Enhancing Engineering Education Through LLM-Driven Adaptive Quiz Generation: A RAG-Based Approach

Sreekanth Gopi
Department of Computer Science
Kennesaw State University
Kennesaw, GA
sgopi@students.kennesaw.edu

Devananda Sreekanth
Department of Computer Science
Kennesaw State University
Kennesaw, GA
dsreekan@students.kennesaw.edu

Nasrin Dehbozorgi
Department of Software Engineering
Kennesaw State University
Kennesaw, GA
dnasrin@kennesaw.edu

***Abstract*—This research-to-practice study aims to develop an Artificial Intelligence (AI) MCQ generation system for engineering students, with a focus on adaptive learning, educational technology, and innovative assessment tools, to enhance personalized learning. Engineering education faces significant academic performance challenges, with first-year retention rates in STEM fields ranging between 27% to 46%, largely due to poor academic achievements. Multiple Choice Questions (MCQs) identify misconceptions, reinforce knowledge retention, and offer efficient assessment methods for engineering education. This interactive method improves attention and memory retention, reinforces knowledge, and improves comprehension. In this context, the emergence of Large Language Models (LLMs) such as GPT-4 has marked a significant advancement. Our literature review method employed a systematic approach, analyzing peer-reviewed articles, conference papers, and authoritative reports to uncover the trends and challenges in AI-driven quiz generation. The notable gap identified in our literature review is the lack of LLM-based adaptive quiz generation methods specifically for engineering education. Our methodology involved sourcing relevant structured datasets, data pre-processing, embedding generation, vector database storage, hybrid-search retrieval, LLM query results feed, prompt engineering, and context-based response. In this research, we adopted Vectara as a vector database tool for its automatic data ingestion capabilities and seamless integration with generative AI applications. Prompt engineering involves a dual-prompt approach, where the Contextual Question Prompt formulates questions based on user topics and chat history, while the Answer Question Prompt manages MCQ responses with explanations, ensuring relevant and contextually accurate interactions. Evaluation includes topic relevancy, answer relevancy, and a contextual relevancy score. Preliminary results indicate promising results for the generation of accurate and contextually appropriate questions with minimal hallucinations. The quiz generation system was deployed using Streamlit cloud-based architecture to showcase the functionality. Looking forward, we aim to expand the dataset to include more diverse engineering disciplines and to refine the retrieval algorithms to better handle complex diagrams and mathematical expressions commonly found in engineering texts.**

***Index Terms*—AI quiz generation, engineering education, personalized learning, Large Language Models, GPT-4, RAG, Vectara, prompt engineering, LLM evaluation.**

I. Introduction

Engineering education faces significant academic performance challenges, with first-year retention rates in STEM fields ranging between 27% to 46%, largely due to subpar academic achievements [1]. Due to its active engagement of students, Quiz-based learning has a definite advantage over passive learning methods, such as lectures or reading assignments. Multiple Choice Questions (MCQ) give students insight into possible misconceptions through distractor options that reveal common errors in understanding. This interactive method improves attention and memory retention, reinforces knowledge, and improves comprehension. [2].

Engineering students display a range of cognitive styles and learning needs, including analytical problem-solving, creative visualization, systematic planning, and adaptable thinking [3]. Adaptive learning through personalized quiz systems caters to these diverse styles by dynamically modifying the difficulty of text-based questions. This approach helps analytical thinkers engage with logical problem-solving tasks, while creative thinkers benefit from visually stimulating prompts and challenges. Systematic learners could receive progressively structured questions that reinforce progression, and adaptable learners could get varied question types that sustain their interest [3]. By analyzing student responses in real-time, adaptive learning aligns with individual capabilities and preferences, resembling a personalized human tutor [4].

