IIAI Open Conference Publication Series
IIAI Letters on Informatics and Interdisciplinary Research
Vol.006, LIIR445
DOI: https://doi.org/10.52731/liir.v006.445

Evaluating and Enhancing RAG Systems through Test and Source Analysis

Zelan Shi [*], Oranus Kotsuwan [†],
Kazunori Matsumoto [*]

Abstract

This paper presents a prototype Retrieval-Augmented Generation (RAG) system developed for university curriculum guides and evaluates its performance through experiments. RAG, which combines large language models (LLMs) with independent information sources, is emerging as a solution to address generative AI challenges such as hallucinations and the lack of domain-specific knowledge. By prioritizing information from dedicated databases, RAG can enhance factual accuracy and reduce hallucinations. Through experimental trials, the system demonstrated reliable performance in some cases, although issues related to the quality of information sources and data extraction were identified. These findings underscore the importance of robust testing and systematic revisions of information sources. This paper reports on an outline of the system implementation, the guides for improvement, and the experimental results. We find that an iterative improvement process is crucial for enhancing the overall quality of RAG. This process involves not only optimizing retrieval and generation mechanisms but also continuously reviewing and refining the information sources themselves, the system can systematically adapt to ensure sustained relevance and improved response accuracy over time.

Keywords: Generative AI, RAG, white box test, black box test

1 Introduction

Recent advancements in LLMs and generative AI have spurred significant research into systems capable of automated question answering and conversational dialogues [1][2]. While they offer clear benefits, such as increasing efficiency, reducing response time, and enabling personalized interactions, several challenges remain. One major challenge is ensuring the models' reliability and fairness, particularly in handling diverse user inputs [3][4]. Techniques such as fine-tuning [5, 6] with domain-specific data, and employing RAG for improving factual accuracy have been explored to address these issues [7][8][9].

Among these approaches, this study focuses specifically on RAG. Unlike fine-tuning, RAG does not require additional training of LLMs. This makes RAG a cost-effective solution that can be implemented relatively easily. RAG is based on a retrieval mechanism that selects relevant information from external databases and uses it to create answers or responses. It however has certain limitations, such as missing or noisy retrieval results. Additionally, errors can occur during the generation process, where irrelevant or hallucinated information may be included

[*] Kanagawa Institute of Technology, Kanagawa, Japan
[†] Thammasat University, Bangkok, Japan

despite accurate retrieval. To overcome these issues, several studies have explored improvements to RAG [7][8][9][10][11][12][13] such as refining retrieval methods to reduce noise and increase accuracy, integrating filters to ensure the relevance of retrieved data. Among these we focus on testing methods test data generation for RAG.

Testing relies on test data, which serves as the input for the system during the evaluation process [10] [11]. The quality of the test data directly affects the quality of the testing. High-quality test data should meet certain criteria: it should represent diverse and realistic scenarios that users might encounter, and it must be accurate, complete, and free from bias to ensure reliable results. Carefully designed test data helps identify errors and weaknesses in the system, guiding improvements and enhancing its robustness. Recent methods [11][12][13] for test data generation using LLMs involve extracting segments of the information source and adding additional guiding information to assist the generation process. These methods leverage the flexibility of LLMs to produce diverse outputs, they however often fail to fully utilize the characteristics of the extracted information segments, which limits their effectiveness

Implementing a basic RAG system is relatively straightforward; however, its performance is often insufficient for practical applications. Achieving a reliable and usable system requires iterative testing and continuous improvements. This paper outlines the development and testing of a prototype RAG system, offering insights that may serve as a valuable reference for other developers working on similar systems.

2 Prototyping a RAG System for Curriculum Guidance

We developed and tested a prototype RAG system designed to assist with university course guidance. As the number of courses grows and curricula become increasingly complex, many universities face challenges in helping students navigate course materials effectively. While some institutions provide FAQ services online, these systems rarely offer detailed answers. Due to the diversity among universities, general LLMs struggle to address specific needs, making RAG a promising solution in this domain.

