Evaluating RAG Strategies in a Modular LLM Architecture

Vinicius Aguiar Alboneti[1], **Leonardo Afonso Amorim**[1],
Jonatas Tomazini[1], **Ricardo Costa**[1]
Sávio Salvarino Teles de Oliveira[1], **Arlindo Rodrigues Galvão Filho**[1],
Anderson da Silva Soares [1]
Tales Brumon Medeiros de Figueiredo[2]

[1] Instituto de Informática – Universidade Federal de Goiás (UFG),
Caixa Postal 131 – 74.001-970 – Goiânia – GO – Brazil

vinicius_aguiar2@discente.ufg.br,leonardoafonsoamorim@egresso.ufg.br

tomazini@discente.ufg.br,ricardodouglasrodrigues@protonmail.com

savioteles@ufg.br,arlindogalvao@ufg.br

andersonsoares@ufg.br

tales.figueiredo@cemig.com.br

[2]CEMIG, Companhia Energética de Minas Gerais
Avenida Barbacena, 1200, Santo Agostinho, Belo Horizonte, 30190-131 – Brazil

Abstract. *This paper presents an empirical evaluation of three Retrieval-Augmented Generation (RAG) patterns — Naive RAG, RAG-Fusion, and Self-RAG — applied in a real-world corporate environment in the energy sector. Using a modular and consolidated architecture built on managed services and widely adopted technologies (such as pgvector, Cloud Run, and LLM APIs), we conducted experiments with actual data from an energy company, focusing on regulatory audit processes. We compared RAG strategies using automated evaluation metrics (RAGAS), considering faithfulness, contextual precision, answer relevance, and response time. The results show that the Self-RAG pattern achieves the best balance between response quality and performance, making it the most suitable for enterprise applications that require accuracy, efficiency, and scalability. This practical validation offers guidance on adopting RAG in corporate environments, highlighting the trade-offs associated with selecting different approaches.*

1. Introduction

The growing demand for intelligent solutions in the energy sector has driven the adoption of Artificial Intelligence (AI) models focused on optimizing data analysis and decision support. Large companies in this field operate across multiple directorates, departments, and technical areas, each with specific information access requirements, which calls for mechanisms capable of delivering relevant content in a personalized manner [Raihan 2023].

In recent years, the advancement of Large Language Models (LLMs) has significantly impacted the field of Natural Language Processing (NLP), enabling the development of agents capable of generating highly contextualized and accurate responses. However, the practical application of these models in enterprise settings still faces challenges related to scalability, context handling, and the quality of generated outputs [Zhao et al. 2025].

To address these limitations, Retrieval-Augmented Generation (RAG) strategies have emerged as a promising approach by incorporating external retrieval mechanisms into the generative process [Lewis et al. 2020]. Several RAG patterns have been proposed, including Naive RAG, RAG-Fusion [Rackauckas 2024], and Self-RAG [Asai et al. 2024], each offering different trade-offs between latency, precision, and contextual robustness.

This paper presents a comparative empirical evaluation of these three RAG patterns in a real-world corporate environment within the energy sector. Given the sector's inherent complexities, characterized by stringent regulatory demands and the critical need for precise, reliable information across diverse operational areas, it becomes imperative to thoroughly assess how different RAG architectures perform. We conducted the experiments using real data related to regulatory audit processes, and the analysis relied on automated evaluation metrics (RAGAS) [Es et al. 2024], focusing on factual accuracy, contextual precision, answer quality, and response time.

This work's core contribution lies in the practical validation of RAG strategies in a corporate setting, offering insights into the trade-offs involved in their adoption. By analyzing the performance and trade-offs of each RAG pattern, this study provides actionable guidance for selecting the most suitable approaches for enterprise applications in the energy sector.