MCQ and Adaptive Learning in Engineering Education

Leveraging the strengths of MCQs, adaptive learning systems can further enhance educational outcomes by tailoring question difficulty and types to individual learning styles. Research shows that Artificial Intelligence (AI) is increasingly integral to enhancing engineering education by facilitating personalized learning experiences that adapt to the needs and preferences of individual students. AI, particularly through LLMs like OpenAI's ChatGPT, enables a dynamic learning environment where content and challenges are tailored in real-time to match the academic level and learning style of each student [5]. Chat-based systems, which leverage conversational

agents such as ChatGPT, offer an interactive approach that can simulate one-on-one tutoring experiences [6]. Moreover, the integration of AI into educational tools provides a platform for students to receive immediate feedback and clarification, enhancing their understanding and ability to apply knowledge practically [7]. Such systems can simulate various pedagogical roles, from tutor to evaluator, adapting the complexity and nature of questions based on the learner's performance.

The Role of AI in Personalized Learning

OpenAI's GPT-3.5 and GPT-4 LLM models are often preferred over others due to their advanced capabilities in handling complex educational content and providing in-depth question generation. GPT-4, an advancement over GPT-3.5, offers significant improvements with a larger context window and enhanced reasoning abilities, making it more suitable for generating detailed assessments that require a deeper understanding of the subject matter [8]. It can also process complex scientific contents, interpret programming languages, and debug algorithms making it suitable for engineering education [9]. Even more, engineering education has documents containing technical drawings, schematics, complex charts, mathematical formulas, flowcharts, and diagrams. GPT-4 Vision, an extension of OpenAI's advanced language models, integrates capabilities to read and analyze visual content like circuit diagrams or complex system designs and transcribe them in the form of textual content [10].

Advanced Capabilities of OpenAI's Models

LLMs, particularly autoregressive models like GPT-3 or GPT-4, function as next-token-prediction machines. They analyze sequences of tokens, such as words or characters, and predict the likelihood of the next token based on the preceding ones. For instance, given a sequence of tokens, the LLM estimates the probability distribution of the next token conditioned on the previous ones, subsequently selecting the most probable token to append to the sequence. However, limitations arise when the model's predictions diverge from expected outcomes, leading to what is known as hallucinations [11]. Retrieval-Augmented-Generation (RAG) systems enhance LLMs by dynamically incorporating external, verifiable data sources, which significantly improves the accuracy and relevance of the generated responses. RAG addresses the static nature of LLM training datasets by providing access to updated and authoritative information, ensuring responses are current and contextually appropriate [12]. This integration not only prevents the generation of outdated or incorrect information by the LLM but also reduces the need for frequent retraining of the model on new data, thus saving computational and financial resources [13]. Moreover, RAG enables the personalization of responses, which can be crucial for applications like customer service, where responses must be tailored to individual queries or issues [14]. By utilizing vector databases, RAG systems efficiently index and retrieve the most relevant data for any given query, enhancing the overall performance of LLMs in generating responses.

Our Contribution

In this study, our contribution is as follows:

- **Specialized AI Quiz System:** Developed an AI-based quiz generation system for engineering education using GPT-4.
- **Advanced Prompt Engineering:** Implemented a prompt engineering strategy utilizing zero-shot prompting and contextual data retrieval to dynamically generate and rank MCQ content based on difficulty levels.
- **RAG Integration:** Enhanced quiz precision by incorporating Retrieval-Augmented Generation (RAG).
- **System Evaluation:** Assessed topic relevancy and answer accuracy, confirming the system's educational value.
- **Scalability Assessment:** Deployed via Streamlit cloud for scalable educational use.

II. Literature Review

We conducted a deeper literature review on the above concepts on research platforms like Google Scholar, PubMed, IEEE Xplore, ACM, and other relevant academic databases to gather comprehensive insights. The literature shows that recent advancements in AI have significantly influenced quiz generation systems within engineering education, enabling more effective and personalized learning experiences. The development of automatic question generation systems using LLMs has been validated for their efficiency in creating diverse and challenging questions that adapt to student needs [15]. Research indicates that AI-based learning content generation enhances student engagement by providing tailored learning pathways [16]. Moreover, surveys of automatic question generation methods highlight the importance of integrating multiple data sources to improve the relevance and complexity of quiz content [17]. Adaptive learning systems, employing real-time analytics and feedback, offer personalized assessments that align with individual cognitive styles and educational requirements [18].

