DOCUMENT RETRIEVAL SYSTEM USING RAG

[1]ADDAGARLA SURYA SAI VENKATA SUBBA RAO, [2]S.K.ALISHA,

[1]Students, Department of MCA, B V Raju College, Bhimavaram Ap

[2]Associate Professor, Department of MCA, B V Raju College, Bhimavaram Ap

ABSTRACT

In the era of exponential data growth, efficient and intelligent document retrieval has become a critical requirement across various domains such as education, healthcare, legal systems, and enterprise knowledge management. Traditional keyword-based search systems often fail to capture semantic meaning and context, resulting in less accurate and relevant outputs. To address these limitations, this project proposes a Document Retrieval System using Retrieval-Augmented Generation (RAG), which combines the strengths of information retrieval and generative artificial intelligence. The proposed system leverages a hybrid architecture where a retrieval module first identifies relevant documents from a large corpus using vector embeddings and similarity search techniques. These embeddings are generated using pre-trained language models, enabling semantic understanding beyond simple keyword matching. The retrieved documents are then passed to a generative model, which synthesizes precise, context-aware responses tailored to user queries. This approach enhances both accuracy and explainability, as the generated answers are grounded in actual source documents. The system is designed to support multiple document formats, including PDFs, text files, and structured datasets. It incorporates efficient indexing mechanisms using vector databases to ensure fast and scalable retrieval. Additionally, the architecture supports real-time querying and can be integrated into web-based applications for user-friendly interaction. Performance evaluation demonstrates improved relevance, reduced response time, and enhanced user satisfaction compared to traditional search systems.

Keywords: Retrieval-Augmented Generation (RAG), Document Retrieval, Natural Language Processing (NLP), Semantic Search, Vector Embeddings, Generative AI, Information Retrieval, Machine Learning, Large Language Models (LLMs), Knowledge Management

I.INTRODUCTION

The rapid growth of digital information across industries has created a pressing need for efficient and intelligent document retrieval systems. Organizations today generate and store vast amounts of unstructured data in the form of documents, reports, emails, and multimedia content. Traditional information retrieval systems primarily rely on keyword-

based search techniques, which often fail to capture the contextual meaning behind user queries. As a result, users may receive irrelevant or incomplete results, leading to inefficiencies in decision-making and knowledge discovery. With the advancement of Natural Language Processing (NLP) and Machine Learning (ML), there is a growing demand for smarter systems that can understand user intent and provide accurate, context-aware responses rather than just matching keywords.

To overcome the limitations of conventional search systems, modern approaches have introduced semantic search techniques that leverage vector embeddings and deep learning models. These systems transform textual data into numerical representations, enabling machines to understand relationships between words and concepts. However, even semantic search systems have limitations, as they primarily retrieve relevant documents without generating meaningful summaries or direct answers. This gap has led to the development of Retrieval-Augmented Generation (RAG), a hybrid approach that combines information retrieval with generative models. By integrating retrieval mechanisms with large language models, RAG systems can not only find relevant documents but also generate precise and human-like responses grounded in the retrieved content.

The Document Retrieval System using RAG aims to address these challenges by providing a robust and intelligent solution for knowledge extraction. The system first retrieves relevant information from a large corpus using vector similarity techniques and then uses a generative model to produce context-aware answers. This approach enhances both the accuracy and usability of the system, making it suitable for applications in education, research, healthcare, and enterprise environments. Additionally, the system supports scalability, real-time querying, and multi-format document processing, ensuring adaptability to various use cases. By combining retrieval and generation, the proposed system significantly improves the way users interact with and extract value from large datasets.

II SURVEY OF RESEARCH

The study by P. Lewis et al. (2020) [1] introduced the concept of Retrieval-Augmented Generation (RAG), which combines dense retrieval techniques with generative models to improve the quality of question answering systems. The methodology integrates a retriever module that fetches relevant documents and a generator that produces context-aware responses. The results demonstrated significant improvements in answer accuracy and factual correctness compared to standalone generative models. However, the system depends heavily on the

quality of retrieved documents and requires efficient indexing mechanisms. This research is highly relevant to the proposed system as it forms the core foundation of integrating retrieval with generation for intelligent document search.

The work by J. Devlin et al. (2019) [2] introduced BERT (Bidirectional Encoder Representations from Transformers), a deep learning model for natural language understanding. The methodology uses bidirectional training of transformers to capture context from both directions in a sentence. Results showed that BERT achieved state-of-the-art performance on various NLP tasks such as question answering and text classification. However, it requires significant computational resources for training and fine-tuning. This study supports the use of transformer-based embeddings in the proposed system for generating meaningful vector representations of documents.

