Open Access International Journal of Science and Engineering

ISSN: 2456-3293

Volume 9 Issue 1s (2026) | Pages: 258-260

Dynamic Multilingual Retrieval-Augmented Generation (RAG) System for Table-Heavy PDFs

Mahek Pewekar[1], Sonal Jamdade[2], Sudarshana Abbad[3]

[1,2,3]Dept. of Computer Science and Engineering Bharati Vidyapeeth Deemed University College of Engineering Pune, India

Peer Review Information

Type: *Article*
Received: *04 April 2026*
Revised: *30 April 2026*
Accepted: *05 June 2026*
Published: *26 June 2026*

Abstract

Baseline Retrieval-Augmented Generation (RAG) systems show great promise in helping AI generate relevant and grounded responses while reducing hallucinations and lowering computational costs [1][2]. By merging information retrieval with large language models (LLMs), RAG frameworks let models access external knowledge sources, which improves factual accuracy and response reliability. However, baseline RAG systems mainly work with unstructured sequential text and struggle with structured data formats, especially tables [3][4]. Tabular data often has complex relationships between rows, columns, headers, and numerical values. Conventional text-based retrieval methods do not capture these relationships well.

In real-world situations, many documents, like research papers, financial reports, and enterprise PDFs, mix unstructured text with structured elements such as tables, charts, and multi-level headers. Traditional RAG pipelines often fail to keep the two-dimensional structure of tables during preprocessing. This leads to fragmented or misleading retrieval results. Such structural issues can harm downstream generation, causing incomplete or inaccurate responses, especially when numerical or table-related information is involved.

To tackle these challenges, this study investigates a better way to retrieve complex multi-format data in a format friendly to LLMs [5][6]. The suggested method emphasizes structure-aware preprocessing and semantic chunking strategies that maintain contextual and structural relationships in both text and tables. By creating semantically meaningful chunks with controlled overlap, the approach ensures that essential contextual information stays intact during retrieval.

Additionally, the system supports table-heavy documents written in multiple Latin-based languages by using multilingual embedding models. This allows for both same-language and cross-language retrieval, enabling users to query documents in different languages without losing semantic accuracy. Overall, the proposed method improves the performance of RAG systems for complex, real-world document collections by enhancing retrieval accuracy, contextual grounding, and usability across various document formats.

Keywords: Retrieval-Augmented Generation; Large Language Models; Semantic Chunking; Structured and Unstructured Data; Vector Embeddings; Information Retrieval.

How to Cite This Article

Pewekar, M., Jamdade, S., & Abbad, S. (2026). Dynamic multilingual retrieval-augmented generation (RAG) system for table-heavy PDFs. *Open Access International Journal of Science and Engineering*, 9(1s), 258–260.

Introduction

In daily life, documents consist of numerous data formats along with text [3], which pose a challenge to traditional retrieval systems as they mainly take inputs in the form of text. Data retrieval in table-heavy documents poses a challenge to baseline systems, as it struggles to read structured data. LLMs provide strong language understanding but are prone to hallucinations without document grounding [2][12]. In this study we will explore an approach that retrieves structured tables in an LLM- friendly format. Chunks are formed semantically and some overlap is maintained to preserve context [6]. These chunks are embedded, then the relevant top-k chunks are semantically retrieved and reranked for optimal results. This approach helps retrieve unstructured and structured data across multiple Latin languages [7][8].

Method

Traditional RAG systems are mainly operating on unstructured sequential text and fail to retrieve tables in a machine understandable form. The two-dimensional layout [3][4] of a table is disrupted due to the failure of identifying tabular ques. We propose an alternative approach to retrieve tables in an LLM-friendly format [1][10].

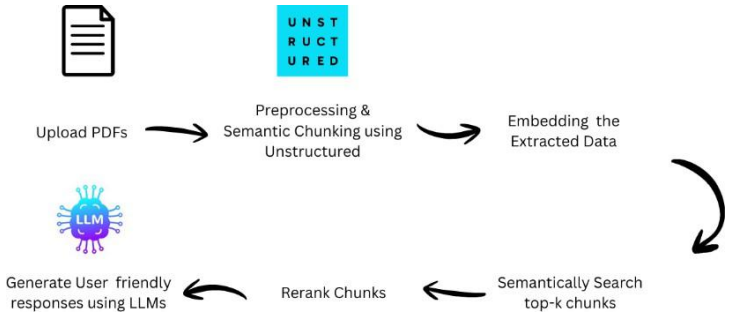

Fig. 1. *Architecture of Model*

Preprocessing & Semantic Chunking using Unstructured

The first step, involves uploading the PDF, this can be achieved through uploading the file beforehand or through inputting the download link of the respective PDF. The PDF document, consisting of tables and multilingual text, is parsed using the Unstructured library [5]. Different elements are extracted and stored separately, in a sequential manner. Chunks are formed semantically by identifying headings or subheadings. An overlap is introduced as well to retain context [6].

