Evaluating Domain-Specialized LLMs in Multi-Agent RAG for Enterprise Retrieval

Vinícius A. Aguiar[1], Leonardo A. Amorim[1], Artur M. A. Novais[1], Gustavo L. B. Pereira[1], Tales de Oliveira[2], Carlos A. de Souza[2], Arlindo R. G. Filho[1], Anderson S. Soares[1], Sávio S. T. de Oliveira[1]

[1] Instituto de Informática – Universidade Federal de Goiás (UFG)

{vinicius_aguiar2, artur.matos, gustavobueno}@discente.ufg.br

leonardoafonsoamorim@egresso.ufg.br

{arlindogalvao, andersonsoares, savioteles}@ufg.br

[2]CEMIG, Companhia Energética de Minas Gerais

{tales.figueiredo, casal}@cemig.com.br

Abstract. *This paper evaluates a multi-agent architecture for enterprise knowledge retrieval where a semantic router directs queries to specialized LLM agents covering thematic domains (e.g., legal, regulatory). We benchmarked prominent models, including GPT-4, LLaMA 4, and Gemini, on metrics like answer relevance, faithfulness, and execution time. Our results demonstrate that this specialized approach achieves superior precision and lower latency compared to centralized configurations. We identify GPT-4o, LLaMA 4, and Gemini-2.5-Flash as offering the best balance of accuracy and efficiency. These findings provide a practical guide for designing scalable, high-fidelity retrieval systems for regulated, multi-domain environments.*

1. Introduction

The integration of Large Language Models (LLMs) into corporate environments has transformed how organizations retrieve and interact with internal knowledge. However, challenges such as information governance, factual consistency, and domain specialization remain critical, especially in regulated sectors that demand accurate and auditable responses.

Retrieval-Augmented Generation (RAG) addresses these issues by combining semantic search with generative models, grounding outputs in relevant knowledge bases [Lewis et al. 2020b]. While widely adopted, centralized RAG pipelines often lack scalability, modularity, and fine-grained access control [Chang et al. 2024].

To overcome these limitations, multi-agent architectures delegate queries to domain-specialized agents coordinated by a semantic router. Each agent maintains its own vector store and LLM, enabling contextual precision, modular scalability, and policy-compliant governance.

This work evaluates such an architecture in the complex energy sector, which presents dense technical language, evolving regulations, and varied domains including

Audit, Legal, R&D, Regulatory, Investor Relations, and Sustainability. We benchmark leading LLMs on answer relevancy, faithfulness, execution time, and routing accuracy, providing a practical comparison of trade-offs in quality, accuracy, and latency. Our results offer guidance for designing scalable, trustworthy, and governance-aware enterprise retrieval systems.

2. Background and Related Works

Recent advancements in natural language processing have led to the widespread adoption of LLMs across various domains. This section provides an overview of the foundational concepts underlying our work, including RAG mechanisms, the role of LLMs in multi-agent systems, and evaluation strategies.

2.1. Retrieval-Augmented Generation (RAG)

Despite their impressive capabilities, LLMs face notable limitations in enterprise applications. One of the challenges is the lack of grounding in factual, up-to-date, and context-specific information. In response to these challenges, the community has increasingly explored hybrid retrieval-generation strategies, such as Retrieval-Augmented Generation (RAG), that combine the fluency of LLMs with the factual accuracy of external knowledge bases [Lewis et al. 2020a].

RAG is a technique designed to enhance the reasoning and response capabilities of LLMs by incorporating external sources of information during inference. Rather than depending exclusively on pre-trained knowledge, RAG first retrieves semantically relevant data from a knowledge base, which is then used to guide and constrain the model's generation process [Lewis et al. 2020b]. This approach helps reduce hallucinations and increases factual grounding. A typical RAG pipeline consists of three stages: encoding the user query as an embedding, retrieving the top-matching documents from a vector index, and generating a response conditioned on the retrieved content.

2.2. Multi-Agent Architectures for Knowledge Access

Multi-agent systems (MAS) comprise multiple autonomous agents that interact with each other and their environment to achieve individual or shared goals. By leveraging distributed reasoning and communication, MAS enables the decomposition of complex tasks into specialized roles, facilitating scalable and adaptive solutions in domains such as logistics, industrial automation, healthcare, and data governance [Wooldridge 2009].