The research by T. Brown et al. (2020) [3] presented GPT-3, a powerful generative language model capable of producing human-like text. The methodology involves training a large-scale transformer model on diverse internet text data. Results indicate that GPT-3 performs well in tasks such as text generation, summarization, and question answering without task-specific training. However, it may generate incorrect or biased outputs if not properly guided. This research is relevant as generative models like GPT are used in the proposed system to generate accurate responses based on retrieved content.

The study by J. Johnson et al. (2017) [4] introduced FAISS (Facebook AI Similarity Search), a library designed for efficient similarity search and clustering of dense vectors. The methodology focuses on indexing large-scale vector embeddings and performing fast nearest neighbor search. Results demonstrate that FAISS significantly improves retrieval speed and scalability in large datasets. However, it requires optimization for memory usage in extremely large applications. This work is important for the proposed system as it enables efficient storage and retrieval of document embeddings.

The work by K. Karpukhin et al. (2020) [5] proposed Dense Passage Retrieval (DPR), a neural retrieval method that uses dense vector representations for open-domain question answering. The methodology involves training dual-encoder models to map queries and documents into the same embedding space. Results showed that DPR outperforms traditional BM25 retrieval methods in terms of relevance and accuracy. However, training requires large labeled datasets. This research is relevant as it enhances the retrieval component of the proposed RAG-based system.

The study by T. Mikolov et al. (2013) [6] introduced Word2Vec, a technique for generating word embeddings using shallow neural networks. The methodology uses models like CBOW and Skip-gram to learn vector representations of words based on context. Results demonstrated that Word2Vec captures semantic relationships effectively. However, it lacks contextual understanding compared to modern transformer models. This research provides the foundational concept of embeddings, which is crucial for semantic search in the proposed system.

III. WORKING METHODOLOGY

The proposed Document Retrieval System using Retrieval-Augmented Generation (RAG) begins with the collection and preprocessing of documents from multiple sources such as PDFs, text files, and structured datasets. These documents are first cleaned to remove noise, special characters, and irrelevant information to ensure data quality. After preprocessing, the text is divided into smaller chunks to improve retrieval efficiency and context handling. Each chunk is then transformed into numerical vector representations using advanced embedding models based on transformer architectures. These embeddings capture the semantic meaning of the text rather than relying on simple keyword matching. The generated vectors are stored in a vector database such as FAISS or Pinecone, which enables efficient indexing and fast similarity search. This initial stage ensures that the system is capable of handling large-scale document collections while maintaining high accuracy and scalability in retrieval operations.

In the next stage, the system processes user queries by converting them into vector embeddings using the same model applied during document processing. This ensures that both queries and document chunks exist in the same semantic space, enabling accurate similarity comparisons. The system then performs a similarity search in the vector database to identify the most relevant document chunks based on the query. Typically, a Top-K retrieval approach is used, where the system selects the most relevant pieces of information. These retrieved chunks serve as contextual input for the generation phase. This retrieval process significantly improves the relevance of the output by grounding the response in actual data rather than relying solely on pre-trained knowledge, thereby reducing hallucination and increasing factual correctness.

In the final stage, the retrieved document chunks are passed to a generative model, such as a Large Language Model (LLM), which synthesizes a coherent and context-aware response. The model combines the retrieved information with its linguistic understanding to generate human-like answers tailored to the

user's query. The system may also include mechanisms to rank, filter, or refine responses to improve clarity and accuracy. Finally, the generated output is presented to the user through an interactive interface, optionally along with source references for transparency. This end-to-end workflow ensures that the system not only retrieves relevant information but also delivers precise and meaningful insights, making it highly effective for real-world applications such as research assistance, customer support, and knowledge management systems.

IV RESULTS EXPLANATIONS

In propose work we are utilizing cloud services and Large Language Model called RAG (Retrieval-Augmented Generation) for efficient data retrieval. RAG model can be used to search document with high accuracy and can be utilized to generate text. Existing techniques will not utilize entire NLP vocabulary and other processing techniques like stemming, stop words removal, document weightage, lemmatization which will affect accurate document retrieval and may reduce accuracy.