Advantages of LLMs in Engineering Education

While LLMs bring numerous benefits to engineering education, they also present several challenges that need careful consideration. The integration of AI-driven quiz generation systems in engineering education faces certain challenges associated with these systems, such as the potential for generating misleading or inaccurate information, which necessitates rigorous oversight and validation processes to ensure content reliability [19]. Moreover, there are limited systems developed in engineering education that can dynamically adjust to the varied learning paces and styles of individual students [20]. This issue stems from the fact that AI systems are not always supported by sufficiently diverse and relevant datasets, which restricts their ability to effectively tailor content to meet different classroom needs and curricula [20].

An increasing body of literature shows that LLMs, trained with billions of parameters, can handle complex scientific data to deliver personalized, adaptive learning in engineering education [21]. AI systems like GPT-4 showcase an impressive

ability to autonomously solve complex problems without training on a specific topic [22]. Even more, they identify students' learning patterns through massive educational data and thus can offer customized tutoring like real-time problem-solving, learning advice, and academic guidance through dialogue and interaction with students, based on specific context [23]. LLMs possess the capacity for educational assessment, autonomously gauging students' mastery of knowledge, learning outcomes, and expressive skills [23]. Through interactive dialogue, this empowers students to take responsibility for their learning, as they can independently learn, and acquire knowledge and skills with self-motivation and difficulty management. Thus, recent models of LLMs with capabilities as shown in Table I, have striking proximity to human-level performance and can act like omnipresent personal tutors, available for assistance at any hour.

LLM Feature	**Description**
Personalization	Customized learning experiences
Adaptive feedback	Immediate, tailored guidance
Diverse resources	Wide range of materials
Natural language	Conversational interaction
Continuous support	24/7 learning assistance
Content creation	Automated resource generation
Multilingual	Language diversity
Learning analytics	Insightful progress tracking

TABLE I: The LLM advantage

Challenges and Limitations

While LLMs offer significant advancements in personalized learning, literature also points to critical gaps in their application. These include the need for specialized training to prevent the generation of inaccurate information, which can undermine the educational integrity and efficacy of these models. And, the generalized LLM models without specialized expertise, are prone to inaccuracies and out-of-context answers. Moreover, their inner workings are a mystery (black box opacity), making it hard to understand how they arrive at their answers. To fine-tune an LLM could incur high training costs and may require massive computational resources, datasets, and ML expertise. As an alternative to this, the integration of RAG systems and vector databases could enhance the accuracy and relevancy of generated content [24]. RAG systems work by dynamically pulling in external, verified data at the time of query, which helps LLMs produce more accurate and grounded responses. This method enriches the LLM's output by providing contextually relevant, real-world information, ensuring that the generated responses are not only relevant but also current [13]. This research builds on findings from [19] and [20], which discuss the operational challenges and data diversity issues in AI-driven educational systems. These studies highlight the need for robust and versatile AI systems that can adapt to varied educational content and learner profiles.

The Advantages of RAG System

Research on Retrieval-Augmented Generation (RAG) systems shows their effectiveness in enhancing the accuracy of LLM-generated content by incorporating real-time, verified data. The vector database retrieval system operates by semantically analyzing queries to fetch contextually relevant documents, enhancing the quality of generated responses [25]. Subsequent to an initial prompt like, "Recommend design improvements for a suspension bridge", the retrieval component dynamically queries a vector database containing the documents. This query uses a semantic similarity search to identify relevant articles focused on suspension bridge design and documented challenges. The retrieved documents are then incorporated into the original prompt, constructing an augmented prompt enriched with contextual information. This augmented prompt may resemble: "Recommend design improvements for a suspension bridge. Prior research suggests that optimizing truss configurations can enhance wind shear resistance. Furthermore, it emphasizes the significance of material selection for long-span bridges..." The augmented prompt is then provided to the LLM by hybrid querying, enabling it to utilize the retrieved information on the example of wind resistance and material selection, for generating a more comprehensive and data-driven response [25]. Research on Retrieval-Augmented Generation (RAG) systems shows their effectiveness in enhancing the accuracy of LLM-generated content by incorporating real-time, verified data. These systems dynamically pull in external data at the query time, improving the reliability and relevance of the responses generated by LLMs.

