Performance Enhancement of Agentic Retrieval Augmented Generation Using Relevance Generative Answering

Sanjay Kukreja
Department of Machine Learning
SP Jain School of Global Management
Mumbai, India
sanjay.ds18dba008@spjain.org

Tarun Kumar
COE AI-ML
eClerx Services Ltd.
Chandigarh, India
tk7ua1678@gmail.com

Vishal Bharate
COE AI-ML
eClerx Services Ltd.
Pune, India
vishalbharate@gmail.com

Sweta Gadwe
COE AI-ML
eClerx Services Ltd.
Pune, India
swetagadwe1996@gmail.com

Abhijit Dasgupta
Department of Data Science
SP Jain School of Global Management
Mumbai, India
abhijit.dasgupta@spjain.org

Debashis Guha
Department of Machine Learning
SP Jain School of Global Management
Mumbai, India
debashis.guha@spjain.org

2025 5th International Conference on Artificial Intelligence and Education (ICAIE) | 979-8-3315-2295-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICAIE64856.2025.11158296

***Abstract*— The aim of this research paper is to present a novel approach of using Relevance Generative Answering (RGA) in the trending field of Agentic Retrieval Augmented Generation (RAG). The paradigm shift in the RAG system by the introduction of Agentic RAG has opened a new research paradigm. The major issue of hallucination is overcome with the use of a traditional RAG system with some limitations like accuracy and relevance, lack of reasoning, the lost in the middle problem, etc. The Agentic RAG system attempts to address a few of these limitations. However, interpreting results based on the user's intent remains a significant area of research. This research aimed to understand user intent by introducing relevance detection block in the proposed architecture. Different performance metrics like precision, recall, F1 score, relevance, latency are used to validate the proposed approach. The results presented in this research reveal that the performance of the proposed system is much more relevant compared to agentic RAG system. For context and intent specific applications proposed framework suits well.**

Keywords*— *Artificial Intelligence, Agentic Retrieval Augmented Generation, Relevance Generative Answering, Generative AI, Large Language Model, Natural Language Processing.

I. INTRODUCTION

The rise of Generative AI has changed the way applications and services are provided to the end users. As Large Language Models (LLM) are trained on previously available datasets, they lack recent, real-time knowledge and try to answer the query in an inappropriate way without having context [2]. Hence, LLMs hallucinate to respond to user queries that are dependent on real-time context. This significant issue of hallucination is resolved using a Retrieval Augmented Generation (RAG) system. RAG emerged as a solution by integrating real-time data retrieval into LLMs, enhancing their contextual relevance [3]. Despite this, traditional RAG systems struggle with multi-step reasoning, adaptability, and complex task management due to rigid workflows [1].

RAG has made significant changes in the applications, like an internal Chatbot to assist employees of a company where context-relevant and optimum answers based on a knowledge base are required [5]. Researchers have researched and implemented multiple advancements in the RAG system, such as Naive RAG, Advanced RAG, and Modular RAG [7]. More recently, the concept of Agentic RAG has emerged. This approach goes one step further by embedding autonomous agents into the RAG pipeline to manage retrieval, reasoning, and generation in a dynamic and iterative way [1, 4]. The agentic framework has several advantages, like resolving complex queries, real-time decision-making, learning over time, flexibility, customization, improved efficiency, and accuracy, etc. [9, 12]. Agentic RAG breaks the question into different manageable steps and assigns appropriate agents to associated tasks. The feature of the agent is that it can understand and can assign tasks seamlessly to different agents and integrate the information received from different agents. Additionally, their design allows them to continuously learn and enhance themselves. Traditional RAG systems do not adapt if the first round of retrieval or generation does not meet quality criteria. Agentic RAG addresses limitations of traditional RAG by embedding autonomous AI agents into the RAG pipeline [10]. These agents employ agentic design patterns namely, reflection, planning, tool use, and multi-agent collaboration to dynamically manage retrieval strategies, refine contextual understanding, and adapt workflows for complex tasks [14].

