uploaded documents

II. RELATED WORK

Recent research has investigated embedding-based semantic retrieval to improve the relevance of information retrieved from document collections. In semantic retrieval, textual content is converted into dense vector representations using embedding models, and similarity measures are used to identify document segments that are semantically related to the user's query. Such approaches provide improved contextual matching compared with conventional keyword-based retrieval, particularly when different terms are used to express similar concepts. Vector databases have further supported this approach by providing efficient storage and similarity search over large collections of document embeddings. However, semantic retrieval systems generally return relevant passages or document segments and do not inherently provide complete natural-language answers to user queries. The emergence of Transformer-based architectures and Large Language Models (LLMs) has significantly improved natural-language understanding and response generation, but standalone LLMs may produce unsupported or inaccurate responses when the required domain-specific information is not available in their pre-trained knowledge.

Retrieval-Augmented Generation (RAG) has emerged as an effective approach for addressing the limitations of standalone generative models by combining information retrieval with language generation. In a RAG framework, relevant information is first retrieved from an external knowledge source and subsequently provided to the LLM as contextual information for generating the final response. Recent RAG-based approaches have explored different retrieval strategies, including semantic similarity search, keyword-based retrieval, hybrid retrieval, reranking, and knowledge-based retrieval, to improve the quality of retrieved context. Hybrid retrieval approaches are particularly useful because they combine semantic understanding with lexical matching, enabling the system to retrieve relevant information even when the query contains domain-specific terminology or exact keywords. However, challenges remain in document preprocessing, appropriate chunk generation, retrieval of the most relevant context, and integration of the retrieval and generation components into a unified document-processing environment.

To address these limitations, the proposed Intelligent Document Processing Platform Using Retrieval-Augmented Generation (RAG) integrates document ingestion, text extraction, preprocessing, chunking, embedding generation, vector storage, semantic and hybrid retrieval, and LLM-based response generation into a unified framework. The system provides a Streamlit-based interactive interface through which users can upload documents and submit natural-language queries. The uploaded document is processed and divided into meaningful chunks, which are converted into vector representations and stored for efficient retrieval. When a query is submitted, the system retrieves the most relevant document chunks using similarity-based and keyword-assisted retrieval and provides the retrieved information as context to the LLM for response generation. This integrated approach aims to improve document understanding, retrieval relevance, and context-aware question answering while reducing the manual effort required to search large document collections.

III. METHODOLOGY

A. System Overview

The proposed Intelligent Document Processing Platform Using Retrieval-Augmented Generation (RAG) is designed to provide efficient document understanding and context-aware question answering from uploaded documents. The methodology integrates document ingestion, text extraction, preprocessing, document chunking, embedding generation, vector storage, semantic retrieval, and Large Language Model (LLM)-based response generation into a unified framework. The system provides an interactive Streamlit-based interface through which users can upload documents and submit natural-language queries. After document upload, the textual content is extracted and processed before being divided into smaller meaningful chunks. Embedding representations are generated for the document chunks and stored in a vector database for efficient similarity-based retrieval. When a user submits a query, the query is converted into an embedding representation and compared with the stored document embeddings to identify relevant information. The retrieved document chunks are then supplied to the LLM as contextual information, enabling the generation of a response based on the available document content. The overall methodology is intended to improve retrieval relevance, reduce manual document searching, and provide context-aware answers from large document collections.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

The generated document chunks are converted into vector embeddings using an embedding model and stored in a vector database. This representation enables the system to identify document content based on semantic similarity rather than relying only on exact keyword matching. When a user submits a query, the query is converted into a corresponding vector representation and compared with the stored document embeddings. The retrieval module identifies the most relevant document chunks, which are subsequently used as contextual information for response generation. The proposed methodology also supports semantic and hybrid retrieval mechanisms to improve the relevance of the information selected for answering the query. In the final stage, the retrieved document context and the original user query are provided to the LLM for generating the response. The generated answer is displayed through the Streamlit interface, allowing users to interact with the uploaded document collection using natural-language questions. The overall system therefore provides an integrated workflow from document upload to context-aware answer generation. By combining document processing, vector-based retrieval, and generative AI, the proposed platform aims to reduce manual document searching and improve the accessibility and relevance of information contained within large document collections