Model Evaluation

Hallucinations in LLMs stem primarily from their training data and the absence of robust mechanisms for assessing response accuracy. Additionally, LLMs lack mechanisms to acknowledge uncertainty or insufficiency of information, often opting for the most probable response irrespective of its veracity [11]. To evaluate the RAG-based LLM system, a multi-pronged approach could be adopted. First, relevance could be assessed by measuring semantic similarity, factual accuracy, and alignment with the question's intent. This ensures the generated questions target the intended learning objectives and avoid irrelevant or misleading information [26]. Second, the groundedness metric could be used to evaluate the coherence and consistency between the questions and the provided context. This metric measures the out-of-context or nonsensical questions by verifying their logical connection to the supplied information [27]. Finally, overall coherence and quality could be assessed, by examining factors like clarity, conciseness, and linguistic fluency. This ensures the questions are well-structured, grammatically sound, and free from ambiguity, promoting effective knowledge assessment [27]. In short, the evaluation of Retrieval Augmented Generation (RAG) models helps in reducing risks of LLM hallucinations and inaccuracies, vital for engineering education.

In this research, we adopted Vectara as a vector database tool for its automatic data ingestion capabilities and seamless integration with generative AI applications. Vectara's vector database provides in-built data ingestion by converting

diverse files into structured data formats, helping maintain data relationships and the special significance of specific data types. This structured approach ensures efficient retrieval through filters and advanced searches. When a document is ingested, Vectara automatically divides the content into semantic chunks, encoding them into vectors with metadata that can be searched via keyword, semantic, or hybrid techniques [28].

III. Methodology

Our methodology is designed to harness the power of AI to create a dynamic and adaptive quiz generation system tailored to the needs of engineering education. We selected Vectara's [29] information retrieval engine for its superior indexing and search capabilities, which are important for efficiently handling the complex datasets typical in engineering education. Similarly, OpenAI's LLMs are applied for their advanced natural language processing capabilities to enable the generation of context-aware MCQs that are both challenging and educational (figure 1). We begin by integrating and indexing a science question dataset using Vectara's API. Extracted documents are then transformed into structured formats using custom functions. The system utilizes a hybrid search approach along with Vectara's configuration for optimal retrieval. Finally, we call GPT-4 API to generate context-aware MCQs and implement an adaptive quiz system using LangChain.

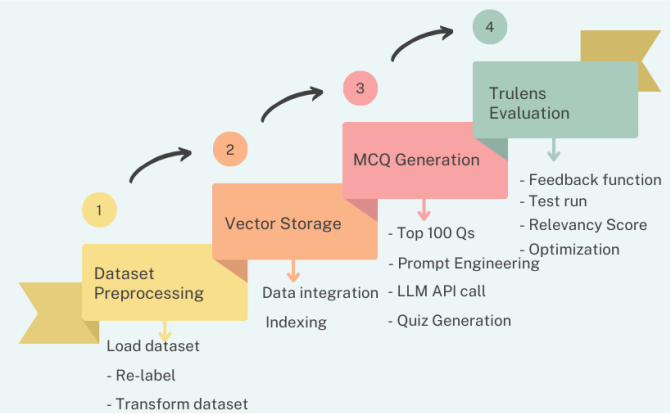

Fig. 1: MCQ Generation Pipeline

A. Dataset

The SciQ dataset encompasses 13,679 multiple-choice questions across scientific disciplines. This dataset was chosen due to its research-based, structured content, and data segmentation into test, train, and validate sets. Each entry in the SciQ dataset is formatted in a JSON structure that includes a primary question, three distractors as incorrect answers, a correct answer, and context. While this dataset primarily covers general science questions, its structured format, and extensive question bank provide an ideal foundation for developing and testing our quiz generation pipeline, which can then be adapted to more specialized engineering topics. To better align with the focus of this paper on engineering education, an example from mechanical engineering is provided below in the structured JSON format used by the dataset:

```
[
{
"question": "Which term describes the
mechanical advantage?",

"distractor1": "Work ratio",

"distractor2": "Energy coefficient",

"distractor3": "Load factor",

"correct answer": "Mechanical advantage",

"support": "Mechanical advantage is..."
}
]
```

In the pre-processing step, we transform the column names of the dataset, for clarity and system compatibility, to 'A', 'B', 'C', and 'D', respectively, and the data is reorganized to follow the structure required for standard multiple-choice question (MCQ) generation as shown below.

```
[
{
"question": "Which term describes the
mechanical advantage?",

"A": "Work ratio",

"B": "Energy coefficient",

"C": "Load factor",

"D": "Mechanical advantage",

"context": "Mechanical advantage is..."
}
]
```

Custom functions developed for this project include algorithms for reordering MCQ distractors to mitigate answer bias and enhance the fairness of the quizzes. Vectara's configuration settings were optimized to balance retrieval speed with accuracy, ensuring timely and relevant question generation.

