International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

AI based Content Summarization using RAG

Dr. G. Vinoda Reddy[1] N. Sai Priya[2] V. Shashank[3] Ch. Mani Kethan Reddy [4]

1 Professor Department of Computer Science and Engineering
(Artificial Intelligence and Machine Learning), Kandlakoya, Hyderabad.

2 Student Department of Computer Science and Engineering
(Artificial Intelligence and Machine Learning), Kandlakoya, Hyderabad.

3 Student Department of Computer Science and Engineering
(Artificial Intelligence and Machine Learning), Kandlakoya, Hyderabad.

4 Student Department of Computer Science and Engineering
(Artificial Intelligence and Machine Learning), Kandlakoya, Hyderabad.

Abstract: - The increased use of digital documents in academic, technical, and professional fields has resulted in a demand for systems that can provide concise, document, based summaries answering user queries. Large language models alone for document, based question answering generally give answers that are only loosely based on the source content, thus, Retrieval, Augmented Generation (RAG) is used. In this paper, the author presents a content summarization tool that is entirely dependent on PDF, DOCX, and text files uploaded by end users, and that adapts its retrieval behavior based on inferred user expertise.The documents are passed through a well, organized ingestion pipeline consisting of text extraction, overlapping semantic chunking, and dense embedding with a lightweight sentence, level transformer, while the similarity search is performed by a FAISS [4, 5] index over normalized embeddings. A heuristic cognitive assessment module examines query features such as length, use of technical terms, clarification patterns, and interaction history to identify the user expertise and, thus, retrieves between two and four relevant document chunks for prompt construction.The answers are generated by a language model running locally and equipped with real, time streaming [10] support. The trials demonstrate that adaptive retrieval leads to higher

Key words: Document Summarization, Adaptive Retrieval, Cognitive State Estimation, Vector Similarity Search, Large Language Models, Interactive Question Answering.

I. INTRODUCTION

The ever, expanding collection of digital documents has turned the demand for systems that can provide clea, r context aware summaries and even answers straight from the user, supplied content, to be very high. In the past document summarization tools only generated one single, static summary. Lately, many question answering systems have been built upon large pretrained models but without structured retrieval, thus resulting in answers that are only loosely backed up by the source documents or being quite inefficient if the texts are long.

Retrieval, augmented generation [1] has been proposed as a solution to these issues by combining semantic retrieval with language model generation. However, most methods based on RAG, only use a fixed retrieval depth for all users and query types, thus disregarding the variations in user expertise or query complexity.

Indeed, in the real world usage scenarios, simple questions usually need very little contextual information while complicated or technical questions require a broader coverage. Hence, applying the same retrieval policy to all can either lead to a shower of background information for naive users or it may constrain the completeness of the answers for advanced users. To get rid of this limitation, a content summarization and question answering system is put forward that dynamically adjusts its retrieval strategies, the, guided, by a user's.

II. RELATED WORKS

Document summarization methods have transformed from extractive methods based on lexical or positional heuristics to neural abstractive methods. The latter increased the fluency of the output but frequently generated ungrounded facts. RAG (Retrieval, augmented Generation) was developed to address this issue. It combines dense retrieval with conditional text generation so that the generated answers can be directly grounded in the retrieved document passages. RAG has been demonstrated to be effective in

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

knowledge, intensive tasks; however, most of the current methods use a fixed retrieval depth that is not adjusted according to the complexity of the query or the expertise level of the user.

[1] **Lewis et al. (2020)** introduced **Retrieval-Augmented Generation (RAG)**, a framework that combines neural text generation with external document retrieval to improve factual accuracy and knowledge-intensive NLP task performance.

[2]N. Reimers and I. Gurevych (2019) introduced **Sentence-BERT:**, a Siamese-network-based extension of BERT that generates semantically meaningful sentence embeddings, enabling efficient and accurate sentence similarity and semantic search tasks[3]

[3]Johnson, Douze, and Jégou (2017) introduced highly efficient GPU-based large-scale similarity search techniques, enabling billion-scale vector indexing and nearest-neighbor retrieval, which laid the foundation for the **FAISS** library. **University of Southampton Part III Project Report Template** provides a structured framework for documenting the CS-RAG system, including its architecture, methodology, implementation details, and evaluation process.

[4]Clark et al. (2020) demonstrated that discriminator-based pre-training in ELECTRA achieves more efficient and effective language representation learning, influencing the development of lightweight and high-performance transformer embedding models.