Though it has several advantages, there are certain challenges as well to implement agentic RAG systems. These

challenges include complexity in the coordination of multiple autonomous agents, an increase in computational resources for complex workflows, and scalability limitations. Another limitation is that the agent is unable to recognize the intent of the user queries. The major research is required in this field to get the most accurate response to the user query according to the intent of the user query.

By leveraging these advantages of agents in the RAG system, gave rise to Agentic RAG system arose. As agents have different capabilities like resolving complex queries, retrieving relevant contexts efficiently, allow to connect with real-time data, they provide excellence in the traditional RAG system [12, 15]. Adopting agents enhances the accuracy and adaptability of the RAG framework.

In this research, we try to overcome the challenge of retrieving intent from the user query on ambiguous tasks by generating similar types of queries and giving them to agents for identifying answers to those queries and generating the response, which is according to the intent of the user query. The Relevance Generative Answering (RGA) technique is a complex AI- based method for giving users answers that are very relevant and accurate in their context [6]. It integrates relevance detection with generative answering to guarantee that the responses are both contextually suitable and customized to the user's specific requirements. The integration of Relevance Generative Answering with Agentic RAG is a novel contribution.

The following sections further structure this paper. Section II represents related work of agents and agentic RAG systems. It conducts a thorough literature survey on existing agentic RAG systems. Section III discusses the methodology of the proposed agentic RAG system. Section IV highlights the evaluation metrics and evaluation results obtained after experimenting with the proposed framework. Section V gives a brief overview of the advantages, a thorough discussion of challenges and future trends associated with the agentic RAG system. We conclude the research paper by summarizing the key contributions and findings of the proposed framework.

II. Related Work

The traditional RAG system follows a linear workflow which includes retrieval of data from external knowledge bases, augment the LLM contexts and generate the responses [2, 3]. It is effective for the basic tasks, but when ambiguous or complex queries are asked by user, traditional RAG system fails as they lack in the flexibility of handling multi-domain data and reasoning of selecting the contexts. The limitations of traditional RAG include static workflows and contextual gaps.

The evolution of RAG is broadly categorized as Naïve RAG, Advanced RAG and Modular RAG [3, 7, 8]. Naïve RAG depends on keyword based retrieval and static datasets, whereas advanced RAG relies on the semantic search of dense vector retrieval from vector database and iterative query refinement. Advanced RAG struggles with computational overhead. Modular RAG decomposes retrieval and generation into reusable components which enables domain-specific customization. These advancements in the traditional RAG systems enables Agentic RAG that involves agentic intelligence to overcome challenge of dynamic workflows.

Agentic RAG systems include autonomous agents which employs agentic design patterns to enhance decision-making [1, 9]. The agentic patterns include reflection, planning and multi-agent collaboration. Planning stage evaluates response generated, identify errors and based on the generated response and error, iteratively refine the response [4, 10]. In the planning stage agents decompose complex tasks into subtasks. In multi-agent collaboration, specialized agent monitors the distributed tasks to subagents [13, 14]. On the other hand, work flow pattern includes prompt chaining, routing, orchestration of working models and evaluator optimization [5, 12]. Prompt chaining breaks the tasks into sequential steps. Routing directs the queries to specialized agents and Orchestrator which acts as a central agent delegates subtasks to workers for parallel processing of the query. Using iterative feedback loop Evaluator optimizes the final response.

The agentic RAG systems are classified into Single-Agent, Multi-Agent, Hierarchical, Adaptive and Graph based depending on the operational structure. In Single-Agent system, a single agent manages retrieval and generation task [5]. It is known for its simplicity and implementation cost which is low, but lack with scalability for complex tasks. In Multi-Agent systems, teams of agents collaborate with each other [12]. These systems are efficient in handling multistep tasks but includes coordination complexity and redundancy risks. Agents are organized in hierarchies in Hierarchical systems [9]. These systems are mainly used in time-series analysis. In case of Adaptive systems, agents dynamically adjust the retrieval strategies based on the contexts [12]. Graph based systems integrates graph databases to capture the entity relationships [8].

