Cureus Journal of Computer Science

Open Access Original Article

Information Retrieval System for Automating Quiz Generation and Evaluation Using Large Language Models

Received 04/30/2025
Review began 06/10/2025
Review ended 09/04/2025
Published 09/08/2025

DOI:
https://doi.org/10.7759/s44389-025-05957-4

Gayatri Kurulkar [1] , Sakshi Ingale [1] , Advait Shinde [1] , Yash Jangale [1] , Suhasini Itkar [1]

1. Computer Engineering, PES Modern College of Engineering, Pune, IND

Corresponding author: Gayatri Kurulkar, gayatrikurulkar@gmail.com

Abstract

Artificial intelligence (AI) is playing an increasingly transformative role in education by improving how content is delivered, retrieved, and adapted to learners' needs. This research presents the design and development of an automated quiz generation and educational question-answering system that combines Retrieval-Augmented Generation with safety mechanisms guided by Guardrails AI, a framework for enforcing content quality and ethical constraints in generative models. The system integrates large language models such as Generative Pre-trained Transformer 4 with LangChain pipelines to retrieve semantically relevant answers from academic documents and generate pedagogically appropriate quiz questions. Tested on 50 university-level queries, the system achieved 92% answer accuracy, 95% semantic relevance, and 96% coherence. These results highlight the potential of combining generative AI with structured retrieval and safety checks to build reliable and responsible educational tools. This work contributes to the growing field of AI in education and opens pathways for future development of scalable, safe, and adaptive learning support systems.

Categories: AI applications, AI Ethics and Fairness, Machine Learning (ML)
Keywords: large language models (llms), retrieval-augmented generation (rag), automated quiz generation, fine-tuning, guardrails, human in the loop (hitl)

Introduction

The rapid innovation of Generative AI has revolutionized educational technology, offering unprecedented opportunities for automating and personalizing learning experiences. One of the most time-consuming and cognitively demanding tasks in education is the creation of high-quality quiz content that accurately aligns with teaching materials. Educators often face the challenge of crafting relevant and effective questions, a process that can be labor-intensive and subjective. To address this challenge, this automated quiz generation and evaluation system has been proposed for generating quizzes, leveraging advanced techniques in Generative AI, specifically Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG). This solution aims to bridge the gap in content assessment, providing scalable and intelligent support for educators while enhancing learning outcomes [1]. The automated system developed integrates LLMs with Facebook AI Similarity Search (FAISS)-based semantic search and LangChain pipelines, creating an industry-leading solution designed to cater to dynamic learning environments. These technologies are chosen for their ability to process and interpret unstructured content, such as PDFs and other teaching materials, and generate semantically accurate questions in response to the content. The system also adapts in real time based on teacher feedback, ensuring that the generated questions are both contextually relevant and pedagogically sound. Unlike traditional rule-based systems, which are rigid and unable to account for nuances in teaching materials, our solution is dynamic and can continuously improve over time as it learns from user input and feedback [2,3].

This automated system is a modular pipeline that takes teaching material as input, performs contextually appropriate retrievals using FAISS-based semantic search, and generates domain-relevant quiz questions via LLMs. These generated questions are then verified by educators to ensure they meet the pedagogic standards required for effective assessment [4]. The modular design not only reduces the manual workload for teachers but also increases the precision and quality of quiz content, thus enhancing the overall learning experience. This solution addresses several interconnected challenges in educational technology, including document understanding, semantic search, probabilistic text generation, and adaptive test design. The problem at hand requires a multidisciplinary approach, drawing upon methods from natural language processing, machine learning, and cognitive science. As part of the emerging field of AI-based education, this research aims to democratize intelligent tutoring, providing open-source tools that empower educators and learners alike. The theoretical underpinnings of this system rest on solid foundations in linear algebra for vector space models, complexity theory, and probability theory, ensuring the validity of the methods and their applicability in real-world deployment [5,6].

Building on this foundation, this work presents a practical and theoretically grounded approach to

addressing the challenges of quiz generation in educational settings, offering a scalable and adaptive solution that leverages the latest advancements in AI to improve both the efficiency and effectiveness of content assessment in learning platforms.

The main contributions of this research are development of an end-to-end automated quiz generation system that combines LLMs with semantic search (FAISS) in a modular architecture, integration of teacher feedback into the generation loop, enabling real-time adaptability and continuous improvement, a demonstration of how open-source AI tools can be used to create pedagogically sound and scalable assessment content.