This paper is structured as follows. Section 2 introduces key concepts related to Retrieval-Augmented Generation (RAG). Section 3 reviews related work and contextualizes this research. Section 4 presents the modular evaluation architecture. Section 5 discusses the experimental setup and results. Finally, Section 6 summarizes the findings and suggests future directions.

2. Background

Large Language Models (LLMs) have transformed Natural Language Processing (NLP), enabling improvements in tasks such as text generation, summarization, and conversational AI. However, the practical deployment of LLMs presents challenges related to scalability, response accuracy, and efficient context handling [Wang and Iida 2024].

2.1. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) enhances Large Language Model (LLM) capabilities by injecting external knowledge into the generation process. Instead of relying solely on training data, RAG first retrieves relevant content from a knowledge base and then conditions the LLM's output on that context [Lewis et al. 2020]. It improves factual accuracy and mitigates hallucinations. RAG typically involves three steps: transforming a user query into an embedding, retrieving semantically similar documents from a vector store, and generating a final response based on the retrieved context.

A practical example in the energy sector involves regulatory audits. For instance, to verify compliance of an electrical substation with current regulations, a user submits a query, which is converted into a vector representation. Relevant documents are retrieved from a regulatory database, and the LLM utilizes this context to generate a precise response, thereby reducing manual workload and enhancing reliability.

2.2. RAG Patterns Evaluated

This work evaluates three Retrieval-Augmented Generation (RAG) strategies, each offering distinct mechanisms for balancing response quality and computational efficiency. The first, referred to as Naive RAG, employs a straightforward retrieve-and-generate methodology without any refinement or re-ranking processes. While it ensures low latency, it is more susceptible to retrieving irrelevant or low-quality information. The second strategy, RAG-Fusion, enhances retrieval robustness and contextual diversity by issuing multiple reformulated queries and aggregating results using the Reciprocal Rank Fusion (RRF) method [Rackauckas 2024].

Although this approach increases the richness of the retrieved content, it also leads to longer response times due to additional computational overhead. The third strategy, Self-RAG, incorporates a self-reflection mechanism that dynamically evaluates the need for further information retrieval and critiques its outputs. This results in improved factual accuracy and contextual precision, albeit with a moderate increase in latency [Asai et al. 2024].

These strategies reflect inherent trade-offs among accuracy, contextual depth, and computational cost. The objective of our experiments is to empirically compare their performance in real-world business scenarios, thereby identifying the most effective approach for enterprise-grade natural language applications.

3. Related Work

In the domain of energy sector, Qiao et al. [Qiao et al. 2024] addressed the challenges posed by the vast and complex regulations within the energy industry by proposing an intelligent question answering system. This system is built upon the RAG architecture. It allows to transform regulatory documents into a vector database and employs a multi-channel recall and fusion ranking strategy to enhance retrieval accuracy. Their work demonstrates improvements in information retrieval efficiency and response generation, achieving high performance metrics in experimental evaluations.

Medeiros et al. [Medeiros et al. 2024] proposed a Q&A system for contract management combining GPT-4, RAG, and Prompt Engineering. The study demonstrated improved relevance without retraining LLMs, highlighting the benefits of RAG in corporate document workflows. Nonetheless, no systematic comparison of different RAG patterns was presented.

Albuquerque et al. [Albuquerque et al. 2024] introduced an implicit feedback mechanism to evaluate RAG-based applications through user interactions. It contributes to continuous system improvement, but does not cover comparative analysis between retrieval strategies or response quality under different configurations.

While significant progress has been made in understanding RAG frameworks, with Zhao et al. [Zhao et al. 2024] offering a comprehensive survey on how Large Lan-

guage Models (LLMs) can effectively utilize external data, there is a notable absence of comparative studies focusing specifically on different RAG strategies within the energy sector. Despite extensive research into RAG architectures and their applications across various domains, a comprehensive evaluation and comparison of distinct RAG approaches tailored for, or rigorously evaluated within, the unique context of the energy sector is yet to be widely explored in the literature.