In the context of knowledge access, MAS offers an architectural paradigm well-suited to handling domain-specific segmentation, policy-based access control, and modular retrieval strategies. Each agent can act as a specialist responsible for a particular domain or function. The integration of LLMs into these architectures improves not only human-agent interaction but also inter-agent collaboration through shared semantic representations [Park et al. 2023].

However, enabling effective knowledge access in multi-agent LLM environments presents challenges, such as controlling hallucination, maintaining up-to-date knowledge, and balancing performance with factual consistency. To address these issues, recent works have proposed hybrid strategies that allow agents to dynamically ground their outputs in domain-specific evidence, supporting modularity, scalability, and accuracy in complex enterprise scenarios [Shinn et al. 2023].

2.3. Related Works

The use of Multi-Agent Systems (MAS) in distributed environments has long been explored. Foundational studies such as [Wooldridge 2009] highlight key MAS principles—autonomy, cooperation, and communication—as effective strategies for decentralized and adaptive systems. While MAS has been applied to cognitive architectures, recommendation systems, and industrial automation, its use in domain-segmented, access-controlled knowledge retrieval is still limited. RAG was introduced by [Lewis et al. 2020b] and it marks a breakthrough in combining vector databases with generative models. Enhancements like domain-specific fine-tuning, contextual memory, and semantic pipelines followed. However, most implementations remain centralized and monolithic, limiting scalability and policy enforcement.

Recent work has proposed modular alternatives. Agentic Mesh [Broda 2025] and topic-based pipelines [Puschmann et al. 2022] support orchestration and modularity, though they do not explicitly address federated governance. Architectures such as [Team 2025] and [Dulay 2024] introduce collaborative and event-driven multi-agent designs. Tools like LLM Agentic Tool Mesh [Enterprise 2024] abstract orchestration for LLM-based agents and services.

Han et al. [Han et al. 2024] present a detailed review of MAS+LLM architectures, identifying challenges such as inter-agent coordination, memory sharing, and consistent context handling. Our work aligns with their recommendations, offering empirical validation for modular and role-specific agent setups with domain isolation.

Costa et al. [da Costa and e Souza Filho 2024] compare fine-tuning and RAG strategies for adapting LLMs to Portuguese domain-specific QA tasks. Their results highlight RAG as a lightweight and adaptable alternative, especially in multilingual or low-resource settings, reinforcing its relevance for specialized applications and complementing our architecture-oriented approach.

Siqueira et al. [Siqueira et al. 2024] explore the use of structured data inputs to enhance chatbot integration in enterprises, emphasizing the importance of curated knowledge for providing reliable, domain-aligned responses. While centered on task-oriented chatbots, their findings support the use of structured retrieval pipelines, complementing our modular RAG-based approach.

Brazilian contributions have also advanced MAS and RAG research. Barbosa et al.[de Albuquerque et al. 2024] assessed RAG effectiveness using implicit user feedback. Though more focused on usability, their findings support the relevance of RAG for efficient knowledge access.

Medeiros et al. [Medeiros et al. 2023] applied RAG to enhance LLM responses in educational benchmarks, demonstrating the impact of contextual retrieval even in centralized setups. Despite these advances, few studies systematically compare multi-agent and single-agent RAG architectures or explore environments with private, segmented knowledge bases and differentiated access policies.

This paper addresses that gap by proposing and evaluating a multi-agent RAG architecture with domain-specialized agents and a semantic router. Experimental results confirm improvements in precision, modularity, and scalability over centralized ap-

proaches, contributing to the development of secure and efficient semantic retrieval in enterprise contexts.

3. Methodology

This study employs an experimental methodology to systematically evaluate the performance of leading LLMs within a multi-agent RAG architecture. The methodology is structured around three core components: the system architecture, the experimental setup, and the quantitative evaluation metrics used to measure performance.