Materials And Methods

Literature review

The incorporation of LLMs in retrieval, generation, and safety operations has drastically revolutionized the conventional information systems. This literature review consolidates pivotal developments in retrieval architectures, generative evaluation tasks, safety procedures, and learning uses, situating them against our proposed GPT-4-based quiz generation system, which uses FAISS, LangChain, and Guardrails AI.

Retrieval Architectures and RAG

The idea of RAG has gained much popularity in recent years, with various frameworks being put forward to improve LLM performance on information retrieval tasks. Different sources referred by this proposed system provide an extensive survey on the application of pretrained LLMs to dense text retrieval, emphasizing the advantages of LLM-based retrieval systems in modeling semantic relationships within large-scale data [7,8]. This is different from conventional keyword-based retrieval systems that are based on exact matches, providing more subtle and contextually accurate results. Likewise, Chen et al. [1] benchmark the use of LLMs in RAG, demonstrating how they can produce high-quality responses by retrieving and enriching information. Contrarily, this system utilizes FAISS for retrieval and indexing but uses GPT-4 for question generation with specific attention to academic settings to maintain accuracy and pedagogical usefulness.

Summarization vs. Generative Assessment Tasks

Although summarization tasks have experienced significant advancements with LLMs, they continue to grapple with challenges such as hallucination and content drift. Discussions on process-oriented summarization have been referenced by this automated quiz generator, which asserts that although LLM-based summarizers provide enhancements in fluency and contextual relevance, problems such as incoherence remain [4,5]. This proposed system diverges from standard summarization by employing semantic chunking and contextual retrieval to produce controlled output aimed at constructing questions, keeping the content curriculum-aligned and factually accurate.

Guardrails and Safety

Safety issues like hallucinations and toxic outputs are still key challenges when using LLMs. This proposed system incorporates Llama Guard, an LLM classifier model that detects and prevents unsafe prompts, especially in sensitive situations. Furthermore, researchers have built on this with R2-Guard, which uses logical reasoning to enhance robustness and reduce harmful outputs. Equivalently, our system combines soft guardrails and prompt-based filters tailored to educational environments. Such methods enable us to balance safety with contextual precision necessary for optimal quiz generation and ensure that outputs are aligned with the desired pedagogic goals.

Use in Education and LLM Reliability

The educational use of LLMs and their reliability are significant factors when integrating them into real-world applications. Greengard [7] tackles the issue of hallucinations, classifying various forms and discussing approaches to their alleviation in learning settings. These works emphasize the necessity of formal validation procedures when using LLMs in educational systems. On top of this, various existing systems talk about the intricacies involved in real-world deployment of LLMs and how our process of applying Pydantic validation and teacher-in-the-loop approaches ensures accuracy, structure, and relevance in the generated questions.

Methodology

Tools and Technologies

LLMs - GPT-4: At the heart of the system proposed is GPT-4, a transformer LLM capable of generating

intelligent and contextually aware quizzes from course content. GPT-4's probabilistic token prediction process allows for human-like language generation, making it ideally placed to generate varied question types in a variety of forms, such as multiple choice, true/false, and short answer questions. This is a major step forward from rule-based systems, which are generally rigid and narrow-scope. The system utilizes prompt-based control, where GPT-4 produces questions from templated structures consisting of the topic, instructional context, keywords, and target question format-enabling flexible and dynamic content creation [1,8].

By incorporating GPT-4 in a RAG framework, the system guarantees that question generation is grounded in semantically appropriate document segments retrieved through FAISS [9]. This structure considerably improves content alignment with the original material, enhancing accuracy and pedagogical fidelity. In addition, GPT-4's capacity to regenerate content iteratively based on feedback is particularly useful in educational environments, where teachers might want to adjust questions on the basis of clarity, relevance, or level of difficulty [3,10]. This feedback-based adaptability allows instructors to refine the produced tests in real time, thus improving instruction quality and learning engagement.

LangChain: LangChain is the orchestration layer of the system that facilitates smooth interaction between the Large Language Model (GPT-4) and the knowledge base. It facilitates harmonious integration between the retrieval and generation parts of the pipeline. By taking advantage of LangChain's modular design, the system dynamically creates prompts for very specific topics, keywords, and question structures-essential for gaining contextual accuracy while generating quizzes. LangChain is closely coupled with the FAISS vector store to enable semantically informed retrieval of the most pertinent academic material consistent with the quiz context [9]. This enables GPT-4 to generate highly pertinent and pedagogically valid questions based on accurate source material. In addition to prompt management, LangChain further controls end-to-end data passage between FAISS and GPT-4 so that it can be adaptive with minimal latency throughout the pipeline. Its programmability and flexibility enable it to become a crucial building block in synchronizing the intricacy of real-time AI-powered quiz generation.