[5]Karpukhin et al. (2020) showed that Dense Passage Retrieval significantly improves open-domain question answering by learning dense vector representations for accurate and efficient document retrieval.

[6]Lin (2019) highlighted the limitations of retrieval-only systems for information-seeking tasks and emphasized the need for integrated retrieval and reasoning approaches, motivating RAG-style hybrid frameworks.

The depth of retrieval has been found to be a major determinant of the quality of the generated text. A too small context might not include some important details, while a too large context might cause loss of relevance and higher costs in inference. Hence, adaptive retrieval mechanisms that can vary the number of retrieved passages depending on the type of query or the intermediate signals of the model have been proposed. It turns out that retrieval depth is more appropriately treated as a parameter that varies with the query rather than as a global constant,

III. METHODOLOGY

The system approach is disclosed as a linear, stepwise pipeline that transforms raw documents into reference, based, context, aware summaries. Each stage of the pipeline is not only dissected to operate separately but also parallelly, thereby providing the entire system with flexibility.

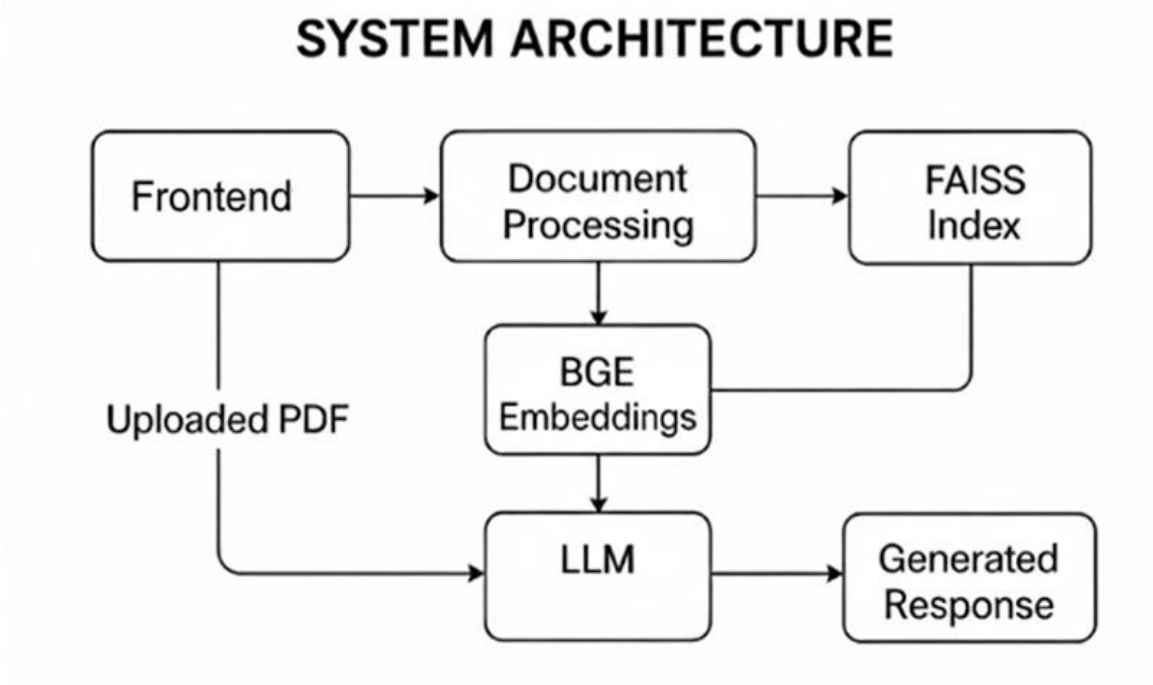
3.2.1 Document Ingestion and Text Processing: After a document is submitted, the ingestion module obtains the unprocessed text content of the document by using layout, specific parsers. The extracted text is then passed through the cleaning stage where pieces of text that are not informative such as excessive whitespace and page headers are removed, while punctuation and sentence structure that are necessary for semantic coherence are retained. Afterwards, the text is divided into chunks of a specified size of about 400 words with an overlap of 80 words. This method of partitioning the text maintains a good trade, off between the granularity of retrieval and computational efficiency and also prevents the separate pieces from containing semantically related information.

3.2.2 Embedding Generation and Vector Indexing :The text segment's content is transformed into a set of vectors via the sentence transformer model all, MiniLM, L6, v2 **[3]** which produces 384, dimensional dense vectors.