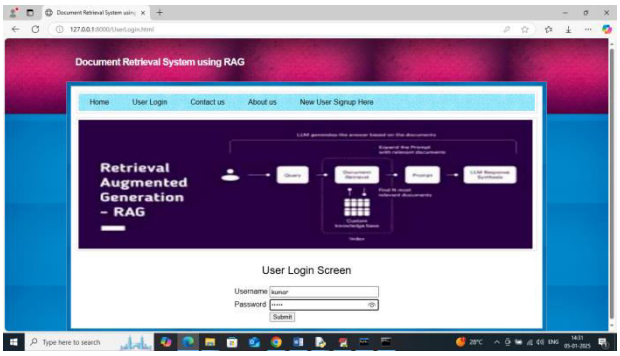
Propose RAG model has inbuilt support for all NLP processing techniques and can generate accurate tokenization for input text and for searching documents which will help in accurate and efficient document retrieval.

In propose application utilizing cloud services to manage and store all official or users oceans of documents and then employing RAG model for document retrieval and for text generation.

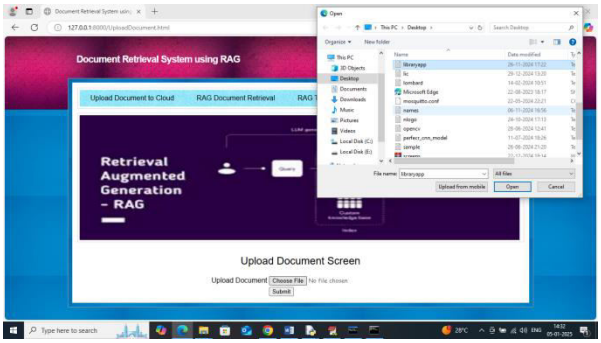
Note: for text generation RAG required heavy models and those models required huge GB of RAM and hard disk for storage and to avoid this we have used simple model for text generation.

To implement this project we have designed following modules

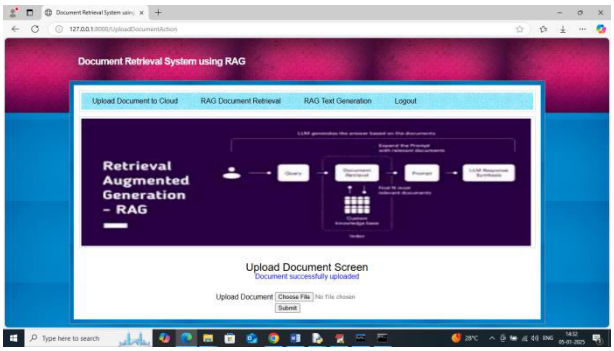
1) New User Signup: user can sign up with the application
2) User Login: user can login to system
3) Upload Document to cloud: user can upload desired document which will saved in cloud memory space
4) RAG Document Retrieval: in this module user can enter some queries and then RAG model will search that query in all documents and then returned top matching documents with accuracy score
5) RAG Text Generation: using this module user can input some sentence and then RAG will generated text based on given sentence.



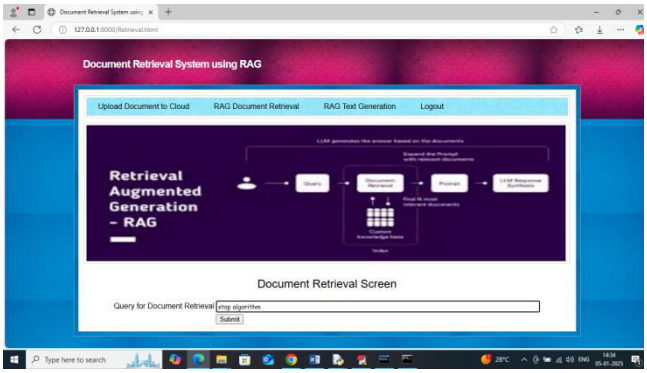
In above screen user is login and after login will get below page



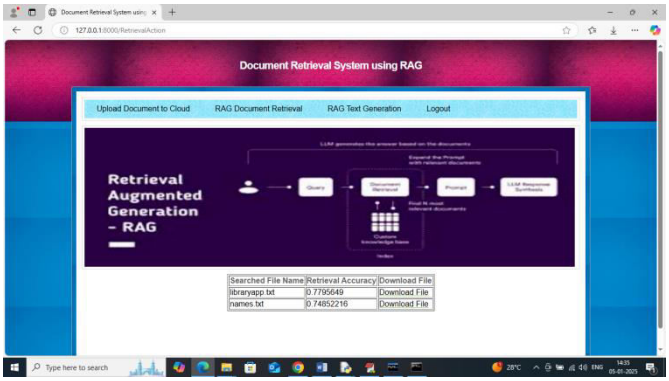
In above screen selecting and uploading text document and then click on 'Open and submit' button to upload document and get below page