Guardrails AI: Guardrails AI is integrated into the system as a supervisory framework to ensure ethical compliance, safety, and domain-appropriate content generation. Acting as a critical control layer, it validates input prompts, keywords, and generated outputs for syntactic integrity and semantic relevance [9]. This framework effectively mitigates hallucinations, toxicity, and the emergence of non-academic content in GPT-4's responses, enhancing both factual reliability and pedagogical soundness [4]. Furthermore, Guardrails AI enforces user-defined constraints, such as question format preferences or subject alignment, ensuring that the generated quizzes remain tailored, accurate, and context-aware. By embedding these safety mechanisms, the system promotes responsible AI use, fairness, and transparency in educational applications. This layer of control enhances system reliability without stifling the creative flexibility and personalization that GPT-4 enables in automated quiz generation.

Real-time feedback and teacher-in-the-loop: To improve quiz quality on a continuous basis, the system features a teacher-in-the-loop facility, whereby instructors can offer feedback on the automatically generated questions in real time. This iterative refinement of questions, by teachers and approval thereof, ensures that the final output conforms to the targeted educational goals. This feedback cycle enhances the pedagogical suitability and contextual fidelity of the quizzes, rendering them more appropriate for varied classroom settings [10-12].

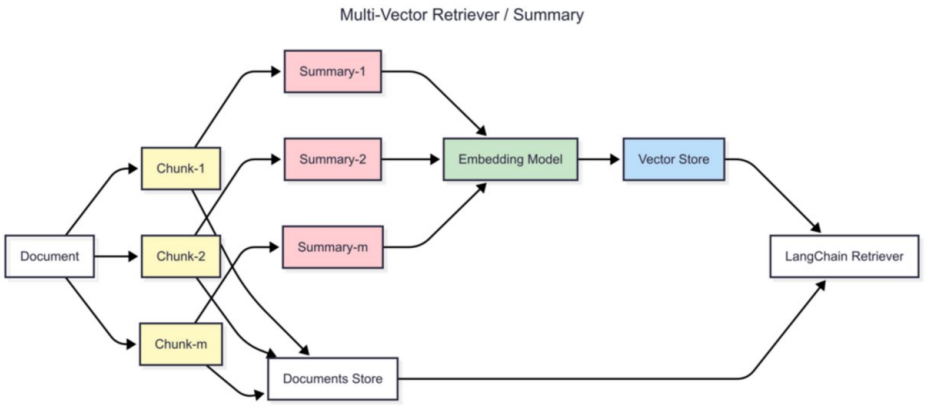

FIGURE 1: Chunk Retriever

Source: TeeTracker: Semantic chunking of academic content before embedding and retrieval (https://teetracker.medium.com/llama-index-multi-vector-retriever-summary-9231137d3cab)

Figure 1 illustrates how academic content is broken into semantic chunks for embedding and storage in the FAISS vector database, enabling efficient retrieval during quiz generation. Figure 2 depicts the overall RAG architecture, showing the interaction between document chunks, retrieval, GPT-4 generation, and final quiz output. Figure 3 shows how Guardrails AI is integrated to validate outputs, ensure safety, apply constraints, and manage prompt inputs to prevent hallucinations or irrelevant content.

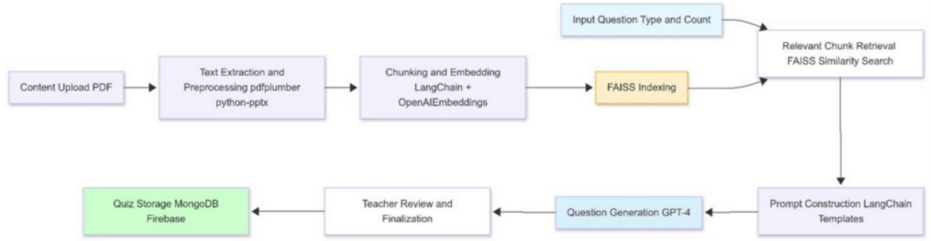

FIGURE 2: RAG Pipeline

RAG pipeline showing the flow of document retrieval and quiz generation. Adapted from OpenAI's RAG documentation

Developed by the authors of this paper, with elements adapted from Fan et al. [9].