B. Data Preparation and Indexing

Following the dataset pre-processing, the Data Preparation and Indexing phase involves leveraging Vectara's Standard Indexing API for efficient data integration and indexing [29]. This phase is crucial for organizing the SciQ dataset into a structured, searchable format, enhancing retrieval capabilities for quiz generation. Vectara's indexing system processes documents by segmenting them into sections, each with a unique identifier and metadata, which facilitates efficient and precise

data retrieval. The system supports various data formats and handles documents ranging from brief texts to extensive content collections, making it highly adaptable for educational applications [29].

C. MCQ Generation

After indexing and using Vectara's vector database to store the SciQ dataset, we employ Vectara's retrieval features to fetch the top 100 most relevant multiple-choice questions for quiz generation for context. The further step involves a multi-step process to generate questions using an LLM. Initially, a system prompt guides the creation of MCQs from a given context, ensuring that questions are inherently understandable on their own. This prompt-building process passes dynamically retrieved RAG documents to the LLM to formulate questions that are distinct yet relevant to the original data. Subsequently, another system prompt creates and manages interactive quiz sessions. Here, questions are presented to users, explanation is given, and user responses are evaluated. Further, the responses are analyzed by the LLM and as per prompt engineering, it measures student capacity and adjusts the question difficulty levels. The system offers a final score, review by the LLM suggesting the user's current knowledge level and areas for improvement, and saves the quiz history for personalized future learning pathways.

IV. Evaluation and Optimization

The RAG model retrieves relevant context from a vector store based on the user's input and chat history and then generates a response using that context. The Trulens-eval library is used to define a feedback function of context relevance that measures the relevance between the user's query and each retrieved context chunk. This feedback function is applied to the retrieved context chunks, and the relevance scores are aggregated. The custom application is then run, allowing the user to input queries and receive responses from the RAG model, while the topic relevancy feedback is recorded and evaluated [30]. The same process is repeated for evaluating LLM responses to ensure continual refinement and optimization of LLM responses. Thus, the evaluation process is orchestrated through a TruChain instance, configured with specific feedback to capture and analyze the performance of the RAG and LLM models in answering a diverse set of questions [31].

V. Results

The implementation of the methodology facilitated the establishment of a functional quiz-generation system that operates as intended. User interaction commences with the submission of a topic, which triggers the system to generate pertinent questions through the retrieval capabilities of RAG and LLM, as previously described. User Input Handling ensures that queries are accurately processed to fetch relevant data. The Question Retrieval and Response phase involves the AI in generating related MCQs and identifying the appropriate answers based on the context provided by the retrieval system and user inputs. The system then evaluates these responses, provides feedback, and adaptively generates subsequent questions, maintaining a continuous and engaging learning environment (Figure 1).

Trulens plays a pivotal role in independently verifying the effectiveness of the system through its rigorous methodology. The parallel Invocation function entails Trulens simultaneously invoking the LLM alongside the Vectra Retriever, facilitating real-time access to user inputs and retrieved data. This configuration supports Relevance Assessment, where Trulens independently evaluates the relevance, groundedness, and context relevance of the generated outputs, comparing them to established Trulens retrieval benchmarks aligned with the correctly supplied MCQ dataset. During this phase, Score Generation occurs, where Trulens computes and displays scores on a dashboard, providing an impartial assessment of the system's performance. This impartial scoring framework not only serves to validate the reliability of the generated questions but also establishes a clear benchmark for continuous system improvement. By leveraging the Trulens scoring dashboard, researchers can easily identify specific aspects of the quiz-generation pipeline that require refinement to ensure ongoing optimization of the RAG model.