4. Experimental Platform

We conducted the experiments presented in this study using a modular and cloud-based architecture designed to support the integration of Large Language Models (LLMs) with Retrieval-Augmented Generation (RAG) techniques. Rather than proposing a new architecture, this environment leverages widely adopted tools and practices to provide a reliable foundation for testing different RAG strategies in a realistic corporate setting, as shown in Figure 1.

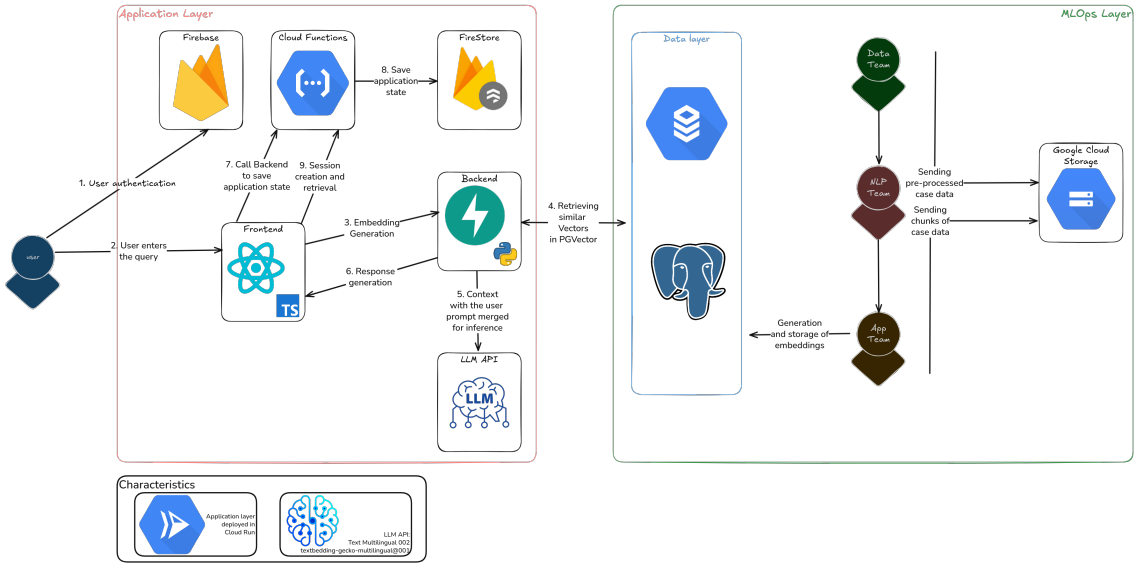

Figure 1. Modular NLP Platform Architecture for RAG Experiments

The architecture consists of three main components: data ingestion and embedding generation, retrieval and context management, and LLM-based response generation. It integrates a vector database for similarity search, a RESTful API layer for managing user queries, and a cloud-hosted frontend for user interaction. We deployed all components using managed services to ensure scalability, low latency, and ease of integration.

The workflow begins with the ingestion of textual data from corporate documents. These are preprocessed and converted into semantic embeddings, which are stored in a vector-enabled relational database. When a user submits a query through the frontend interface, the system generates an embedding for the query, retrieves the most similar content from the database, and sends the combined input to a pre-trained LLM.

The generated response is then returned to the user. Throughout the process, the system ensures session management and persistent logging to support reproducibility and analysis. This environment allowed for the controlled and consistent evaluation of

the Naive RAG, RAG-Fusion, and Self-RAG patterns using the same infrastructure and datasets.

By relying on established technologies and modular design principles, this platform provided a robust and reproducible foundation for comparing RAG approaches in a real-world enterprise scenario, thereby minimizing variability introduced by architectural differences.

5. RAG Strategy Evaluation

This section presents the empirical evaluation of three RAG strategies—Naive RAG, RAG-Fusion, and Self-RAG—using a consistent testing environment and dataset[1]. The objective was to compare their performance in a realistic enterprise scenario, focusing on both answer quality and system efficiency.