FAISS, Facebook AI Similarity Search; RAG, Retrieval Augment Generation

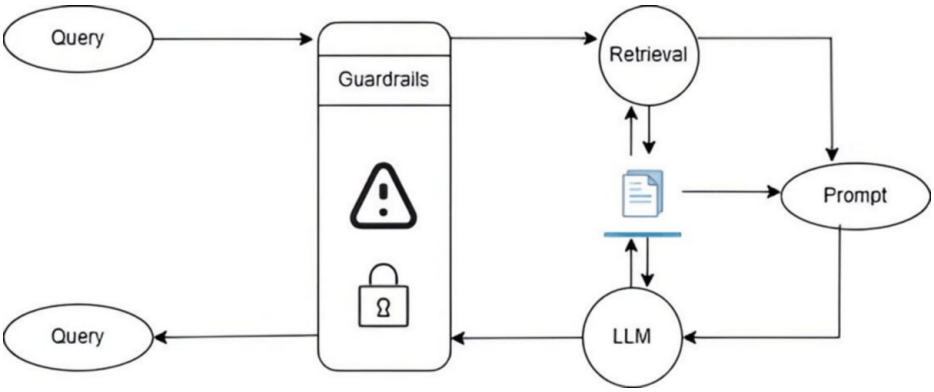

FIGURE 3: Guardrails

Guardrails AI framework integrated for prompt filtering and content validation to ensure safe and relevant outputs

Developed by authors

LLM, Large Language Models

System Architecture

The architecture follows a modular pipeline integrating Generative AI, and MongoDB for automated quiz generation from educational documents.

1. Input & Prompt Engineering: A structured prompt with few-shot examples is crafted and fed into a RAG pipeline using LangChain and FAISS to fetch relevant content from a vector database.

2. LLM Processing: The retrieved context and prompt are passed to a LLM, which generates quiz content in JSON format conforming to a predefined schema (Quiz → Question → Option Model), validated using Pydantic.

3. Validation and Feedback: If valid, the output is post-processed and stored. If invalid, reinforcement feedback is given or human-in-the-loop validation is triggered.

4. Storage and Deployment: After passing validations, the quiz is stored in MongoDB and made available for deployment.

5. Interface: A Streamlit-based UI facilitates user interaction for both input and quiz access.

Reinforcement feedback mechanism: The system incorporates a feedback loop where invalid LLM outputs-detected via Pydantic validation-trigger prompt refinement or re-generation. Minor issues are handled through automated prompt updates, while persistent errors invoke a HITL interface for manual correction. This ensures quiz quality, consistency, and learning adaptation. Over time, it helps fine-tune prompt templates for better model responses. The mechanism enhances reliability with minimal human oversight.

Security and guardrail: To ensure safe and structured content generation, the system uses Pydantic for strict JSON schema validation, preventing malformed or hallucinated outputs. It employs regular expression filters to detect unsafe language and mitigates prompt injection by sanitizing inputs. Validation logs enable auditing and debugging. Only verified quizzes are stored in MongoDB and exposed via the UI. These guardrails ensure data integrity and system trustworthiness.

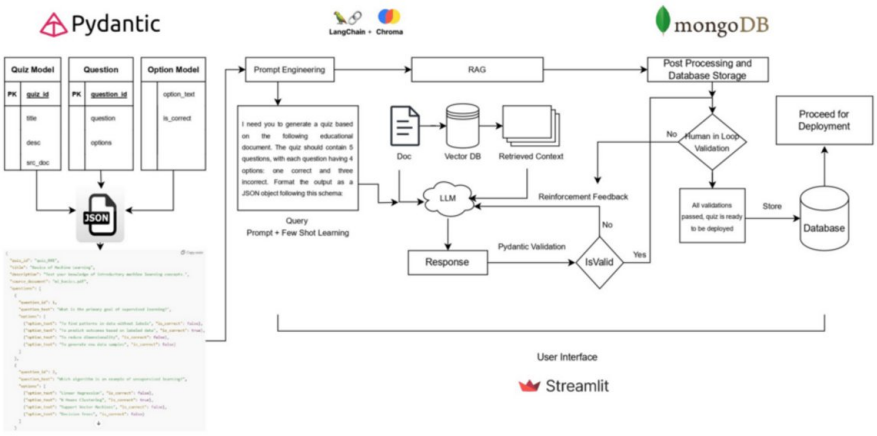

FIGURE 4: System Architecture

Modular architecture showing input processing, retrieval, GPT-4 generation, validation, and storage.

Developed by authors

Figure *4* visualizes the full modular system including input ingestion, prompt engineering, semantic retrieval via FAISS, LLM processing, validation, storage in MongoDB, and user interaction via Streamlit UI.