2.1 Outline of System

Our system has a straightforward architecture as shown in Figure 1. The information source provides data independent of the LLM, and forms the basis for RAG processing. We used here a PDF version of a university course guide [14] in Japanese. A text loader extracts text from the information source, excluding non-text elements like tables or images. For the experiment, we apply preprocessing to remove obvious errors and noise. However, other common text processing techniques, such as stop-word removal and morphological analysis, are not performed. Instead, raw text data is used in the experiment without additional modifications. Figure 2 shows a part of course guide, simplified and translated from Japanese using ChatGPT.

The extracted text is divided into smaller segments called chunks using a text splitter. The chunking method is a key design parameter that affects RAG performance. Each chunk serves as the main source of information for generating answers to system queries. To support search processes, each chunk is embedded into vectors [1] and stored in a vector database. The embedding algorithm and vector dimensions influence search efficiency and, ultimately, RAG

performance. These preprocessing steps are completed before the query-answering process begins. The query-answering workflow, shown with dashed arrows in Figure 1, starts with users entering questions in English. The input questions are embedded using the same method as the

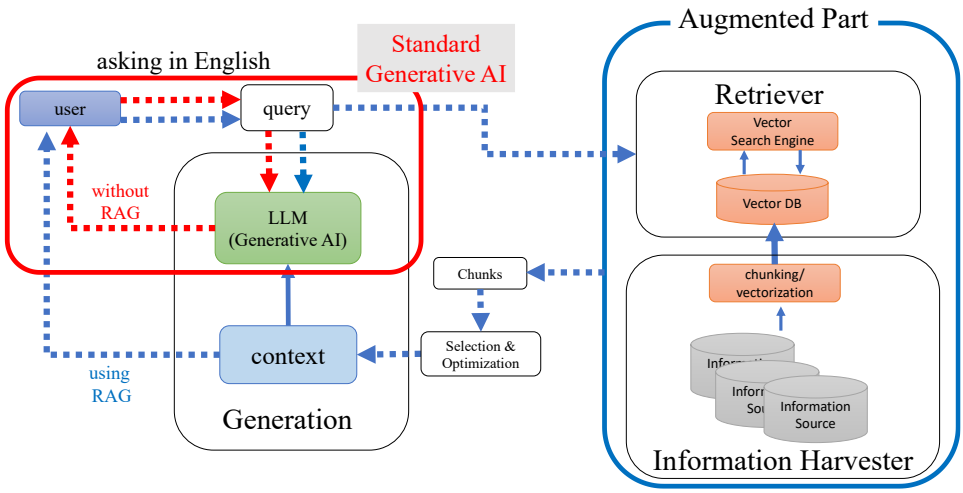

Figure 1: Outline of RAG system

source chunks. The system then performs similarity-based approximate searches [15] in the vector database, retrieving the top n most similar chunks. Parameters like the search algorithm and the value of n significantly impact RAG's overall performance. The retrieved chunks serve as the context in prompts provided to the LLM to generate the final output.

Introduction to Programming
It is recommended to have completed Information Literacy.
This course serves as an introduction to fundamental concepts of programming that are applicable across all programming languages. Students will learn the basics of computational processing and control, gaining practical experience through lectures and hands-on exercises.
The course utilizes Visual Studio as a beginner-friendly development environment, enabling students to explore programming concepts using real-world examples. By working on familiar topics, students will develop a deeper understanding of problem-solving through programming.
Key Learning Topics:
(1) Fundamentals of program creation
(2) Data input, processing, and output
(3) Variables and data types
(4) Operators and expressions
(5) Control structures
(6) Functions and modular programming
(7) Graphics programming and visualization

Figure 2: Small Part of Course Guide

2.2 Black-Box and White-Box Testing Approaches for RAG

The evaluation of RAG systems requires high-quality test data and reliable metrics, which have been the subject of active research [13][14]. Testing methodologies are broadly classified into two types: black-box testing and white-box testing [16]. In black-box testing, the internal mechanisms of the system remain unknown to the tester, whereas white-box testing leverages internal system knowledge for evaluation. This study adopts a similar classification for testing RAG systems, utilizing both black-box and white-box approaches. The black-box method relies on persona-driven user analysis, where evaluations are conducted without direct reference to the

information source. Instead, test cases are designed based on typical user behaviors and characteristics. In contrast, the white-box approach assesses RAG outputs by comparing responses against the information source. The primary objective is to determine whether the system-generated answers align with predefined data within the source material. Since RAG systems operate by processing queries and generating responses, their evaluation is fundamentally centered around question-answer verification. Consequently, test data consists of structured question-answer pairs.