5.1. Experimental Setup

We used a dataset composed of 50 synthetic question-answer pairs derived from a corpus of regulatory audit documents in the Brazilian energy sector. We constructed a corpus from 200 documents sourced from the Internal Audit department of an electric utility company, with a focus on regulatory and normative aspects of the power industry. The collection comprises legislation, regulatory guidelines, technical manuals, and procedural documents, organized thematically into areas such as power generation, concessions, distribution, tariff regulation, transmission, energy efficiency, commercialization, and sector-specific accounting and asset management.

The evaluation queries were synthesized by domain experts at the utility company based on real information needs, ensuring strong alignment with practical use cases. The data curation process prioritized institutionally sourced textual documents of high regulatory relevance, capturing realistic information retrieval scenarios for technical and compliance-oriented decision-making. We described the infrastructure in Section 1 used to ensure uniformity across tests. For embeddings, we used the `text-multilingual-embedding-002` model, and for generation, the `Gemini-1.5-Pro` model. We conducted an evaluation using the RAGAS framework [Es et al. 2024].

5.2. Evaluation Metrics

We conducted the comparison using a set of automated evaluation metrics designed to assess both the quality and efficiency of the responses. Faithfulness measures the degree to which the generated response accurately reflects the retrieved context. Context Recall evaluates the completeness of relevant contextual information obtained during the retrieval phase. LLM Precision quantifies the extent to which the retrieved content is meaningfully incorporated into the final response. Answer Relevancy assesses the semantic alignment between the generated response and the original user query. Lastly, Response Time captures the average latency observed in producing a final response.

Table 1. Performance comparison of RAG patterns

Pattern	Faithfulness	Context Recall	LLM Precision	Answer Relevancy	Response Time (s)
Naive RAG	0.7787	0.8883	0.6354	0.6486	**1.6808**
RAG-Fusion	0.8340	**0.9367**	0.2971	**0.6887**	21.8566
Self-RAG	**0.8408**	0.9283	**0.6568**	0.6766	8.9790

5.3. Results and Analysis

Naive RAG showed the fastest response time (1.68s), making it attractive for latency-sensitive scenarios. However, it also produced the least accurate and contextually aligned answers. Its simplicity (single-query retrieval) results in lower faithfulness and answer relevancy, limiting its use in applications that require high precision.

RAG-Fusion achieved the highest context recall (0.9367) and answer relevancy (0.6887), reflecting its ability to enrich the prompt using multiple reformulations. However, its average response time exceeded 21 seconds due to the cost of re-ranking and aggregating multiple results. It makes it less practical for real-time systems.

Self-RAG demonstrated the best overall balance. It achieved the highest scores in faithfulness (0.8408) and LLM precision (0.6568), with moderate response time (8.97s). The self-reflection mechanism enabled it to determine when retrieval was necessary, thereby improving factual consistency and contextual focus.

5.4. Discussion

The comparison highlights a clear trade-off between response quality and latency among the evaluated RAG strategies. The Naive RAG approach demonstrates lower latency but limited precision, making it more appropriate for non-critical or exploratory tasks. In contrast, RAG-Fusion retrieves more comprehensive and contextually rich information, albeit at the cost of higher computational demands, which makes it more suitable for offline analyses or scenarios where accuracy is paramount. Self-RAG presents the most balanced performance, offering a favorable compromise between precision and responsiveness, thereby making it particularly well-suited for enterprise applications that demand both contextual reliability and real-time interaction. These results suggest that Self-RAG is the most suitable strategy for corporate natural language processing systems, particularly in contexts where both contextual accuracy and low latency are crucial. In future work, we intend to expand the evaluation dataset, enhance the precision of latency measurements, and explore hybrid approaches to optimize performance further.