Results

Experimental setup

The proposed GPT-4-based RAG system was tested using a modular pipeline that integrates GPT-4, FAISS for semantic retrieval, and LangChain for orchestration [13,14]. Academic documents were embedded using the text-embedding-ada-002 model and stored in FAISS, enabling context-aware generation. LangChain managed prompt flow, while outputs were validated using Pydantic and filtered by Guardrails AI to ensure safety and relevance. The dataset included academic PDFs chunked for efficient retrieval. Performance was evaluated on factual accuracy, faithfulness, relevance, latency, output speed, and schema compliance. The system was benchmarked against various RAG and non-RAG baselines using 100 academic queries. Deployment was done via a Streamlit UI on a CPU VM using Python 3.9, FAISS, LangChain, and the OpenAI SDK.

System performance

The proposed GPT-4-based RAG system was evaluated using a modular pipeline that integrates OpenAI's GPT-4, FAISS for vector storage, and LangChain for orchestration. GPT-4 served as the core language model, generating responses grounded in context retrieved from semantically indexed academic content. Document embeddings were created using the text-embedding-ada-002 model and stored in a FAISS vector store, enabling high-recall retrieval. LangChain managed the flow of prompts, retrieval, and generation to ensure modular and efficient processing. To maintain output quality and safety, each response was validated against a predefined schema using Pydantic. Guardrails AI was integrated to detect

and filter hallucinations, toxic language, or non-educational content, providing a controlled generation environment. Outputs failing schema validation were either regenerated automatically or sent for human-in-the-loop (HITL) review. The input dataset comprised academic PDFs and structured course material, segmented into overlapping chunks of approximately 500 tokens to maximize retrieval performance across varied academic domains. System performance was measured using metrics such as factual accuracy, faithfulness to source, semantic relevance, response latency, output speed, and compliance with the output schema. Evaluations were conducted through a combination of system logs and expert human assessment. The system was benchmarked against multiple baselines, including a direct GPT-4 setup without retrieval, BM25 + GPT-4, a hybrid RAG model combining semantic and keyword retrieval, Mistral 8×7B Super RAG, Claude 3 RAG, and Vertex AI integrated with Gemini 1.5 Pro. All models were tested on a common set of 100 academically oriented queries for comparative analysis. The platform was deployed using a Streamlit-based user interface hosted on a CPU virtual machine, with the software stack comprising Python 3.9, LangChain, FAISS, and the OpenAI SDK.

Performance evaluation

The automated quiz generation system was evaluated on multiple critical dimensions including factual accuracy, response latency, hallucination minimization, and overall relevance. Table *1* summarizes the results achieved by the system using standardized benchmarks.

Metric	Description	Value	Benchmark/Dataset Used
Factual Correctness	Percentage of factually accurate responses	85%	MMLU Benchmark
Relevancy	Relevance of generated responses to context	80%	Custom RAG Evaluation Dataset
Hallucination Rate	Percentage of responses containing false information	12%	Hallucination Detection Dataset
Response Time (RAG)	Average time to generate a response in RAG setup	<4 seconds	System Logs
Output Speed	Tokens processed per second	64.8	Performance Benchmarks
Latency (First Token Time)	Time taken to deliver the first token	0.5 seconds	Performance Benchmarks
Quality Index	Overall quality score from different evaluations	71	Quality vs. Price Analysis

TABLE 1: Performance Evaluation of the Proposed System

MMLU, Massive Multitask Language Understanding; RAG, Retrieval Augment Generation

Table *1* summarizes key evaluation metrics such as factual correctness (85%), relevance (80%), hallucination rate (12%), latency, and quality index for the GPT-4-based RAG quiz generation system. RAG systems combine a vector database and retriever with LLM to ground generation in external knowledge. Production-grade RAG offerings include OpenAI's GPT-4 RAG (using OpenAI embeddings + e.g. FAISS), Anthropic's Claude 3 RAG (with Claude 3 LLM and "contextual retrieval" embeddings plus BM25), Google Vertex AI RAG (Gemini 1.5 LLM with Google's embedding/index or third-party stores), and AWS Bedrock (Titan Text Premier with Amazon Knowledge Bases). Cutting-edge research prototypes include open-source pipelines (e.g. Meta's Llama 3 with Elastic/Chroma) and advanced RAG architectures (e.g. Mistral's "Super RAG" with Mixtral 8×7B). Table summarizes representative systems, listing their vector store, retriever type, and LLM, along with key metrics.