3.1. Multi-agent Architecture

The foundation of our experiment is a hierarchical multi-agent system, depicted in Figure 1. At the core of the architecture is a central Router Agent, which acts as the initial point of contact for all user queries. This agent's primary function is to perform a high-level classification of the query's intent and delegate it to one of three specialized downstream paths. For Generic Questions or conversational interactions, the query is passed to a Conversational Agent. Queries identified as pertaining to specific, high-priority external topics, such as those related to the ANEEL regulatory body, are directed to a dedicated Regulatory Agent. For queries related to the company's core operational domains (Internal Domain), the router delegates the task to a secondary-level Domain-Based Agent. This Domain-Based Agent then performs a more granular classification, routing the query to one of several highly specialized agents, including Audit, R&D, Legal, Sustainability, and Investor Relations. Regardless of the path taken, each terminal agent executes its task using a dedicated RAG pipeline, ensuring that the final answer generated by the LLM is grounded in the correct contextual knowledge from its exclusive vector store.

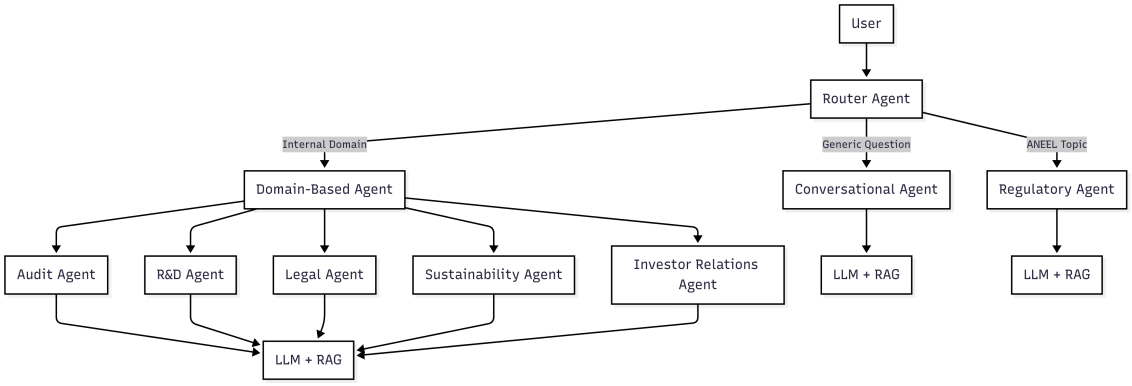

Figure 1. Hierarchical multi-agent architecture used in the experiment. A central router directs queries to either generic or specialized agents.

3.2. Experimental Setup

Eight state-of-the-art LLMs were selected to be evaluated for their generative and routing capabilities: GPT-4o, LLaMA 4, Claude-3.7, Gemini-2.5-Pro, Gemini-2.5-Flash, GPT-4.1, GPT-4.1-mini, and OpenAI o4-mini. The knowledge base consisted of proprietary corporate documents from the energy sector, chosen for their linguistic and technical complexity. To ground our evaluation in a realistic, non-English enterprise context, all source documents and evaluation queries were in Brazilian Portuguese. This context allows for

assessing model performance in a major global language that is often underrepresented in benchmarks compared to English.

A dataset of over 120 unique questions was manually curated, with at least 20 questions targeting the specific information within each of the six domains. For each test case, the workflow was standardized: (1) a query was sent to the router agent; (2) the router model delegated the query to a domain agent; (3) the domain agent performed a semantic search, retrieving the top-k relevant document chunks; and (4) the agent's LLM generated a final answer grounded in the retrieved context.

To ensure a fair comparison, all models were tested under identical conditions. Each model was evaluated both as a generative agent within the domains and as the decision-making model in the semantic router role. This allowed us to assess their performance in both content generation and query classification.

3.3. Experimental Configuration

All models were accessed via API under identical conditions: Gemini via google.genai (v1.13.0), Claude via anthropic.AnthropicVertex (v0.50.0), GPT via openai (v1.77.0), and LLaMA 4 (meta/llama-4-maverick-17b-128e-instruct-maas) through an OpenAI-compatible client on Vertex AI Model Garden. Tested models included: gpt-4.1-2025-04-14, claude-3-7-sonnet@20250219, gemini-2.5-flash-preview-04-17, gemini-2.5-pro-preview-05-06, gpt-4.1-mini-2025-04-14, gpt-4o-2024-08-06, o4-mini-2025-04-16, and meta/llama-4-maverick-17b-128e-instruct-maas.