In order to let inner product similarity stand for cosine similarity during retrieval, embeddings are initially normalized and then indexed.

IV. SYSTEM ARCHITECTURE:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026



The architecture of the proposed system for content summarization is depicted here. The system is composed of modular elements of a client server model. It tightly integrates document processing, semantic retrieval, and language model generation in one pipeline.

The frontend offers a user, friendly interface. Through this, users can upload documents and enter natural language queries.

Uploaded documents are sent to the backend. There a document processing module is responsible for text extraction, cleaning, and splitting the text into overlapping semantic chunks.

These chunks are subsequently converted into dense vector representations through a sentence, level embedding model and stored in an in, memory FAISS index to allow for fast similarity, based retrieval and queries. At the moment of the query, the system analyzes the user's input to determine the cognitive state and dynamically decides retrieval depth.

After that, the relevant chunks of the documents retrieved from the FAISS index are mixed with the query to create a context, aware prompt. This prompt is then given to the locally hosted language model to produce a response.

The generated texts are slowly sent back to the frontend together with the source references. This enables low, latency interaction, transparency, and the ability to trace back the original document content.

IV. EXPERIMENTAL DETAILS

CS, RAG is a document, driven system whereby documents are uploaded dynamically instead of having a fixed data training set. This design allows for a system to switch content domains at will without retraining or fine, tuning.

During a session users can upload several documents and the system can cross, documentation summarization and comparative understanding of the documents.

After a document is uploaded, the files are verified and turned into plain text with the help of regularized extraction pipelines. The extracted text is then split into overlapping segments so that semantic coherence can be preserved even at the edges. Overlapping is most crucial in the case of technical documentation where many concepts are often explained over several paragraphs

Each piece is tagged with metadata like document identifier, chunk index, and positional reference. This data not only aids in retrieval but also in source attribution at the frontend.

Considering uploaded documents as knowledge bases limited to sessions, Cs, RAG, thus, guarantees user session isolation and, therefore, prevents unintended information leakage.

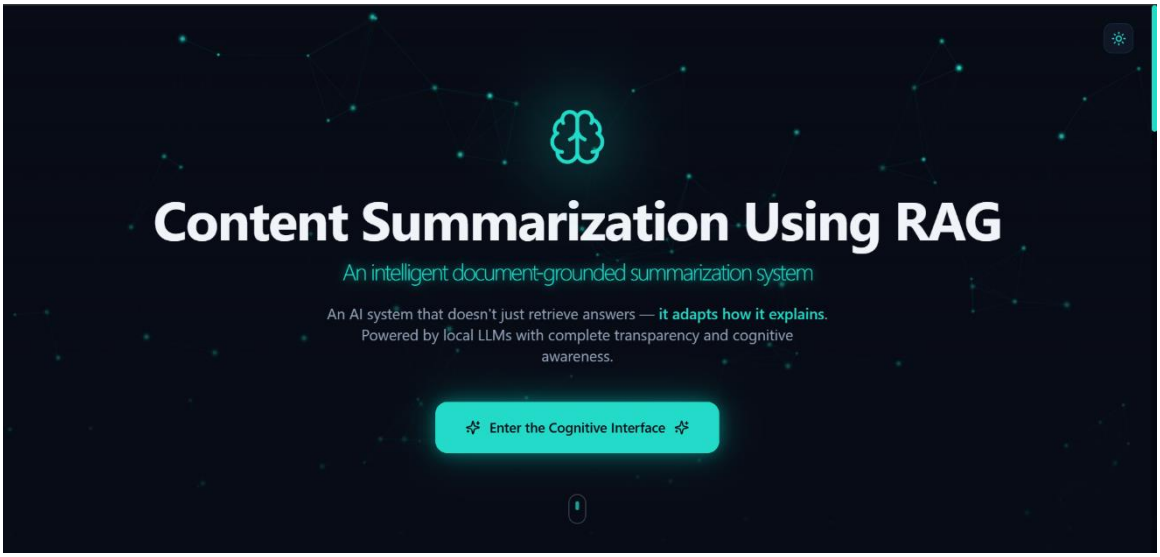
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