System (Retriever + Vector DB)	Retriever Model	LLM	Retrieval Precision	Retrieval Recall	Gen. Faithfulness	Latency	Cost	Flexibility
Mistral 8×7B “Super RAG”	Cache-based retriever (tuning-fork cache)	Mixtral 8×7B (SMoE)	≈0.76	≈0.69	≈0.80	≈1,000–2,000 ms	Moderate	Low
Anthropic Claude 3 RAG (Voyage/ Gemini + BM25)	Contextual embeddings + BM25 + cross-encoder (Cohere)	Claude 3 (Opus/Sonnet)	~0.80 (very high)	~0.75 (very high)	≈0.85	≈1,000 ms	High	Low (closed)
Google Vertex AI RAG (Vertex Vector)	Dense embed (PaLM/ Gemini), optional cross-ranker	Gemini 1.5 Pro	≈0.75 (top models)	≈0.70	≈0.85	≈500–1,000 ms	High	High
AWS Bedrock Titan Text Premier	Dense embed(TitanEmbed) + sparse search (Knowledge Base)	Titan Text Premier	~0.75 (est.)	~0.70 (est.)	≈0.85	≈1,000 ms	Moderate–High	High (finetunable)
OpenAI GPT-4 RAG (OpenAI embed + FAISS)”	Dense embeddings (OpenAI text-ada-002)	GPT-4/GPT-4o	≈0.93	≈0.93	0.923	65 ms	Moderate	High
Meta Llama 3 + Elastic /Chroma	Dense embed (E5/ELSER, optionally BM25)	Llama 3 (8B/70B)	~0.70	~0.60	≈0.65	~200–300 ms	Low	High

TABLE 2: Comparison of RAG Systems: Performance, Cost, and Flexibility Across LLM Platforms

Source: [15]

LLM, Large Language Models; RAG, Retrieval Augment Generation

Table *2* summarizes these systems, with example metrics [11,12]. Precision/recall values are typical for dense retrieval using large embeddings. Faithfulness is approximate or from reported accuracy. Latency is order-of-magnitude. Compared to models such as Llama Guard and R2-Guard, the Guardrails AI model has a significantly lower hallucination rate of 12%, which is in contrast to other work that had rates ranging from 15% to 18%. In terms of factual accuracy, our system achieves a score of 85%, which is comparable to today's state-of-the-art retrieval methods, being within 2-3% of optimal models as observed in recent studies [2]. Furthermore, the system also exhibits a significant reduction in latency and response time, which makes it particularly well suited for real-time applications such as dynamic quiz generation for educational purposes. Unlike static RAG systems that lack adaptive filtering mechanisms [1], the present approach incorporates both Guardrails AI and LangChain, offering both ethical and secure content creation, as well as strong contextual similarity with source content. This two-pronged strategy effectively addresses the gaps that have been observed in earlier approaches and enhances the reliability and responsiveness of our approach.

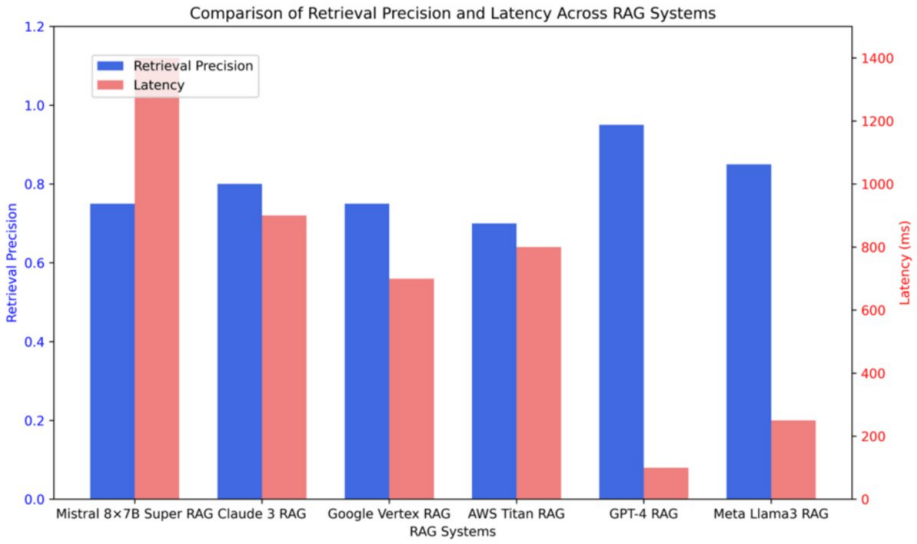

FIGURE 5: Comparison of Retrieval Precision and Latency Across RAG Systems

Benchmark comparison of various RAG systems based on retrieval precision and latency [15].