2.2.1 Persona Based Black-Box Test Data Generation

For the development and evaluation of test data, a semi-automated persona model approach was employed, a method commonly used in fields like market analysis and customer behavior research [16]. Personas help model typical user behaviors, needs, and inquiries, enabling the creation of more tailored system responses and enhancing usability. For instance, persona models include various student types, such as first-year students unfamiliar with course registration, students prone to registration errors due to lack of understanding, and senior students well-versed in the process. Beyond students, personas also encompass academic staff and professors who provide guidance and advice. Each persona encapsulates distinctive characteristics, behaviors, and inquiry patterns.

Table 1: Course Selection Criteria and Student Perspectives

Criteria	Low Ability & Motivation Students	High Ability & Motivation Students
Ease of Completion	Prefer easy courses with minimal assignments & exams. *Which courses have the least amount of homework and tests?*	Seek intellectually stimulating courses. *What advanced concepts will this course cover beyond the basics?*
Schedule Convenience	Choose classes based on preferred timing. *Are there any afternoon-only classes?*	Optimize schedules to balance commitments. *How does this course fit into an optimal study plan?*
Professor's Leniency	Favor lenient grading & attendance policies. *Which professors are known for easy grading?*	Consider professor's expertise & teaching quality. *What research has the professor contributed to this field?*
Peer Influence	Follow friends' recommendations. *Which courses are popular among students?*	Select based on academic & career goals. *What courses are best for serious students in this field?**
Effort Required	Prefer courses with minimal readings & simple assessments. *Does this course require a lot of reading and assignments?*	Accept challenging coursework for deeper learning. *What types of projects are included in this course?*
Graduation Requirements	Focus on earning necessary credits easily. *Will this course help me graduate smoothly?*	Select courses strategically for academic growth. *How does this course support my career development?*
Difficult Subjects	Avoid analytical/problem-solving courses. *Are there any courses that don't require complex thinking?*	Embrace rigorous subjects for intellectual expansion. *Will this course enhance my critical thinking skills?*
Extracurricular Balance	Prioritize free time for personal activities. *Does this course leave enough time for my part-time job?*	Balance workload with research or internships. *Will this course provide networking opportunities?*
Familiar Topics	Stick to comfortable & known subjects. *Is this course similar to what I studied before?*	Explore new and complex topics for growth. *Will this course introduce groundbreaking ideas?*
Flexible Learning Options	Prefer online or relaxed attendance courses. *Can I take this course online with flexible attendance?*	Opt for interactive & engaging learning environments. *Does this course promote discussion-based learning?*

Table 1 provides a structured comparison of two student persona types, illustrating their distinct approaches to course selection. To improve readability, each criterion is accompanied by a representative question (italicized) that reflects the typical concerns of students within each category. As the table demonstrates, student perspectives vary significantly depending on their academic ability and motivation, leading to fundamentally different approaches to course selection and inquiry generation.

2.2.2 Chunk Based White-Box Test Data Generation

White-box testing involves generating test data directly from the text within an information source. For each chunk Ci, a set of corresponding questions Q is generated using a LLM. Eaxg question in Q is designed to be closely related to the content of Ci, ensuring a high likelihood that the answer can be derived from the chunk itself.

Table 2 presents a subset of the questions generated from the course description shown in

Table 2: Examples of Generated Questions

Type	Question
Who	Who is recommended to take the Introduction to Programming course?
What	What is the main purpose of the Introduction to Programming course?
When	When is it recommended to take the Introduction to Programming course?
Where	Where can students gain practical experience in programming during this course?
Why	Why is Visual Studio used in this course?
How	How does the course help students understand problem-solving through programming?
What	What are the key learning topics covered in this course?
Who	Who benefits from using Visual Studio in this course?
Why	Why is problem-solving emphasized in this course?
How	How does the course introduce programming concepts to students?