Another relevant aspect concerns the operational cost and scalability of RAG architectures in production. While Naive RAG offers a minimal computational footprint, RAG-Fusion incurs higher costs due to multiple retrieval queries and ranking stages. Self-RAG, although moderately more expensive, presents a cost-benefit equilibrium that makes it feasible for real-time applications. From a governance perspective, integrating observability tools—such as prompt logs, query analytics, and embedding drift detection—enhances traceability and supports continuous optimization of RAG pipelines.

[1]The implementation of the proposed patterns is available at: `https://github.com/leonardoamorim/cemig-rag.git`

Additionally, the analysis of RAGAS metrics reveals that improvements in faithfulness and contextual precision often come at the expense of response time. This trade-off underscores the importance of adaptive RAG configurations, where retrieval depth and model temperature can be dynamically adjusted based on user intent or system load. Future developments should also consider hybrid caching strategies, combining static document indexing with dynamic knowledge updates, to sustain low latency while preserving factual accuracy. These findings extend the understanding of how modular LLM architectures can evolve toward self-optimizing, enterprise-ready conversational systems.

6. Conclusion

This study evaluates three Retrieval-Augmented Generation (RAG) strategies—Naive RAG, RAG-Fusion, and Self-RAG—within a modular, cloud-based large language model (LLM) platform, applied to a real-world energy sector scenario. Each strategy presents trade-offs: Naive RAG is efficient but less accurate; RAG-Fusion improves retrieval quality at the cost of latency; Self-RAG achieves the best balance between precision, context, and response time. These results support the selection of strategies based on business goals and infrastructure constraints. Instead of proposing a new architecture, the study leverages a stable and reproducible setup to isolate the behavior of each RAG method under consistent conditions. We used real-world data and automated metrics to ensure reliability and relevance. Future work will explore broader domains and datasets, test hybrid or adaptive RAG methods, and implement performance optimizations such as caching, batching, and cost-aware retrieval - enhancing RAG's utility in scalable, real-time NLP applications.

References

Albuquerque, A., Wensing, I., Filho, N. J., and Dorneles, C. (2024). Avaliação de aplicações de geração aumentada de recuperação por meio de feedback implícito. In *Anais Estendidos do XXXIX Simpósio Brasileiro de Bancos de Dados*, pages 253–259, Porto Alegre, RS, Brasil. SBC.

Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. (2024). Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint*.

Es, S., James, J., Espinosa Anke, L., and Schockaert, S. (2024). RAGAs: Automated evaluation of retrieval augmented generation. In Aletras, N. and De Clercq, O., editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158, St. Julians, Malta. Association for Computational Linguistics.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Medeiros, A., Cavalcante, C., Nepomuceno, J., Lago, L., Ruberg, N., and Lifschitz, S. (2024). Contrato360: uma aplicação de perguntas e respostas usando modelos de linguagem, documentos e bancos de dados. In *Anais do XXXIX Simpósio Brasileiro de Bancos de Dados*, pages 155–166, Porto Alegre, RS, Brasil. SBC.

Qiao, Z., Gao, M., Ma, H., Wen, X., Yan, E., Zhao, W., Cao, H., and Wu, H. (2024). Intelligent question answering system for power regulations based on rag. In *2024 4th International Conference on New Energy and Power Engineering (ICNEPE)*, pages 1155–1161. IEEE.

Rackauckas, Z. (2024). Rag-fusion: a new take on retrieval-augmented generation. *arXiv preprint*.

Raihan, A. (2023). A comprehensive review of artificial intelligence and machine learning applications in the energy sector. *Journal of Technology Innovations and Energy*, 2:1–26.

Wang, H. and Iida, F. (2024). Emotional alignment for human-robot cooperation in musical tasks. *IOP Conference Series: Materials Science and Engineering*, 1321(1):012006.

Zhao, S., Yang, Y., Wang, Z., He, Z., Qiu, L. K., and Qiu, L. (2024). Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2025). A survey of large language models.