All ran with temperature=1 and max_tokens=10000. Prompts contained only the domain-specific question to mimic real scenarios. The vector database was pgvector on Google Cloud Platform, using text-multilingual-embedding-002 (dim 1536) and a similarity threshold of 0.3. Latency metrics total and retrieval were measured in a single time window on the same machine and connection. Tests were repeated, but only the final run time was reported, as rankings remained consistent.

3.4. Evaluation Metrics

To measure the effectiveness and efficiency of each model, we employed a suite of automated evaluation metrics, primarily leveraging the RAGAS framework [Es et al. 2024] alongside custom scripts for logging time and accuracy. The key metrics were: Answer Relevancy, which assesses the semantic alignment between the generated answer and the original user query; Faithfulness, which measures the factual consistency of the generated answer against the retrieved document context to avoid hallucination; Router Accuracy, which evaluates the percentage of queries the semantic router correctly assigns to the appropriate domain-specific agent; and Execution Time, which records the average end-to-end response time in seconds as a proxy for system latency.

4. Results

This section presents a comparative analysis of various approaches and components involved in constructing the RAG-based system. We begin by evaluating the performance of single-agent and multi-agent architectures using metrics such as *Answer Relevancy*, *Faithfulness*, average response time, and routing accuracy. Subsequently, we conduct a more in-depth assessment of the language models used in the system, examining their behavior across various thematic domains.

4.1. Answer Relevancy Evaluation

Figure 2 presents the aggregate performance ranking of the evaluated LLMs based on their average answer relevancy across all domains. The results depict a highly competitive landscape at the top, with GPT-4o (0.936), LLaMA 4 (0.935), and Gemini-2.5-Pro (0.934) achieving nearly indistinguishable scores, establishing them as the premier models in this evaluation. Following this leading cluster, a second tier of high-performing models includes Gemini-2.5-Flash (0.922) and OpenAI o4-mini (0.919). The ranking clearly illustrates a performance hierarchy, with the GPT-4.1 series occupying the middle ground and Claude-3.7 (0.849) positioned as the lowest-performing model. This high-level overview provides a baseline for understanding the general capabilities of each model before delving into its domain-specific strengths and weaknesses.

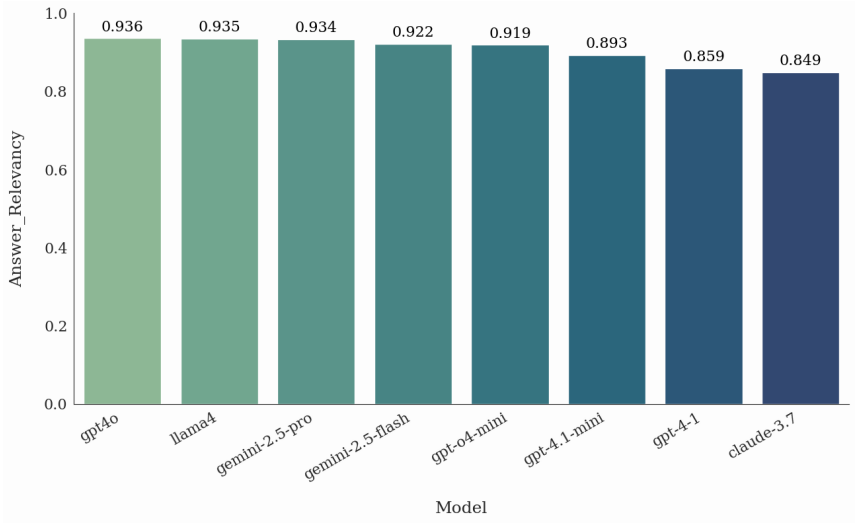

Figure 2. Model ranking based on answer relevancy.

The empirical results demonstrate significant performance variations in answer relevancy, as illustrated by the overall model ranking in Figure 2 and the domain-specific breakdown in Table 1. The aggregate scores from Figure 2 position GPT-4o (0.936), LLaMA 4 (0.935), and Gemini-2.5-Pro (0.934) as the top-tier models, with only marginal differences in their overall performance. However, Table 1 reveals the underlying specializations that contribute to these high-level rankings. For instance, GPT-4o's leading position is driven by its top score in the Audit domain (0.967), while LLaMA 4's strong average is bolstered by its exceptional performance in Legal (0.977) and Sustainability (0.969). Meanwhile, Gemini-2.5-Pro, despite being third overall, is the clear leader in the R&D domain (0.952). Conversely, Claude-3.7's position as the lowest-ranked model (0.849) is explained by its weaker performance in domains like Audit (0.776) and Legal (0.793).