Figure 2. As indicated in the table, the question generation process is structured around the 5W1H framework—Who, What, When, Where, Why, and How—which provides a systematic approach to inquiry formulation. Unlike the black-box method outlined in Table 1, which focuses on persona-driven question generation, this approach produces a distinctly different set of questions. The incorporation of 5W1H principles enhances the diversity of generated questions, ensuring a broader range of inquiry types. Additionally, more refined control over question generation can be achieved by integrating detailed prompts and predefined templates [11][12]. These techniques allow for greater precision in shaping the structure and focus of generated questions, further improving the effectiveness of the testing process.

2.2.3 Experiment Evaluation

For evaluation, we created a dataset of 100 questions of both black-box and white-box types, including ones shown in Table 1 and 2. In standard test data generation, a test dataset typically consists of question-answer pairs, where each question is paired with a predefined correct response. However, in this experiment, only the questions were generated using the previously described methods, without preparing the correct answers in advance. In order to evaluate the system's performance, these questions were then fed into the RAG, then the responses were subsequently reviewed and assessed by human evaluators, who determined their validity and accuracy. Regarding the evaluation criteria in this stage, responses are considered incorrect if they (a) fail to address the query or respond with "unknown" to questions that can be answered; (b) contain factual errors; (c) include unnatural or incoherent text. Responses that do not meet these criteria are deemed incorrect, while those that adhere to them are considered correct.

The evaluation results for 100 test queries, based on the predefined criteria, are summarized in Table 1. Cases A–D involve varying chunk sizes and separators used during processing.

Regardless of the configuration, the accuracy rate remained consistently low at approximately 40%, indicating significant limitations in the system's performance. As shown in the table, changing the chunk size does not lead to significant variations, and the accuracy remains low. The chunk size used in this experiment was determined based on the typical length of course descriptions in the current enrollment guide.

Table 3: Accuracy across Cases

Case	Chunk Size	Separators	Accuracy (Correct / Total)
A	100	. \n	0.40
B	300	. \n	0.42
C	100	.	0.44
D	300	.	0.39

Table 3 examines the reasons for errors observed across the experiments (Cases A–D). In this table, the column "Title Error" involves incorrect course titles, where the LLM generated responses about non-existent courses. This is classified as hallucination, a known issue in generative AI systems. One of the objectives of RAG is to prevent hallucinations; however, as shown by this data, that objective has not been achieved. "Insufficient Source Data" column refers to cases where the original information source provided incomplete or ambiguous explanations, making misunderstandings by the system unavoidable. "Noise" shows errors occur due to the inclusion of noisy or irrelevant characters in the text, often introduced during the text extraction process from the source materials. "Unknown Cause" includes errors where the exact reason for the incorrect response could not be identified, despite the output being clearly inaccurate. In summary, errors occur in both the retrieval stage and the generation stage, with some cases where issues arise due to the interaction between the two. Additionally, some errors stem from insufficient textual information provided in the source data.

Table4: Causes of Errors

Case	Title Error	Insufficient Source Data	Noise	Unknown Cause
A	15	17	9	19
B	14	22	7	15
C	17	15	5	19
D	14	19	9	19

2.3 Iterative Source Revision

The analysis of experimental results indicates that some errors originate from deficiencies in the information source. Addressing these issues can lead to improved accuracy in system responses. Several key scenarios must be considered when refining the information source.

Insufficient Information in the Source: in cases where the source contains incomplete or missing information, errors arise due to the lack of essential details. A straightforward solution is to augment the source with additional data. However, excessive inclusion of minor, non-essential details can reduce readability for human users. To mitigate this, supplementary information can be incorporated as metadata, ensuring clarity while maintaining accessibility.

Fragmentation Due to Chunking: another issue stems from the chunking process, where relevant information is split across multiple segments. When retrieval is limited to a single chunk, critical details may be missing, preventing the RAG system from generating accurate responses. One possible solution is to introduce redundancy through metadata, ensuring that essential information remains accessible across multiple chunks.

Iterative Refinement Process: based on evaluation results, the information source undergoes continuous revision to enhance system performance. This iterative process ensures that errors identified during testing lead to systematic improvements in data structure and retrieval mechanisms. The implementation of this refinement strategy is currently in progress, with Figure 3 illustrating a conceptual overview of the approach.