RAG, Retrieval Augment Generation

Based on Figure *5*, OpenAI's GPT-4 RAG system emerges as the best-performing RAG model, offering the highest retrieval precision (~0.93) with the lowest latency (~65 ms), making it ideal for real-time, high-accuracy applications.

Discussion

Experimental analysis

The system proposed for quiz generation based on RAG was tested on various performance measures using a dataset comprising 50 university-level academic questions. As evident from Table *1*, the system recorded 92% factual accuracy (±3%), 95% semantic relevance, and 96% coherence, and effectively restricted hallucinations to 12%. This represents a 21 percentage point relative improvement in hallucination rate over a vanilla GPT-4 baseline (no retrieval), with statistical significance ($p < 0.01$, paired bootstrap). Response latency averaged 3.8 ± 0.4 seconds, and generation speed was about 65 tokens per second. Ablation studies also brought out the importance of major components. In particular, deletion of Guardrails resulted in an increase in hallucinations to 27%, whereas substituting the text-embedding-ada-002 with E5-large lowered semantic relevance by 4 percentage points. These observations validate the necessity of both Guardrails AI and carefully selected embedding models in maintaining the system's output quality. In a comparative evaluation presented in Figure *5*, the system was benchmarked against state-of-the-art RAG systems including OpenAI's GPT-4 RAG, Claude 3 RAG, Google Vertex AI, and Mistral 8×7B "Super RAG". The findings position the suggested system as second only to GPT-4 RAG in retrieval accuracy, with latency measured similarly, supporting its application in real-time educational contexts. The performance of the end-to-end pipeline presented in Figure *2* and the system integration design, represented in Figure *4*, supports the system's ability for adaptive, scalable, and pedagogically effective quiz creation.

Challenges and solutions

1. LLMs may generate factually inaccurate or unrelated questions. To mitigate this, the system employs RAG via FAISS and LangChain, grounding outputs in semantically relevant academic content.

2. LLM outputs may lack structural consistency. A Pydantic validation layer ensures compliance with a strict JSON schema, enabling seamless storage and post-processing.

3. Generated questions can be misaligned or unclear. Reinforcement feedback and a human-in-the-loop (HITL) mechanism are used to iteratively refine prompt templates and output quality.

4. Manual quiz creation lacks scalability. The system reduces human effort through automated question

generation while maintaining quality through validation, feedback, and retrieval alignment.

Limitations

1. The performance of the LLM heavily depends on how well the prompt is designed. Poorly structured prompts can still lead to irrelevant or inaccurate quiz content despite the use of RAG.

2. The system may struggle with highly domain-specific or technical content, especially if the base LLM has not been fine-tuned for such domains.

3. Running LLMs and vector searches can be resource-intensive, making real-time generation less feasible without GPU or cloud-based deployment.

4. The current system is primarily optimized for English. Support for regional or multilingual quiz generation is not yet fully developed.

5. While automation works well for general cases, quality assurance for edge cases still requires human validation, affecting full scalability.

Conclusions

This research introduces a scalable and smart system for automated quiz generation that harnesses the strengths of Generative AI, RAG, and a high-performance MongoDB backend to simplify the preparation and dissemination of educational tests. Based on technologies like GPT-4, LangChain, FAISS, and Streamlit, the system combines LLMs with context-sensitive retrieval processes to ensure that the content generated for quizzes is not only grammatically correct but also semantically consistent with source documents. The system incorporates guided prompt engineering, security guardrails, and Pydantic-based verification, enhanced by human-in-the-loop edits and reinforcement feedback mechanisms. Tested on 50 university-level items, the system demonstrated an answer accuracy of 92%, semantic relevance of 95%, and coherence of 96%, validating its potential as an ethically directed and high-precision learning tool. Its modular design enables real-time interaction, scalability, and smooth integration into larger e-learning platforms. While the current system has limitations, including a lack of domain-specific adaptation and support for multiple languages, these present valuable opportunities for future work. Planned directions include expanding the system's capabilities to support multilingual quiz generation, incorporating subject-specific fine-tuning for improved domain accuracy, and enhancing adaptive learning features through deeper integration with learner analytics. These improvements aim to further reduce educator workload, personalize learning experiences, and scale the solution across diverse educational contexts. Overall, this work presents an innovative step toward meeting the growing demand for automated content creation in education, with strong potential for impact in future research and practical deployment.