This combined analysis validates the core hypothesis that a multi-agent architecture is effective. But relying solely on aggregate performance metrics can be misleading for specialized applications. While Figure 2 provides a useful high-level summary, the domain-specific data in Table 1 is essential for practical implementation. For example, a system designed primarily for R&D tasks would benefit most from deploying Gemini-2.5-Pro, even though it is not the top-ranked model overall. These findings confirm that the

modular, multi-agent approach not only improves contextual alignment but also enables a more granular and optimized model selection strategy, allowing for the deployment of the best-suited LLM for each specific enterprise domain.

	Audit	Legal	R&D	Regulatory	IR	Sustainability
Claude-3.7	0.776	0.793	0.837	0.897	0.929	0.863
Gemini-2.5-Flash	0.891	0.870	0.917	**0.973**	0.918	0.965
Gemini-2.5-Pro	0.929	0.933	**0.952**	0.932	0.919	0.938
GPT-4.1	0.867	0.842	0.778	0.875	0.923	0.866
GPT-4.1 mini	0.879	0.807	0.915	0.877	0.918	0.962
OpenAI o4-mini	0.912	0.893	0.914	0.923	0.928	0.947
GPT-4o	**0.967**	0.930	0.883	0.968	0.908	0.957
LLaMA 4	0.940	**0.977**	0.857	0.917	**0.955**	**0.969**

Table 1. Average answer relevancy per domain. The highest score in each domain is highlighted in bold.

4.2. Faithfulness Evaluation

In addition to relevancy, we evaluated answer faithfulness, a metric that measures the extent to which a model's response is verifiably grounded in the provided source documents, thereby minimizing hallucination. Figure 3 illustrates the faithfulness metric. A striking finding is the reordering of the model hierarchy compared to answer relevancy, highlighting that the two metrics are not directly correlated. The data reveals that faithfulness is a more challenging task, with the top score, achieved by gpt-4.1-mini (0.795), being substantially lower than the top relevancy score. Following closely are gpt4o (0.790) and Gemini-2.5-Flash (0.784), forming a competitive leading tier. Notably, models that were top performers in relevancy, such as LLaMA 4, are positioned in the middle of the pack here, while OpenAI o4-mini (0.679) shows the weakest performance in factual grounding. This ranking underscores that a model's ability to generate a relevant response is distinct from its ability to ensure that the response is verifiably accurate to the source material.

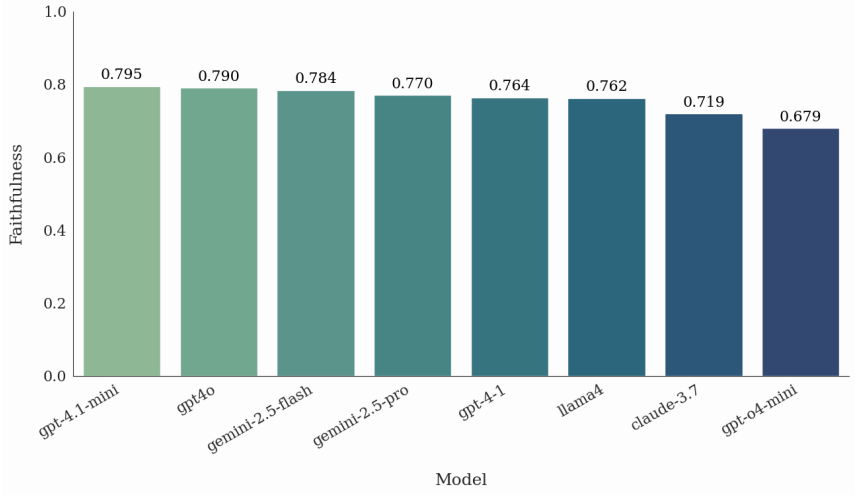

Figure 3. Model ranking based on answer faithfulness.