III. Methodology of Proposed Agentic RAG

Fig. 1 presents the proposed agentic RAG system, which includes relevance detection. Major blocks in this system include relevance detection, agents, knowledge sources, generative answering, the output layer, and the user interaction layer. The user interaction layer accepts the queries from the user and provides generated outputs to the user. We have designed each block with a specific purpose in mind.

The proposed application is built for persons who are competitive exam aspirants. Once the user raises the query, it goes to the relevance detection block. The relevance detection block contains natural language understanding (NLU), semantic analysis, and relevance classification. NLU plays a crucial role not only in understanding words but also in understanding the intent behind the query and recognizing entities in the query, which is generic in nature. The intent may be seeking the information or making a request, etc. As we know the purpose for the developed application, we can classify this intent/relevance into different classes. Once we recognize the query's intent, we provide the meta agent with the classified query. The main task of the meta agent is to orchestrate the complete workflow. Based on user query and intent, the meta agent decides from where to retrieve the relevant data and how to manage interactions between specialized tools and sub-agents. It acts as a central decision maker, which enables intelligent reasoning, dynamic tool selection, and orchestration for complex tasks.

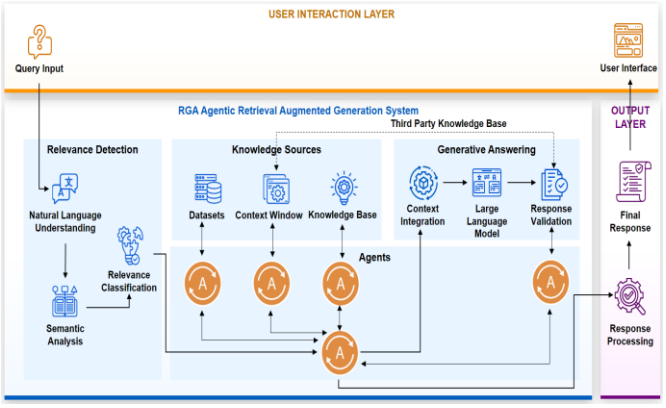

Fig. 1. Block Schematic of Proposed Framework of Agentic RAG System

Fig. 1 illustrates the meta agent's ability to interact with various sub-agents, each assigned specific tasks. These subagents can also interact with each other and coordinate with the meta agent. One sub-agent can retrieve information from the relevant proprietary dataset based on the user's query. The context window will review the user's past interactions with the developed system to determine the user's motivation for raising the query. The knowledge base is a repository of data sourced from websites such as Wikipedia, which provides real-time access to relevant information. The meta agent receives relevant details from datasets, context windows, and knowledge bases.

The meta agent collects all these relevant details and provides them to the generative answering block. The generative answering block consists of three major blocks, namely context integration, large language model (LLM), and response validation. The context integration block collects all relevant data associated with the query from the meta agent and ranks it according to the query. The contexts are provided to the large language model along with the user query and intent. LLM generates a comprehensive response according to the query and all relevant contexts received from the context integrator. A third-party knowledge base, containing Wikipedia and web pages, checks the generated response for validation. The sub-agent provides the meta agent with the validated response, reflecting the user's appropriate intent.

Further, the meta agent provides a validated response to the output layer. The output layer contains response processing and final response blocks. The response processing block removes any double spaces or irrelevant data from the response. The user interface layer receives the cleaned final response. The user interface layer is responsible for displaying responses to the end-user.

The models/technology used for experimentation of the proposed framework are presented in Table I. We used state-of-the-art models that specialize in the relevant tasks of the proposed framework. Table I presents the different models used in the context retrieval block, which contain datasets, a context window, and a knowledge base. There is scope to choose different models to optimize the responses for different use cases. The combination of these models can also be tested to obtain better accuracy with relevance.