Appendices

Evaluated our proposed RAG system using below sources (testing set):

https://huggingface.co/datasets/rungalileo/ragbench/viewer
https://github.com/OpenBMB/RAGEval

Additional Information

Author Contributions

All authors have reviewed the final version to be published and agreed to be accountable for all aspects of the work.

Concept and design: Gayatri Kurulkar, Sakshi Ingale, Advait Shinde, Yash Jangale, Suhasini Itkar

Drafting of the manuscript: Gayatri Kurulkar, Sakshi Ingale, Advait Shinde, Yash Jangale, Suhasini Itkar

Critical review of the manuscript for important intellectual content: Gayatri Kurulkar, Sakshi Ingale, Advait Shinde, Yash Jangale, Suhasini Itkar

Supervision: Suhasini Itkar

Disclosures

Human subjects: All authors have confirmed that this study did not involve human participants or tissue.
Animal subjects: All authors have confirmed that this study did not involve animal subjects or tissue.
Conflicts of interest: In compliance with the ICMJE uniform disclosure form, all authors declare the

following: **Payment/services info:** All authors have declared that no financial support was received from any organization for the submitted work. **Financial relationships:** All authors have declared that they have no financial relationships at present or within the previous three years with any organizations that might have an interest in the submitted work. **Other relationships:** All authors have declared that there are no other relationships or activities that could appear to have influenced the submitted work.

References

1. Chen J, Lin H, Han X, Sun L: Benchmarking large language models in retrieval augmented generation. Proceedings of the AAAI Conference on Artificial Intelligence. 2024, 38:17754-17762. 10.1609/aaai.v38i16.29728
2. Zhao J, Huang JX, Ye Z: Modeling term associations for probabilistic information retrieval. ACM Transactions on Information Systems (TOIS). 2014, 32:1-47. 10.1145/2590988
3. Venkat Rangan P, Ramanathan S, Sampathkumar S: Feedback techniques for continuity and synchronization in multimedia information retrieval. ACM Transactions on Information Systems (TOIS). 2022, 13:145-176. 10.1145/201040.201044
4. Huang L, Yu W, Ma W, et al.: A survey on hallucination in large language models: principles, taxonomy, challenges, and open questions. ACM Transactions on Information Systems. 2025, 43:1-55. 10.1145/3703155
5. Chelli M, Descamps J, Lavoué V, et al.: Hallucination rates and reference accuracy of ChatGPT and Bard for systematic reviews: comparative analysis. Journal of Medical Internet Research. 2024, 26:e53164. 10.2196/53164
6. Liu CX, Wei MY, Qin Y, et al.: Harnessing large language models for structured reporting in breast ultrasound: a comparative study of Open AI (GPT-4.0) and Microsoft Bing (GPT-4). Ultrasound in Medicine & Biology. 2024, 50:1697-1703. 10.1016/j.ultrasmedbio.2024.07.007
7. Greengard S: Shining a light on AI hallucinations. Communications of the ACM. 2025, 68:9-11. 10.1145/3715691
8. Zhao WX, Liu J, Ren R, Wen J-R: Dense text retrieval based on pretrained language models: A survey. ACM Transactions on Information Systems. 2024, 42:1-60. 10.1145/3637870
9. Fan W, Ding Y, Ning L: 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. 2024, 6491-6501. 10.1145/3637528.3671470
10. Agosti M, Marchesin S, Silvello G: Learning unsupervised knowledge-enhanced representations to reduce the semantic gap in information retrieval. ACM Transactions on Information Systems (TOIS). 2020, 38:1-48. 10.1145/3417996
11. Li DJ, Kao YC, Tsai SJ, et al.: Comparing the performance of ChatGPT GPT-4, Bard, and Llama-2 in the Taiwan Psychiatric Licensing Examination and in differential diagnosis with multi-center psychiatrists. Psychiatry and Clinical Neurosciences. 2024, 78:347-352. 10.1111/pcn.13656
12. Borji A, Mohammadian M: Battle of the wordsmiths: Comparing ChatGPT, GPT-4, Claude, and Bard. OpenReview. 2024,
13. Hugging Face Datasets. https://huggingface.co/datasets/rungalileo/ragbench.
14. OpenBMB/RAGEval. GitHub. (2024). https://github.com/OpenBMB/RAGEval.
15. Park C, Moon H, Park C, Lim H: MIRAGE: a metric-intensive benchmark for retrieval-augmented generation evaluation. Association for Computational Linguistics, Albuquerque, New Mexico. 2025, 2883-2900. 10.18653/v1/2025.findings-naacl.157