The results, presented in Table 2, show a more challenging landscape compared to relevancy, with overall scores being notably lower. Gemini-2.5-Flash demonstrates good faithfulness, particularly in the Audit domain (0.885), positioning it as a highly reliable choice for fact-based retrieval. GPT-4.1-mini also demonstrates strong performance, achieving the highest scores in the challenging Investor Relations (IR) domain (0.669) and in Sustainability (0.833). While models like GPT-4o and LLaMA 4 remain competitive, the general decrease in scores across all models underscores the inherent difficulty of maintaining strict factual consistency.

A key observation from this data is the universally poor performance across all models in the Investor Relations (IR) domain, which consistently yields the lowest faithfulness scores. This suggests that the nuanced, forward-looking, and often disclaimer-heavy language of IR documents poses a challenge for grounding answers factually, even for top-tier models. This finding highlights the paramount importance of the faithfulness metric, as a relevant but unfaithful answer can be dangerously misleading. For enterprise systems deployed in regulated environments, these results indicate that while a multi-agent architecture correctly routes queries, the final choice of LLM for each agent must prioritize high faithfulness to ensure the system's trustworthiness and mitigate the risk of generating inaccurate information.

	Audit	**Legal**	**R&D**	**Regulatory**	**IR**	**Sustainability**
Claude-3.7	0.756	0.737	0.722	0.787	0.510	0.804
Gemini-2.5-Flash	**0.885**	0.770	**0.830**	0.829	0.548	0.843
Gemini-2.5-Pro	0.828	0.837	0.808	0.81	0.530	0.809
GPT-4.1	0.782	**0.862**	0.713	0.817	0.547	0.864
GPT-4.1 mini	0.811	0.819	0.805	**0.831**	**0.669**	0.833
OpenAI o4-mini	0.746	0.650	0.708	0.731	0.532	0.706
GPT-4o	0.808	0.758	0.817	0.806	0.633	**0.90**
LLaMA 4	0.818	0.817	0.769	0.823	0.490	0.832

Table 2. Average answer faithfulness per domain. The highest score in each domain is highlighted in bold.

4.3. Router Performance per Model

The effectiveness of the multi-agent architecture fundamentally depends on the semantic router's ability to correctly direct incoming queries to the appropriate specialized agent. Figure 4 presents the accuracy of each LLM when tasked with this routing function. The results reveal a distinct performance hierarchy, with LLaMA 4 emerging as the most capable router, achieving the highest accuracy at 73.9%. It is closely followed by a competitive tier comprising gpt-4.1-mini and Gemini-2.5-Flash, both of which scored an identical 72.8%. Interestingly, some of the models that excelled in generative tasks, such as GPT-4o (66.1%) and Gemini-2.5-Pro (70.0%), exhibited more moderate routing capabilities. This disparity suggests that the skills required for effective semantic routing are distinct from those for content generation, making the selection of the router model a critical and independent design choice for optimizing the end-to-end performance of the multi-agent system.

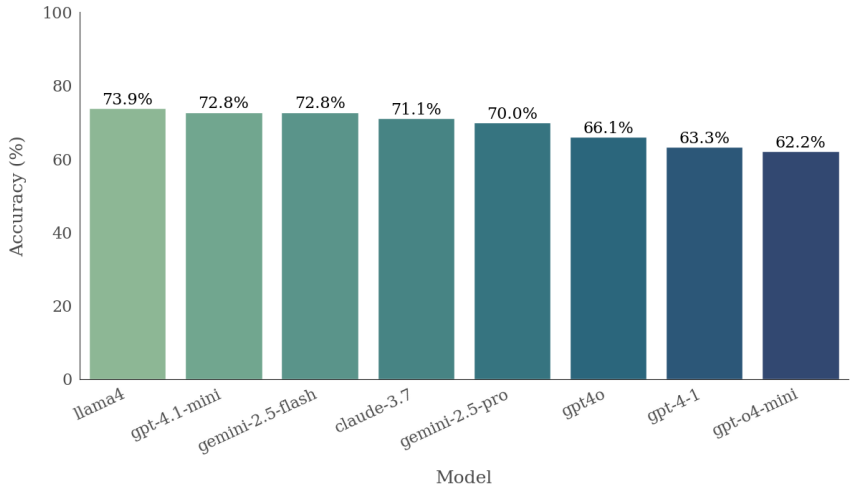

Figure 4. Routing accuracy per model.