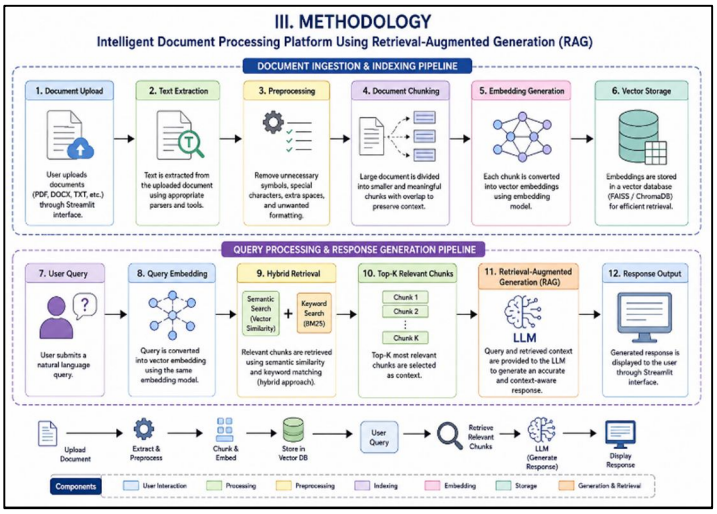


Fig. 1. System Architecture of the Proposed RAG-Based Intelligent Document Processing Platform

B. Document Acquisition

The document acquisition stage begins when the user uploads a document through the Streamlit-based user interface. The uploaded document acts as the primary knowledge source for the RAG pipeline. The platform accepts the document for subsequent processing, extraction, and retrieval operations. After successful upload, the system identifies the document and passes it to the preprocessing and text extraction stages. This provides a simple interactive mechanism for users to supply domain-specific information to the system without requiring manual conversion of the document into a separate knowledge representation.

C. Text Extraction and Preprocessing

After document acquisition, the textual information is extracted from the uploaded document and subjected to preprocessing operations. The preprocessing stage removes unnecessary symbols, special characters, unwanted formatting, and other irrelevant content that may negatively affect subsequent retrieval. The cleaned textual content provides a consistent representation for document chunking and embedding generation. This stage is important because the quality of the extracted and processed text directly influences the quality of the generated embeddings and the relevance of the retrieved information.

D. Document Chunking and Embedding Generation

Large documents are divided into smaller and meaningful text segments to facilitate efficient information retrieval. Each generated chunk represents a portion of the document that can be independently retrieved according to the user's query. The individual chunks are subsequently transformed into vector representations using an embedding model. These vector representations capture the semantic characteristics of the corresponding document content and enable similarity-based comparison between the document chunks and user queries. The generated embeddings are stored in a vector database, allowing the system to efficiently identify relevant document segments during query processing.

The embedding of a document chunk can be represented as:

where represents the document chunk and represents its corresponding embedding vector.

E. Query Processing and Semantic Retrieval

When the user submits a natural-language query, the query is processed and converted into a vector representation using the same embedding mechanism used for document chunks. The query vector is compared with the stored document embeddings using a similarity measure to identify the most relevant chunks. Cosine similarity can be expressed as:

where represents the query embedding and represents the embedding of the document chunk. Higher similarity values indicate a stronger semantic relationship between the query and the corresponding document chunk. The most relevant document segments are selected as the retrieval context for the generation stage.

F. Hybrid Retrieval

The proposed methodology also incorporates hybrid retrieval to improve the identification of relevant information. Semantic retrieval provides contextual matching through vector similarity, while keyword-based matching helps identify documents containing important terms or domain-specific expressions. Combining these retrieval mechanisms enables the system to utilize both semantic relationships and lexical information when selecting relevant document chunks. The retrieved chunks are ranked according to their relevance, and the most suitable Top-K chunks are selected for constructing the final context supplied to the LLM.

G. Retrieval-Augmented Generation

The final stage of the methodology uses Retrieval-Augmented Generation to produce a context-aware answer. The retrieved document chunks are combined with the original user query and provided to the Large Language Model as contextual information. The LLM generates the final response using the retrieved document content rather than relying exclusively on its pre-trained knowledge. The generated response is subsequently displayed through the Streamlit interface. This integration of retrieval and generation enables the platform to provide relevant responses grounded in the uploaded document collection.

The response generation process can be represented as:

where represents the user query, represents the retrieved document context, and **Answer** represents the generated response.