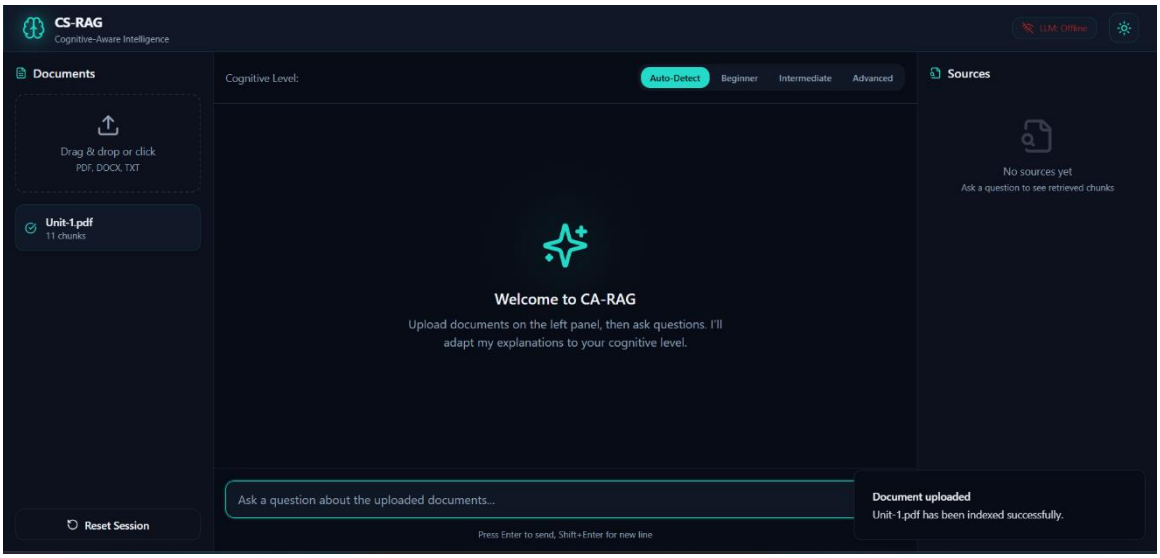
VI. Results and Discussion

The system was tested through different controlled usage trials which assessed variables such as response relevance, retrieval behaviour, latency, and user interaction characteristics. The users uploaded documents of varying lengths ranging from 3, 000 to 18, 000 words which were then processed through a FAISS, based in, memory retrieval pipeline. Queries simulated different expertise levels, and the adaptive retrieval approach always succeeded in determining the right number of document chunks.

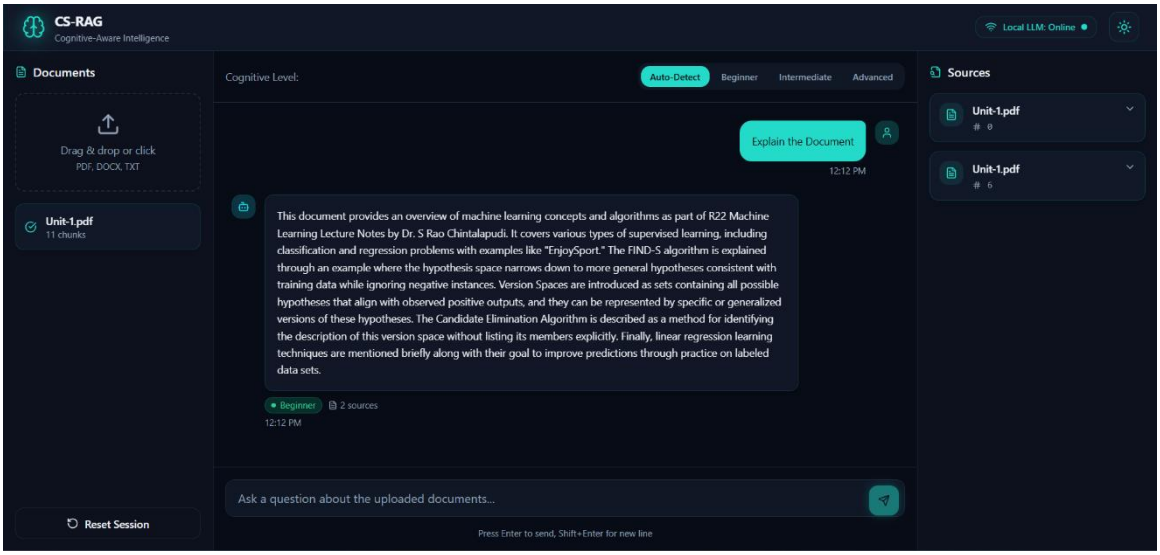
Overall latency to reach the system was 1.2 seconds for documents of average size and 1.6 seconds for bigger ones, while embedding and indexing costs scaled linearly and thus did not interfere with further interactions. Qualitative assessment showed that the alignment of query intent and response detail was better compared to fixed, depth retrieval, overlapping chunks helped in keeping the semantic continuity of the document sections. Source attribution provided users another level of transparency by allowing users to trace the basis of the generated responses.