TABLE I. TECHNOLOGY/MODELS USED FOR PROPOSED RELEVANCE BASED GENERATION AGENTIC RAG SYSTEM

Name of the Block	Technology/Model Stack
Natural Language Understanding and Semantic Analysis	Claude 3.7 Sonnet
Relevance Classification	Gemini 2.0 Pro
Context Retrieval	Vector DB-Deepseek R1 Unstructured search- GPT 4.5 Hybrid Retriever
Context Integration & Response Generation	Gemini 2.0 pro/ GPT-4o (OpenAI)
Response Validation & Processing	GPT-o3-mini

IV. EALUATION METRICS AND RESULTS

A. *Evaluation Metrics*

Different evaluation metrics measure the performance of the proposed agentic RAG system using various autonomous agents. We used relevance, precision, recall, and F1-scores to measure the performance of the developed agentic RAG system. The subsequent paragraphs provide a detailed description of the various metrics employed for performance evaluation.

- Precision- Precision is the ratio of true positive observations to the total number of observations identified as positive. A high level of precision is desired.
- Recall- Recall, which is also known as the true positive rate or sensitivity, measures how accurate the positive observations were compared to all the other observations about a hypothesis.
- F1 score- It represents the harmonic mean of precision and recall. It ought to be of greater value.
- Relevance- It is the degree to which something is connected or significant to a particular situation, need, or context. It is often used to evaluate the importance or applicability of information, ideas, or objects.

We selected the National Council of Educational Research and Training (NCERT) books dataset [16]. We calculate the evaluation metrics using the NCERT solution dataset [17] as the ground truth. To evaluate relevance metric, we assigned a subject matter expert (human expert) for evaluation of it. We applied the standard process to avoid biases in the responses of the subject matter expert. We asked experts to evaluate the response based on different questions, like, "Does the response match the NCERT textbook contents?", “Are the key terms and definitions used correctly?”, “Does the answer avoid unsupported claims?” etc. We asked the expert to rate the generated response for both systems on a scale of 1-5. 1 indicates an irrelevant response, whereas 5 indicates a highly relevant response.

B. *Evaluation Process*

The NCERT solution dataset with ground-truth answers is available for evaluation of the proposed approach. We selected 200 different questions from the test dataset. The evaluation

metrics discussed above are computed when each query is executed by the system. We adopted two approaches to evaluate the system performance: a simple agentic RAG system and a proposed relevance-based generative agentic RAG system. The simple agentic RAG system did not compute relevance. The major difference between the two systems is that the relevance detection block was not present in the simple agentic RAG system.

C. Result and Discussion

Table II presents the average results for all 200 queries, along with selected evaluation metrics. Table II represents a comparison of a simple agentic RAG system and the proposed relevance-based generation agentic RAG system. From Table II, it can be seen that the time to respond to user queries was much less in the simple agentic RAG system than in the proposed relevance-based agentic RAG system. For all other metrics, namely, precision, recall, and F-1 score, the proposed relevance-based agentic RAG system performs much better. The precision value of 74% for the simple agentic RAG system is due to generic retrieval without relevance filtering. However, the 83% precision value of the proposed system indicates superior performance for filtering out irrelevant/off-topic responses.

Similarly, the recall metric indicates that the simple agentic RAG system retrieves broader but less accurate content, whereas the proposed system focuses on contextually aligned answers. The balanced F1 score for agentic RAG is due to the trade-off between precision and recall. The 92% F1-score of the proposed system indicates prioritizing context-aware, NCERT-aligned responses. As the simple agentic RAG uses lightweight retrieval, the time to respond to user queries is much less than that of the proposed relevance-based generation agentic RAG system. The high latency of the proposed system can be justified as it produces superior quality results due to relevance-based filtering and generation.

For an application where time plays a crucial role, one can select a simple agentic RAG system, but if a more relevant response is a concern, one can choose the proposed relevance-based generation agentic RAG system. Students preparing for competitive examinations can use the proposed system, as it underwent testing on the NCERT books dataset.