4.4. Average Execution Time

In addition to accuracy and faithfulness, processing efficiency is a important factor for the practical deployment of enterprise RAG systems. Figure 5 presents a comparative analysis of the average end-to-end response time for each model, revealing stark differences in latency. LLaMA 4 (6.289s) and Gemini-2.5-Flash (6.687s) emerge as the clear leaders in efficiency, delivering responses in under seven seconds. At the opposite end of the spectrum, Gemini-2.5-Pro is a significant outlier, with an average processing time of nearly 50 seconds (49.959s). Between these extremes, the remaining models form distinct performance tiers, with gpt4o and gpt-4.1-mini demonstrating moderate latency (around 17s), while models like Claude-3.7 and gpt-4-1 cluster around the 24-second mark. These findings highlight a trade-off: while a model like Gemini-2.5-Pro may offer high relevancy, its prohibitive latency may render it unsuitable for many real-time or user-facing applications, reinforcing the value of faster, specialized models like LLaMA 4 and Gemini-2.5-Flash for latency-sensitive deployments.

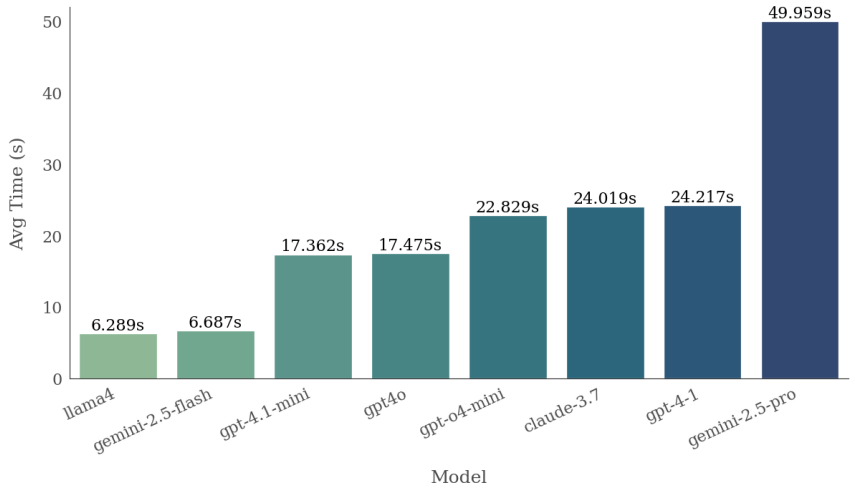

Figure 5. Average response time per model.

4.5. Discussion

Our results reveal a set of trade-offs between answer quality, factual integrity, and latency, providing a clear and practical rationale for the multi-agent RAG architecture. No single LLM excels across all metrics, making a monolithic approach inherently suboptimal. While models like GPT-4o, LLaMA 4, and Gemini-2.5-Pro demonstrate top-tier relevancy, their strengths are domain-specific, validating the use of specialized agents.

The prohibitive latency of Gemini-2.5-Pro makes it impractical for user-facing applications, despite its high answer quality. In contrast, LLaMA 4 and Gemini-2.5-Flash emerge as an excellent balance of high accuracy and low latency, making them prime candidates for production environments. Furthermore, the divergence between relevancy and faithfulness rankings underscores that generating a relevant answer is distinct from ensuring it is factually grounded. The poor faithfulness scores across all models in the Investor Relations (IR) domain highlight the risk of generating plausible-sounding hallucinations, a critical failure point in regulated settings.

Finally, the strong performance of LLaMA 4 as a semantic router—a task where it outperformed models that were superior in generative tasks—proves that query routing is a specialized skill. This finding reinforces the value of a modular design where each component is independently optimized. In essence, building an effective enterprise RAG system requires designing a system of specialized agents, where each component is equipped with the model that offers the optimal balance of relevancy, faithfulness, and latency for its specific function, all operating within a cohesive and intelligently managed architecture.