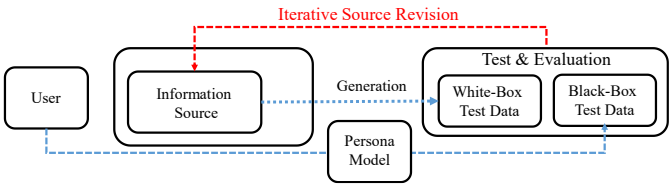

Figure 3: Iterative Source Revision

3 Conclusions

RAG is a promising technology with accelerated development and growing interest. However, issues related to reliability, such as error occurrence, have also surfaced. This has led to active research into techniques for improving reliability, including testing methodologies. This paper reports on the development of a prototype RAG system designed for application in university curriculum guides. As demonstrated by the experimental results, it is evident that the current system's accuracy is insufficient for practical deployment. While some issues may be addressed by advancements in LLM performance, this alone is not enough to achieve the desired reliability. Efforts are currently underway to improve accuracy while simultaneously developing advanced testing techniques to enhance system robustness and reliability. Future work also involves larger-scale validation experiments to further assess the method's robustness.

References

[1] S. Bengesi, H. El-Sayed, MD K. Sarker, Y. Houkpati, J. Irungu, T. Oladunni, Advancements in Generative AI: A Comprehensive Review of GANs, GPT, Autoencoders, Diffusion Model, and Transformer IEEE Access, vol. 12, pp. 69812-69837, 2024.

[2] L.Manduchi, K.Pandey, C.Meister, et al., On the Challenges and Opportunities in Generative AI, arXiv:2403.00025, 2024.

[3] S.S. Sengar, A.B.Hasan, S.Kumar, et al., Generative Artificial Intelligence: A Systematic Review and Applications, Multimedia Tools and Applications, Springer, https://doi.org/10.1007/s11042-024-20016-1, 2024.

[4] K.B. Ooi,, G.W. H.Tan, et al., The Potential of Generative Artificial Intelligence Across Disciplines: Perspectives and Future Directions, Journal of Computer Information Systems, Vol. 65, No.1, 2023.

[5] V. B. Parthasarathy, A. Zafar, A. Khan, A. Shahid, The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities, arXiv:2408.13296, 2024.

[6] R.Patil,and V. Gudivada, A Review of Current Trends, Techniques, and Challenges in Large Language Models (LLMs), Applied Sciences, Vol.14, No.5, 2024.

[7] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T. Chua, Q. Li, A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models, Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'24), 2024.

[8] M. Arslan, H. Ghanem, S. Munawar, C. Cruz, A Survey on RAG with LLMs, Procedia Computer Science, Vol. 246, pp. 3781-3790, 2024.

[9] S. Gupta, R. Ranjan, S. N. Singh, A Comprehensive Survey of Retrieval-Augmented Generation : Evolution, Current Landscape and Future Directions, arXiv:2410.12837, 2024.

[10] X. Huang, W.Ruan, W.Huang, et al. A survey of safety and trustworthiness of large language models through the lens of verification and validation, Artificial Intelligence Review, Vol. 57, No.175, 2024.

[11] S. Filice, G. Horowitz, C. David, Generating Diverse Q&A Benchmarks for RAG Evaluation with DataMorgana, arXiv:2501.12789, 2025.

[12] S. Gupta, C. Berrospi, L. Mishra, M. Dolfi, P. Staar, P. Vagenas, Know Your RAG: Dataset Taxonomy and Generation Strategies for Evaluating RAG Systems, arXiv:2411.19710, 2024.

[13] V. Jeronymo, L. Bonifacio, H. Abonizio, M. Fadaee, R. Lotufo, J. Zavrel, R. Nogueira,Inpars-v2: Large Language Models as Efficient Dataset Generators for Information Retrieval, arXiv:2301.01820, 2023.

[14] Kanagawa Institute of Technology, https://portal.kait.jp/aaa_web/KAIT_WEB/004_curriculum/curriculum.html

[15] https://ai.meta.com/tools/faiss/

[16] S.Kirchem, and J.Waack,Explore the Right Personas for Successful Marketing, Sales, and Service, In: S. Kirchem, M. Stadelmann, M.Pufahl, D. Laux, (eds) CRM Goes Digital. Management for Professionals.

[17] R.Patton, Software Testing, 2nd edition, Sams Publishing, 2005.