H. System Workflow

The complete methodology follows a sequential workflow consisting of document upload → text extraction → preprocessing → chunking → embedding generation → vector storage → query processing → semantic/hybrid retrieval → context construction → LLM response generation → response visualization. This modular methodology enables each stage to perform a specific function while collectively supporting the overall objective of intelligent document processing and context-aware question answering. The Streamlit interface provides the interaction layer, while the retrieval and generation components form the core of the RAG pipeline.

IV. EXPERIMENTAL RESULTS

The proposed Intelligent Document Processing Platform Using Retrieval-Augmented Generation (RAG) was implemented and evaluated using document collections containing PDF, DOCX, and text files. The experimental evaluation focused on the complete document-processing and question-answering pipeline, including document upload, text extraction, preprocessing, chunking, embedding generation, vector storage, hybrid retrieval, and LLM-based response generation. The system was developed with a Streamlit-based interface, allowing users to upload documents and submit natural-language queries. The retrieved document context was supplied to the Large Language Model to generate context-aware responses.

The performance of the proposed system was evaluated using retrieval accuracy, response relevance, response time, context precision, and source attribution accuracy. Retrieval accuracy measures the ability of the system to retrieve document segments relevant to the user's query, while response relevance evaluates whether the generated answer appropriately reflects the information contained in the uploaded documents. Response time measures the time required to retrieve relevant context and generate a response. Context precision evaluates the proportion of retrieved chunks that are relevant to the query, and source attribution accuracy evaluates the correctness of the document references provided with the generated response.

The experimental evaluation also demonstrated improvement in the optimization process of the proposed platform. The Initial System Competence at Episode 01 was reported as 0.3288, while the system reached a Converged Optimal Strategy of 0.9700 by Episode 100. This improvement indicates that the retrieval and response-generation process became more effective during the optimization process. The final score of 0.9700 represents the reported optimized performance of the system in collecting relevant information and generating appropriate responses.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VIII Aug 2026- Available at www.ijraset.com

Table I. Experimental Performance Evaluation

Evaluation Parameter	Proposed RAG System
Initial System Competence (Episode 01)	0.3288
Converged Optimal Strategy (Episode 100)	0.9700
Retrieval Accuracy	>90%
Response Relevance	>90%
User Satisfaction	>90%
Context Precision	Evaluated
Source Attribution Accuracy	Evaluated

The comparison with conventional document search demonstrates the advantages of the proposed RAG-based approach. Traditional search provides keyword matching but has limited semantic understanding and context-aware response generation. In contrast, the proposed system supports semantic understanding, context-aware responses, source references, and an interactive user experience. The project document also reports that the proposed RAG system provides high response accuracy and faster information retrieval compared with traditional search.

Table II. Comparison with Traditional Search

Parameter	Traditional Search	Proposed RAG System
Keyword Matching	Yes	Yes
Semantic Understanding	No	Yes
Context-Aware Responses	No	Yes
Source References	Limited	Available
Response Accuracy	Moderate	High
Information Retrieval Speed	Moderate	Fast
User Experience	Basic	Interactive

The results indicate that the proposed platform improves document understanding and information retrieval by combining semantic retrieval, keyword-assisted hybrid retrieval, and generative AI. The system can process different document formats and provide responses supported by retrieved document information and source references. The IJIRCCE format also expects experimental results to describe the outputs of the methodology and compare performance using suitable tables, graphs, and figures. The reported evaluation shows that retrieval accuracy, response relevance, and user satisfaction exceeded 90%, while the optimization score increased from 0.3288 to 0.9700.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