5. Conclusion and Future Work

This study shows that a multi-agent RAG architecture with domain-specialized agents guided by a semantic router is more effective for enterprise knowledge retrieval than monolithic systems. The modular design improves contextual accuracy, factual faithfulness, and allows better management of trade-offs between performance, efficiency, and cost. Our results indicate that no single LLM leads in all metrics; GPT-4o, Gemini-2.5-Flash, and LLaMA 4 offer the best balance between quality and latency, making them strong production candidates. The semantic router's efficiency in directing queries reinforces its value for domain-specific governance in regulated environments. Data privacy remains a key challenge for real deployments, where sensitive information cannot be externalized. In such contexts, high-performing open-source models like LLaMA 4 are attractive, as they can be self-hosted to meet enterprise security requirements.

Future work includes incorporating human evaluations to assess clarity, tone, and perceived utility; testing fine-tuned domain agents; extending to multilingual contexts; and developing adaptive routing strategies. These steps aim to enhance the reliability and specialization of multi-agent RAG systems for real-world use.

6. Acknowledgements

This work has been funded by P&D CEMIG/ANEEL PD-04950- D0677/2023, and was also supported by the National Institute of Science and Technology (INCT) in Responsible Artificial Intelligence for Computational Linguistics and Information Treatment and Dissemination (TILD-IAR) grant number 408490/2024-1.

References

Broda, E. (2025). Agentic mesh: Patterns for an agent ecosystem. *Medium – Data Science Collective*. https://medium.com/data-science-collective/agentic-mesh-patterns-for-an-agent-ecosystem-ef13469b7cf7.

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., and Xie, X. (2024). A survey on evaluation of large language models. *ACM Computing Surveys*, 15(3):1–45.

da Costa, L. and e Souza Filho, J. O. (2024). Adapting llms to new domains: A comparative study of fine-tuning and rag strategies for portuguese qa tasks. In *Anais do XV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 267–277, Porto Alegre, RS, Brasil. SBC.

de Albuquerque, A. M., Wensing, I. M., Joppi Filho, N. L., and Dorneles, C. (2024). Avaliação de aplicações de geração aumentada de recuperação por meio de feedback implícito. In *Simpósio Brasileiro de Banco de Dados (SBBD)*, pages 253–259. SBC.

Dulay, H. (2024). Event-driven agent mesh. *Medium*. https://medium.com/@hubert.dulay/event-driven-agent-mesh-be29f1c36932.

Enterprise, H. P. (2024). Llm agentic tool mesh: Harnessing agent services and multi-agent ai for next-level gen ai. https://shorturl.at/Uf9ST. Accessed: 2025-06-22.

Es, S., James, J., Anke, L. E., and Schockaert, S. (2024). Ragas: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 150–158.

Han, X., Wang, W., Cao, Y., Zhang, Y., Hu, Z., Jiang, J., Yao, Q., Lin, Y., Liu, Z., and Sun, M. (2024). Llm multi-agent systems: Challenges and open problems. *arXiv preprint arXiv:2402.03578*.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020a). Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

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. (2020b). 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, G. H., Souza, T. F., Bezerra, K. C., and Santana, E. C. (2023). Using retrieval-augmented generation to improve performance of large language models on the brazilian university admission exam. In *Anais do Simpósio Brasileiro de Banco de Dados (SBBD)*. SBC.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. (2023). Generative agents: Interactive simulacra of human behavior.

Puschmann, J., Dietz, L., Behnke, S., and Wehrle, K. (2022). Multi-agent document indexing with topic-specific retrieval pipelines. *Proceedings of the 45th International*

ACM SIGIR Conference on Research and Development in Information Retrieval, pages 1349–1359.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. (2023). Reflexion: language agents with verbal reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Siqueira, C., Fonseca, O., Ferreira, G., and Leiva, O. (2024). Leveraging structured data input for effective chatbot integration in enterprises. In *Proceedings of the 15th Brazilian Symposium in Information and Human Language Technology*, pages 1–5, Porto Alegre, RS, Brasil. SBC.

Team, L. A. (2025). Agentmesh: Unfolding the communication of multiple ai agents. `https://shorturl.at/hEjmb`. Accessed: 2025-06-22.

Wooldridge, M. (2009). *An introduction to multiagent systems*. John wiley & sons.