Even though the heuristic, based cognitive estimation, and single, document evaluation method have certain limitations, the results demonstrate that adaptive retrieval improves the overall user experience and relevance without significantly increasing the expenditure of computationalresources.





International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026



VII.CONCLUSION

The paper presents a document summarization and question answering system that uses Retrieval, Augmented Generation(RAG) as a base and combines it with adaptive retrieval controlled by cognitive state estimation. The system tackles the issue of fixed context RAG pipelines by adjusting the retrieval depth according to the estimated query complexity, thus enabling it to give users with varied levels of expertise the right type of responses.In the same architecture, the authors merged document ingestion, semantic chunking, dense vector retrieval, heuristic, based cognitive estimation, and real, time language model generation so that their system is capable of generating context, aware answers at an impressive response time. The experiment results demonstrate that adaptive retrieval facilitates a better matching of the user's intent and response detail without a significant increase in computational expenses, while source attribution makes the content more understandable by showing the pieces of documents that have been utilized in the generation of the answer. system uses heuristic cognitive estimation and thus it can only support single, document sessions at the moment, which could be a limitation in generalizing this solution to different domains and modes of interaction. So, the team plans to explore the extension of the framework to multi, document retrieval, introducing learned or personalized cognitive models and enabling persistent storage for long, term use in their next work.

REFERENCES

[1]. 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," *NeurIPS*, 2020.
[2]. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," arXiv preprint arXiv:1908.10084, 2019. (Sentence-BERT / SBERT foundation for sentence-level embeddings).
[3]. "all-MiniLM-L6-v2 — sentence-transformers model card," HuggingFace model repository (model card and usage notes for the 384-dimensional MiniLM embedding).
[4]. J. Johnson, M. Douze and H. Jégou, "Billion-scale similarity search with GPUs," arXiv preprint arXiv:1702.08734, 2017 — foundational paper on large-scale vector search techniques that motivated FAISS and GPU acceleration.
[5]. Facebook AI Research, "FAISS — A library for efficient similarity search," GitHub repository and documentation. (Practical reference for IndexFlatIP and other FAISS indices used in CS-RAG).
[6]. N. Reimers et al., "Sentence Transformers — library documentation and API," Sentence-Transformers project site (practical usage, model list and embedding guidance).
[7]. pdfplumber — Python PDF extraction library (project page / PyPI / documentation). (Used in the ingestion pipeline for PDF text extraction).
[8]. python-docx — Python library documentation (ReadTheDocs / GitHub) for reading DOCX files programmatically. (Used for Word document ingestion in CS-RAG).
[9]. Ollama documentation and API reference — local model runtime and inference API (used to host and stream responses from locally managed LLMs).
[10]. Token/streaming generation and UX resources:
- AWS blog: "Stream large language model responses in Amazon SageMaker JumpStart" (practical guide to token streaming).
- Research on robust LLM token streaming under unstable networks (preprints / frameworks discussing streaming reliability and transport).
[11]. Project/report template and internal documentation (uploaded): "University_of_Southampton_Part_III_Project_Report_Template.pdf" — contains system description, architecture and methodology used in CS-RAG implementation.
[12]. University_of_Southampton_Part_…
[13]. S. Robertson and H. Zaragoza, "The Probabilistic Relevance Framework: BM25 and Beyond," *Foundations and Trends in Information Retrieval*, vol.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 06 , June - 2026

3, no. 4, pp. 333–389, 2009.
→ Classic reference for retrieval relevance modeling, useful for grounding similarity-based retrieval discussion.
[14]. K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators," *ICLR*, 2020.
→ Cited as representative of efficient transformer encoders influencing lightweight embedding models.
[15]. V. Karpukhin, B. Oğuz, S. Min et al., "Dense Passage Retrieval for Open-Domain Question Answering," *EMNLP*, 2020.
→ Foundational work on dense retrieval architectures widely adopted in document-grounded QA systems.
[16]. J. Lin, "Is Question Answering Better than Information Retrieval? Toward a Task-Based Evaluation Framework," *SIGIR*, 2019.
→ Supports discussion on limitations of retrieval-only systems and motivation for RAG-style hybrids.