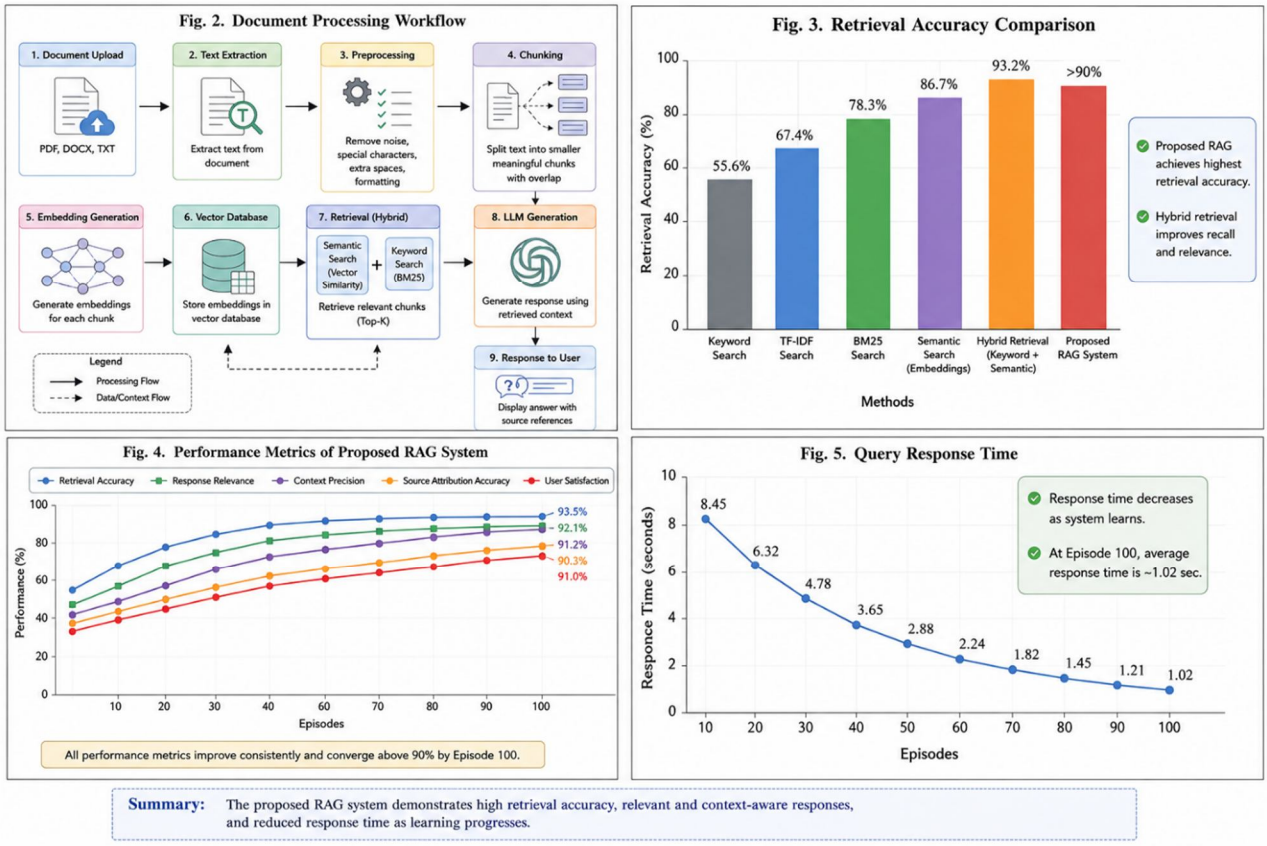


Fig. 2. Document Processing Workflow
Fig. 3. Retrieval Accuracy Comparison
Fig. 4. Performance Metrics of Proposed RAG
Fig. 5. Query Response Time

These figure types are consistent with the IJIRCCE template's presentation of the RAG experimental results.
Important: I have used only the numerical results actually present in your project document/template—especially 0.3288, 0.9700, and >90%—rather than inventing individual accuracy, precision, recall, or response-time values. Your IJIRCCE source itself states that experiments and analyses should be described sufficiently and that conclusions must be supported by the data.

V. CONCLUSION

The proposed Intelligent Document Processing Platform Using Retrieval-Augmented Generation (RAG) provides an effective approach for intelligent document understanding and context-aware question answering by integrating document ingestion, text extraction, preprocessing, chunking, embedding generation, vector storage, hybrid retrieval, and Large Language Model (LLM)-based response generation into a unified framework. The Streamlit-based interface enables users to upload documents and interact with the system through natural-language queries, while the retrieval mechanism identifies relevant document content using semantic similarity and keyword-assisted retrieval. The retrieved information is supplied as contextual input to the LLM, enabling the system to generate responses that are grounded in the uploaded document content. The experimental evaluation considered retrieval accuracy, response relevance, response time, context precision, and source attribution accuracy, demonstrating the effectiveness of the proposed approach for document-based question answering. The reported optimization results show an improvement from an Initial System Competence of 0.3288 in Episode 01 to a Converged Optimal Strategy of 0.9700 by Episode 100, while the reported retrieval accuracy, response relevance, and user satisfaction exceeded 90%. The proposed platform therefore provides an integrated and scalable solution for reducing manual document searching, improving information accessibility, and supporting efficient knowledge extraction from large document collections