TABLE II. COMPARISION OF PERFORMANCE METRICS FOR SIMPLE AGENTIC RAG SYSTEM AND PROPOSED RELEVANCE BASED GENERATION AGENTIC RAG SYSTEM

Metric	Simple Agentic RAG system	Proposed Relevance based generation Agentic RAG system
Precision	74%	83%
Recall	63%	79%
F1 Score	68%	81%
Relevance	70%	92%
Time to Repond (Latency)	1.2 seconds	2.8 seconds

V. ADVANTAGES, CHALLENGES AND FUTURE TRENDS OF AGENTIC RAG SYSTEM

A. Advantages of Agentic RAG

The advantages of Agentic RAG are discussed as follows.

- Enhanced Retrieval- The search precisions are enhanced by hybrid search methodologies and re-ranking algorithms available in agentic RAG. Also content representation and relevance identifications are improved by using multiple vectors per document.
- Dynamic Planning and Execution- Handling of complex queries is made more efficient by using dynamic agents of agentic RAG. It uses real time planning, execution and optimization of query processes to adopt real time evolving information landscape.
- Intelligent Quality Control- To generate reliable and accurate outputs agentic RAG agents evaluate, correct, and verify the information gathered and filter out unreliable information.
- Scalability and Extensibility- Agentic RAG systems allows extension and easy scaling of functionalities. The integration of new data and tools can be easily achieved using modular nature of agentic RAG systems.
- Semantic Coherence- Agentic RAG is able to recognize semantic arguments that are typically overlooked in keyword-based searches, and better understands context, resulting in increased accuracy and recall.
- Breaks Down Complex Tasks- AI agents can break down complex tasks into several subtasks so they become easier to handle.
- Purpose-Driven Approach- Agentic RAG creates a purpose-driven approach to retrieval.
- User Experience- Agentic RAG can provide faster response times, more relevant and accurate answers, personalized information retrieval based on user context and preferences, and intuitive and seamless interactions that simplify complex information retrieval tasks.

B. Challenges of Agentic RAG

- Data Quality and Relevance- The responses may be misleading or inaccurate due to poor data quality issues and their relevance. In order to get quality results high-quality data is required to integrate with agentic RAG systems.
- Latency- The latency in agentic RAG plays crucial role. Data validation steps and multi-step reasoning may introduce latency. To overcome this challenge system architecture and processing capabilities should be optimized.
- Hallucination- The complexity in processing queries may amplify the incorrect information gathered during the retrieval process. This may lead to hallucinate the responses of agentic RAG.

- Model Integration- The integration of retrieval mechanism with the models generating the responses is crucial and challenging task. The choice of LLM to get relevant and accurate results and integrate them with retrieval task requires much attention while designing the system.
- Ethical Considerations- The deployment of agentic RAG system requires addressing of bias, data privacy and explainability feature at every stage as agents are autonomous in nature.
- Scalability Issues- The agentic RAG system grows as volume and diversity of data increases which leads to limitation on scalability.

C. *Future Trends of Agentic RAG*

The future trends of Agentic RAG are illustrated as follows.

- Improved Accuracy and Speed- The advancements in data processing and AI algorithms may enhance the speed and accuracy of agentic RAG system. These advancements will enable the agentic RAG system to handle even more complex queries and speedy output delivery.
- Integration with Emerging Technologies- The technologies like IOT, blockchain, quantum computing will leverage the agentic RAG systems for more reliable and real-time data integration. This requires an interdisciplinary research in different the domains to improve overall efficiency if the system.
- Multi-Modal Retrieval- The integration of text, image, audio, and video require to make response context rich. To collaborate with different modalities of system further research is required.
- Cross-Lingual Capabilities- Agentic RAG will operate across multiple languages, breaking language barriers and broadening its global applicability.

VI. CONCLUSION

This paper discusses the proposed novel architecture of Agentic RAG with relevance generative answering. This novel framework of Agentic RAG system has shown tremendous change in the performance of RAG systems. The measured results indicate that the system is working well for a known purpose of application. The precision, recall, F1 score and relevance metrics outperform over agentic RAG system. The proposed architecture is well suited for applications where highly contextual results are required. The increased latency of the proposed system could be a drawback in time-sensitive applications. For general purpose applications and time sensitive applications simple agentic RAG system works better. In this research, we try to overcome the challenge of retrieving intent from the user query on ambiguous tasks and generating relevant answers according to the user intent. The proposed framework may be used and modified according to the complexity of the application.