A. Topic Relevance

In our methodology for testing topic relevance in the Q&A generation system, we employed a structured testing framework using Vectara's information retrieval combined with OpenAI's LLM. We configured the system to retrieve and reformat questions contextually relevant to the user's input, ensuring the content was aligned with predefined educational objectives. During testing, users interactively submitted thirty different questions, and the system processed these using an adaptive model to assess topic relevance. The results demonstrated an impressive topic relevance rate of 88.5%, with an average latency of 10.1 seconds per question, validating the effectiveness of our approach in providing contextually accurate educational content. Thus, the quiz generation system could accurately retrieve and generate questions across a wide range of educational topics.

B. Answer Relevancy

In our approach to assessing the relevancy of answers generated the system was set up to parse and respond to a variety of educational queries, ensuring that the answers were not only accurate but contextually relevant. Through testing of 30 MCQs, the context relevance scored a perfect 100%, underscoring the system's ability to effectively utilize the provided context in generating answers. Even more, the system achieved an 88% answer relevance score, indicative of its proficiency in generating accurate answers. These scores highlight the importance of maintaining accurate contextual alignment between retrieved data and generated responses to ensure that the learners receive reliable feedback.

C. Streamlit Deployment

The deployment of the quiz-generation system using Streamlit[1] involves several key steps. Initially, the necessary libraries and dependencies are installed, including Streamlit, LangChain, and OpenAI tools. The application then sets up a conversational chain within Streamlit to effectively process user inputs and dynamically generate multiple-choice questions. This configuration enables the app to function interactively: as users submit their inputs, the application retrieves pertinent information and constructs questions that are contextually relevant to the provided content. The conversation history enables the LLM to generate questions adaptively based on the engineered prompt as shown in our GitHub[2]. This seamless integration of LangChain and OpenAI ensures that the application remains highly responsive while providing relevant, accurate content in real time. This functionality allows educators to deploy the quiz-generation system confidently, knowing that it adapts to the evolving needs of their learners.

D. PDF Based Quiz Generation

We also developed a PDF-based quiz generation process where users can upload a PDF and engage with the content through the MCQ quiz generation framework[3]. Within the sidebar of the Streamlit interface, users are provided the option to upload documents and initiate processing, which is linked to the backend where the document is processed, and a conversational AI chain is established. This architecture leverages Streamlit's capability to manage session states that ensure that each user history is maintained and dynamically influenced by subsequent user experiences to maintain a continuous and coherent learning session. As each PDF document is processed and analyzed by the system, the subsequent questions generated create an interactive, relevant learning experience.

VI. Discussion

Large language models face the challenge of hallucination, where models generate inaccurate or misleading information. RAG is gaining traction not only to reduce it but also to dynamically incorporate relevant external information into the conversational process. Reviewing our pipeline, the RAG system involves a systematic approach to integrating relevant information within conversational systems as shown in the architecture (Figure 2). Users initiate this process by providing an initial prompt, which is further expanded using context retrieved through a vector database search. The first version of the prompt, Prompt v1, is expanded with retrieved contextual information and used in the LLM's context window to generate output. Subsequent prompts refine the search and are incorporated into the context window, guiding the system to provide progressively relevant responses. Prompt engineering plays a significant role here, with Prompt v1 contextualizing questions and Prompt v2 enhancing responses based on user input.