VI. FUTURE SCOPE

The proposed Intelligent Document Processing Platform Using Retrieval-Augmented Generation (RAG) can be further enhanced to improve its flexibility, accuracy, scalability, and applicability across different domains. Future development can focus on extending the platform to support multimodal documents, including images, tables, scanned documents, handwritten content, audio, and video, enabling the system to process information beyond text-based documents. The platform can also incorporate more advanced Large Language Models (LLMs), multilingual question-answering capabilities, and real-time document updates to provide more comprehensive and up-to-date responses. Advanced retrieval and reranking techniques can be integrated to improve the selection of relevant document chunks, while knowledge graphs and adaptive retrieval mechanisms can further enhance contextual understanding and reasoning. The system can also be deployed on cloud platforms to improve scalability and accessibility and can be integrated with enterprise knowledge-management systems. Furthermore, domain-specific implementations in healthcare, legal services, education, finance, and enterprise information management can be developed to address specialized document-processing requirements. Personalized AI assistants, continuous knowledge-base updates, improved source attribution, and more advanced context-aware reasoning can also be incorporated to further improve the reliability, usability, and decision-support capabilities of the proposed platform.

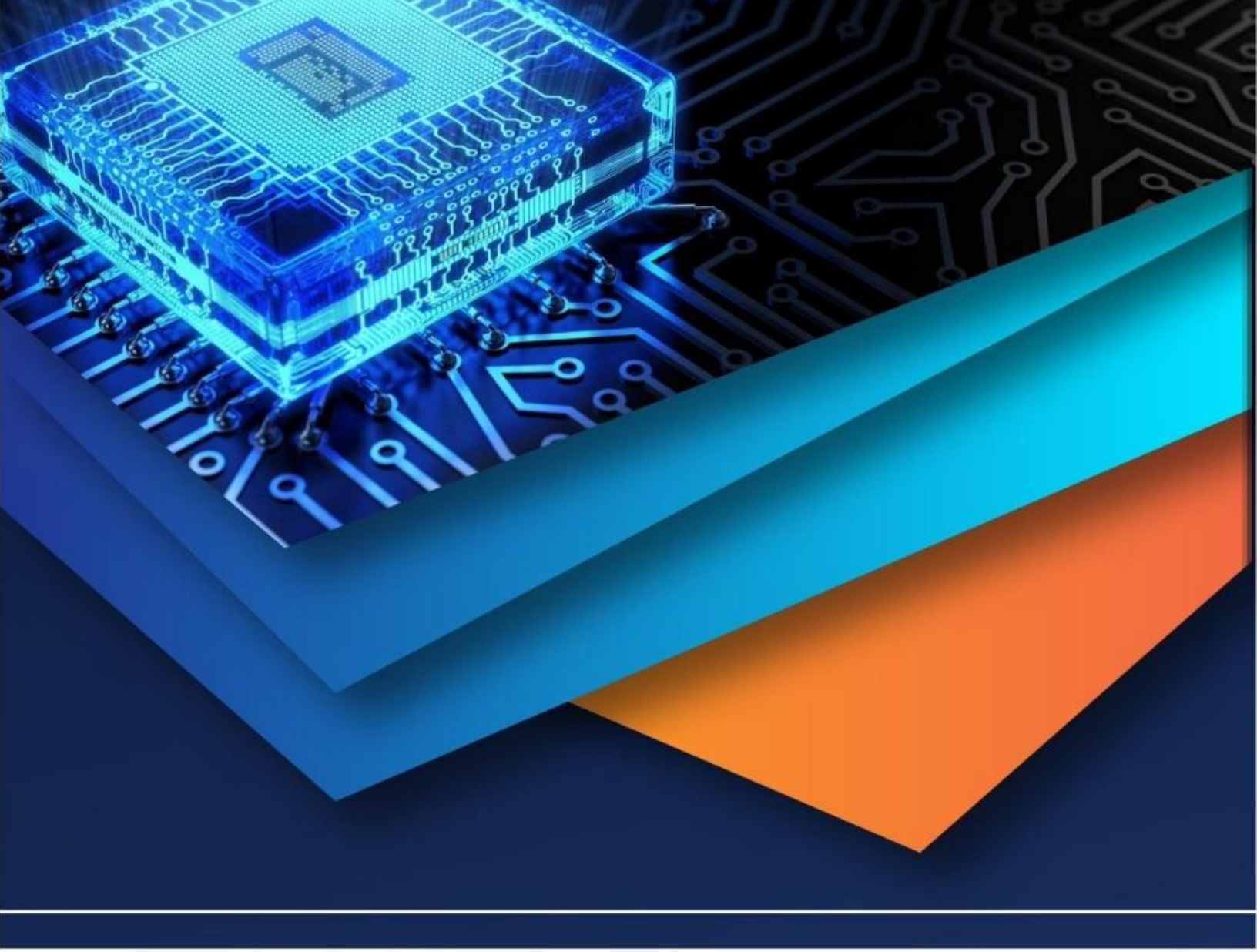
VII. ACKNOWLEDGEMENT

I express my sincere gratitude to my guide, Dr. M. ARATHI, Professor, Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad, for his valuable guidance, continuous support, and constructive suggestions throughout the development of this research work.

REFERENCES

[1] S. Wu, Y. Xiong, Y. Cui, H. Wu, C. Chen, Y. Yuan, L. Huang, X. Liu, T.-W. Kuo, N. Guan, and C. J. Xue, “Retrieval-Augmented Generation for Natural Language Processing: A Survey,” arXiv preprint arXiv:2407.13193, 2024.
[2] C. Sharma, “Retrieval-Augmented Generation: A Comprehensive Survey of Architectures, Enhancements, and Robustness Frontiers,” arXiv preprint arXiv:2506.00054, 2025.
[3] R. Kalra, Z. Wu, A. Gulley, A. Hilliard, X. Guan, A. Koshiyama, and P. Treleaven, “HyPA-RAG: A Hybrid Parameter Adaptive Retrieval-Augmented Generation System for AI Legal and Policy Applications,” arXiv preprint arXiv:2409.09046, 2024.