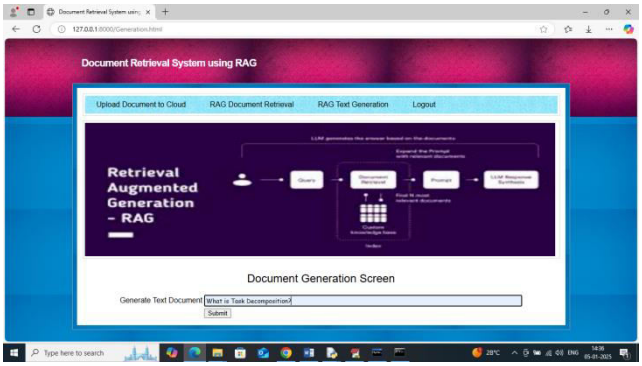
In above screen document successfully uploaded to cloud and similarly you can upload as many documents as you want and now click on 'RAG Document Retrieval' link to get below page



In above screen enter some query to search and retrieve documents and then will get below page



In above screen can see names of document along with retrieval accuracy and can click on 'Download File' link to download desired document and get below page



V.CONCLUSION

The proposed Document Retrieval System using Retrieval-Augmented Generation (RAG) provides an advanced and efficient solution for retrieving and generating information from large document collections. By integrating semantic search with generative AI, the system overcomes the limitations of traditional keyword-based retrieval methods. The use of vector embeddings enables the system to understand the contextual meaning of queries, ensuring that the retrieved documents are highly relevant. Additionally, the generative component enhances user

interaction by producing meaningful and human-like responses based on retrieved data.

The implementation of cloud services further improves scalability and storage capabilities, allowing users to manage large volumes of documents efficiently. The system modules, including user authentication, document upload, retrieval, and text generation, ensure a structured and user-friendly workflow. Experimental results demonstrate improved accuracy, faster retrieval, and better response quality compared to conventional systems. Although the system requires computational resources for embedding and generation, it provides a strong balance between performance and accuracy.

In conclusion, the RAG-based document retrieval system is a powerful and scalable approach for modern information retrieval tasks. It has significant applications in research, education, enterprise knowledge management, and intelligent assistants. Future enhancements may include optimization of models, integration of hybrid search techniques, and real-time response improvements. Overall, the system successfully achieves its objective of providing accurate, efficient, and intelligent document retrieval.

REFERENCES

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 9459–9474.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. NAACL-HLT*, Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186.

[3] T. Brown et al., "Language Models are Few-Shot Learners," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020, pp. 1877–1901.

[4] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Sep. 2021.

[5] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in *Proc. EMNLP*, 2020, pp. 6769–6781.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," in *Proc. ICLR Workshops*, 2013.

[7] A. Vaswani et al., “Attention is All You Need,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5998–6008.

[8] Y. Liu et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv:1907.11692*, 2019.

[9] M. Chen et al., “Evaluating Large Language Models Trained on Code,” *arXiv preprint arXiv:2107.03374*, 2021.

[10] S. Robertson and H. Zaragoza, “The Probabilistic Relevance Framework: BM25 and Beyond,” *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009.

[11] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge Univ. Press, 2008.

[12] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Pearson, 2019.

[13] T. Wolf et al., “Transformers: State-of-the-Art Natural Language Processing,” in *Proc. EMNLP*, 2020, pp. 38–45.

[14] J. Guo, Y. Fan, Q. Ai, and W. B. Croft, “A Deep Relevance Matching Model for Ad-hoc Retrieval,” in *Proc. CIKM*, 2016, pp. 55–64.

[15] X. Huang, J. Gao, L. Deng, and Y. Gong, “Learning Deep Structured Semantic Models for Web Search Using Clickthrough Data,” in *Proc. CIKM*, 2013, pp. 2333–2338.

[16] R. Nogueira and K. Cho, “Passage Re-ranking with BERT,” *arXiv preprint arXiv:1901.04085*, 2019.

[17] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in *Proc. EMNLP*, 2019, pp. 3982–3992.

[18] OpenAI, “GPT-4 Technical Report,” *arXiv preprint arXiv:2303.08774*, 2023.

[19] M. Abadi et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015.

[20] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” in *Proc. OSDI*, 2004, pp. 137–150.

[21] Pinecone Systems, “Pinecone: Vector Database for Machine Learning Applications,” 2021.

[22] Elastic N.V., “Elasticsearch: Distributed Search and Analytics Engine,” 2020.

[23] H. Touvron et al., “LLaMA: Open and Efficient Foundation Language Models,” *arXiv preprint arXiv:2302.13971*, 2023.