In future researchers can work on different challenges addressed in this research work. Different applications and advantages are also addressed in this research work. Challenges like latency, data quality can be further researched to enhance the performance of proposed system.

REFERENCES

[1] A. Singh, A. Ehtesham, S. Kumar, and T. T. Khoei, "Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG," arXiv:2501.09136, 2025.

[2] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Proceedings of the 34th International Conference on Neural Information Processing Systems, 2020, pp. 9459–9474.

[3] V. Karpukhin et al., “Dense passage retrieval for open-domain question answering,” in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 6769–6781.

[4] R. Chidaksh, S. S. Sakhinana, and R. Venkataramana, “Agentic Retrieval-Augmented Generation for time series analysis,” arXiv 2408.14484 [cs.AI], 2024.

[5] D. Richards, “Agentic RAG: A complete guide to agent-based retrieval augmented generation,” News from generation RAG – Dive deep into the transformative world of AI Retrieval Augmented Generation (RAG) technologies, 23-Nov-2024. Available: https://ragaboutit.com/agentic-rag-a-complete-guide-to-agent-based-retrieval-augmented-generation/, accessed Feb. 12, 2025.

[6] S. Robik, Z. Yang, C. Qiuyu, L. Zhiheng, X. Yusheng, and D. Siqi, "FairRAG: Fair Human Generation via Fair Retrieval Augmentation," arXiv:2403.19964, 2024.

[7] G. Shailja, R. Rajesh, and S. N. Singh, "A Comprehensive Survey of Retrieval-Augmented Generation (RAG): Evolution, Current Landscape and Future Directions," arXiv preprint arXiv:2410.12837, 2024.

[8] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang, “REALM: retrieval-augmented language model pre-training,” Proceedings of the 37th International Conference on Machine Learning (ICML'20), Vol. 119. JMLR.org, Article 368, 3929–3938.

[9] K. Prantikos, V. Pallikaras, and M. Pantopoulou, “Agentic Retrieval Augmented Generation for Advanced Reactor Thermal Hydraulic System,” ResearchGate, 2024. [Online]. Available: https://www.researchgate.net/publication/387798703.

[10] X. Li et al., “Search-o1: Agentic search-enhanced large reasoning models,” arXiv preprint arXiv:2501.05366, 2025. [Online]. Available: https://arxiv.org/abs/2501.05366.

[11] N. Dinh and T. Chan, “ENTAgents: AI Agents for Complex Knowledge Otolaryngology,” medRxiv, 2025. [Online]. Available: https://www.medrxiv.org/content/2025.01.01.25319863.

[12] K. Parthasarathy, K. Vaidhyanathan, and R. Dhar, “Engineering LLM Powered Multi-agent Framework for Autonomous CloudOps,” arXiv preprint arXiv:2501.08243, 2025. [Online]. Available: https://arxiv.org/abs/2501.08243.

[13] T. Wang, N. Zhou, and Z. Chen, “CyberMentor: AI Powered Learning Tool Platform to Address Diverse Student Needs in Cybersecurity Education,” arXiv preprint arXiv:2501.09709, 2025. [Online]. Available: https://arxiv.org/abs/2501.09709.

[14] J. Tang, T. Fan, and C. Huang, “MetaChain: A Fully-Automated and Zero-Code Framework for LLM Agents,” arXiv preprint arXiv:2502.05957, 2025. [Online]. Available: https://arxiv.org/abs/2502.05957.

[15] R. Khanda, “Agentic AI-Driven Technical Troubleshooting for Enterprise Systems,” arXiv preprint arXiv:2412.12006, 2024. [Online]. Available: https://arxiv.org/abs/2412.12006.

[16] NCERT Class 10 History Books PDF Download - NCERT Books.

[17] Class 10 History NCERT Solutions PDF (ncrtsolutions.in).