Retrieval

The text and table chunks are embedded using a multilingual embedding model, that transforms the data into dense vector representations [7][8]. These multilingual embedding models support same-language and cross-language queries. These vector representations are stored in a vector store for further use. The user query is embedded using the same model and cosine similarity is utilized to retrieve top-k relevant chunks. Cosine similarity is tasked with finding the top-k semantically similar chunks to the user query. The top-k chunks are further reranked to identify the most contextually accurate chunks with respect to the user query [9]. Reranking is extremely effective when numerical values and table-centric data is involved [10].

Generate User-Friendly Responses using LLMs

The top-k chunks are given to a LLM to generate user-friendly responses. The number of chunks given to the LLM plays a crucial role, an optimal number of chunks should be given to produce the best results. If the quantity of chunks is too high or low, it will affect the output negatively [6][11]. The top-k chunks are then passed onto the LLM along with a base prompt to ensure grounded responses [1][2][12].

Experiments

Set up

1. Tables of varying formats, containing financial statements, multi-level headers, footnotes, merged cells, and irregular layouts, were ingested into the system. The processed chunks were then fed to larger and advanced LLM to test how well this approach performed [3][4].
2. Multiple PDF documents written in various Latin-based languages were passed through the RAG system. Cross-language and same-language retrievals were performed. The responses were evaluated by a larger LLM on the following metrics Semantic Accuracy, Handling of Cultural Context, Emotional & Literary Understanding, Faithfulness to Source (No Hallucination), Clarity & Readability, and LLM-Friendliness (Context Use) [2][11][12].

Results

Table 1. *Performance Evaluation of the Proposed RAG Framework*

Criterion	Score
Table Structure Preservation	9/10
Numerical Reasoning Readiness	8.5/10
LLM Grounding	9/10
Retrieval Quality	8.5/10
Overall LLM-Friendliness	8.5/10

Table 2. *Qualitative Evaluation of the Proposed Multilingual RAG System*

Criterion	Score (out of 10)
Semantic Accuracy	9.5
Handling of Cultural Context	9.0
Emotional & Literary Understanding	9.5
Faithfulness to Source (No Hallucination)	10.0
Clarity & Readability	9.0
LLM-Friendliness (Context Use)	9.5

Conclusion

In this work, we explored an enhanced Retrieval-Augmented Generation (RAG) framework capable of working on table-heavy multilingual PDF documents. This approach helps access information from complex documents, like research papers and enterprise documents. The data is retriened from the documents in an LLM-friendly format and the output is user-friendly. In summary, the system enhances usability, accuracy, and reliability of response, making it ideal for practical document question-answering tasks.

Acknowledgement

The authors are very much thankful to management for providing us facilities to conduct the experiments in the institution's laboratory and colleagues for their support to prepare this paper. We are also thankful to the Science & Technology department for approving our research proposal and providing us grants to conduct research activities.

References

1. P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 9459–9474, 2020.
2. K. Shuster et al., "Retrieval Augmentation Reduces Hallucination in Conversation," *Findings of the Association for Computational Linguistics: EMNLP*, pp. 3784–3803, 2021.
3. J. Herzig et al., "TaPas: Weakly Supervised Table Parsing via Pre-training," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 4320–4333, 2020.
4. Z. Zhang et al., "Table Question Answering with Pretrained Models," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.
5. Unstructured Technologies Inc., "Unstructured: Open-Source Document Parsing Framework," 2023.
6. N. F. Liu et al., "Lost in the Middle: How Language Models Use Long Contexts," *Transactions of the Association for Computational Linguistics (TACL)*, 2023.
7. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.
8. F. Feng et al., "Language-Agnostic BERT Sentence Embedding," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.
9. J. Johnson et al., "Billion-Scale Similarity Search with FAISS," *IEEE Transactions on Big Data*, 2019.
10. R. Nogueira and K. Cho, "Passage Re-ranking with BERT," *arXiv preprint*, arXiv:1901.04085, 2019.
11. OpenAI, "GPT-4 Technical Report," 2023.
12. Z. Ji et al., "Survey of Hallucination in Natural Language Generation," *ACM Computing Surveys*, 2023.