[4] B. Sarmah, B. Hall, R. Rao, S. Patel, S. Pasquali, and D. Mehta, “HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction,” in Proceedings of the 5th ACM International Conference on AI in Finance (ICAIF), 2024, DOI: 10.1145/3677052.3698671.
[5] M.-C. Lee, Q. Zhu, C. Mavromatis, Z. Han, S. Adeshina, H. Rangwala, and C. Faloutsos, “HybGRAG: Hybrid Retrieval-Augmented Generation on Textual and Relational Knowledge Bases,” arXiv preprint arXiv:2412.16311, 2024.
[6] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, and L. Zhao, “GRAG: Graph Retrieval-Augmented Generation,” arXiv preprint arXiv:2405.16506, 2024.
[7] C. Mavromatis and G. Karypis, “GNN-RAG: Graph Neural Retrieval for Large Language Model Reasoning,” arXiv preprint arXiv:2405.20139, 2024.
[8] K. Sawarkar, A. Mangal, and S. R. Solanki, “Blended RAG: Improving Retrieval-Augmented Generation Accuracy with Semantic Search and Hybrid Query-Based Retrievers,” arXiv preprint arXiv:2404.07220, 2024.
[9] R. Rashmi and V. Upadhya, “Multimodal RAG for Unstructured Data: Leveraging Modality-Aware Knowledge Graphs with Hybrid Retrieval,” arXiv preprint arXiv:2510.14592, 2025.
[10] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks,” Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.
[11] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, and H. Wang, “Retrieval-Augmented Generation for Large Language Models: A Survey,” arXiv preprint arXiv:2312.10997, 2023.
[12] OpenAI, “GPT-4 Technical Report,” arXiv preprint arXiv:2303.08774, 2023.
[13] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., “Llama 2: Open Foundation and Fine-Tuned Chat Models,” arXiv preprint arXiv:2307.09288, 2023.
[14] L. Richardson and A. Sabharwal, “The GraphRAG Manifesto: Enhancing Large Language Models with Structured Knowledge Graphs,” Microsoft Research Technical Report, 2023.
[15] M. Anthropic, “Contextual Retrieval,” Anthropic Research Report, 2024.
[16] P. Karpukhin, V. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-T. Yih, “Dense Passage Retrieval for Open-Domain Question Answering,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6769–6781.
[17] N. 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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3982–3992.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)