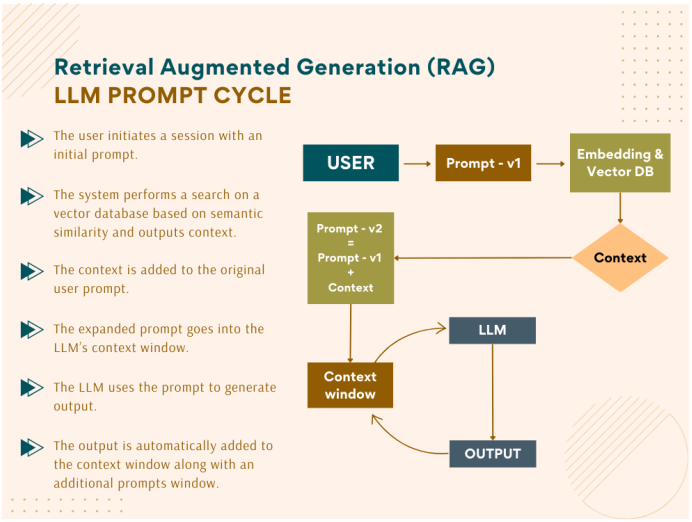

Fig. 2: RAG Architecture

RAG systems mitigate overfitting and catastrophic forgetting by referencing the given datasets, thereby reducing the need for computationally intensive LLM fine-tuning. They also enhance explainability, offering grounded and accurate responses while avoiding sensitive information leakage, as emphasized by [27]. The use of RAG as a knowledge retrieval framework significantly reduces the computational resources required for fine-tuning, improving system scalability. By integrating Vectara, we utilize its advanced indexing and retrieval features to efficiently source, organize, and retrieve relevant chunks of multiple-choice questions from the vector database. Vectara's Standard Indexing API facilitated efficient data integration and indexing, enabling seamless data retrieval. The indexing system adapts to data size and structure, ensuring scalable and efficient retrieval across different quiz contexts. Vectara's use of vector storage optimizes retrieval processes by using embeddings that enhance similarity searches, directly supporting prompt engineering.

Prompt engineering is integral to managing the contextual prompts within the RAG cycle. Contextual Question Prompt V1 (Figure 2) formulates questions based on chat history and user topics, ensuring that the generated questions make sense within the current context. It requires the LLM to form questions as per the chain of thought method [32] and without repeating exact phrases from retrieved documents, emphasizing context awareness. Meanwhile, Answer Question Prompt V2 instructs the system on presenting and managing multiple-choice questions (MCQs), ensuring appropriate responses with explanations. This division of prompts streamlines interactions and maintains relevance throughout the conversation. The clear distinction between these prompt types allows for a seamless transition from question formulation to answer generation, with prompts consistently aligned to user expectations and retrieval context. This methodology helps reinforce user learn-

[1]https://mcq-app.streamlit.app/
[2]https://github.com/datasci888/FIE2024Quiz
[3]https://mcq-app.streamlit.app/

ing while reducing the likelihood of irrelevant or redundant responses.

GPT-4.0 has demonstrated impressive zero-shot performance across various tasks and it effectively handles novel instructions without explicit task-specific training, proving its efficacy in evaluation scenarios [33]. Prompt v2 enhances responses based on user input and utilizes zero-shot prompting, which is the ability of a language model to generate appropriate responses or perform specific tasks without prior training on similar tasks or data [34]. In the context of assessing the difficulty levels of MCQs, zero-shot prompting can be used to evaluate a set of questions by leveraging the model's general understanding and reasoning capabilities [35]. One approach involves passing the top few retrieved relevant MCQs to the LLM to assess the topic, re-rank them as per difficulty, and generate similar MCQs as per difficulty level [36]. We implemented this method for difficulty-based ranking to sequence the MCQ generation as per user responses.

The combination of RAG systems with advanced prompt engineering strategies offers an effective approach to adaptive learning. This integration optimizes the retrieval and generation of MCQ content and enhances user learning through contextually relevant interactions.

VII. Limitation

In the context of engineering education, the quiz-generation system faces limitations when handling niche or highly specialized content due to a lack of comprehensive datasets. Even more, the system's reliance on tools like Vectara presents challenges in reading scientific data and charts, potentially limiting the accuracy and relevance of responses when dealing with chart-intensive engineering topics. Moreover, the lack of human oversight in the final responses may lead to factual inaccuracies or misinterpretation of data, particularly in specialized domains that require expert knowledge.

VIII. Conclusion and Future Work

In the future, the accuracy of answer relevancy can be improved by implementing enhanced retrieval techniques and ranking algorithms, ensuring that responses align more precisely with user intent. Future research should also explore how complex scientific documents can be integrated into RAG systems to transcribe intricate engineering content, similar to GPT-4 Vision. Additionally, incorporating advanced RAG methodologies like Self-reflection RAG — a framework that allows LLMs to self-assess, critique, and refine their output—would improve the system's ability to evaluate factual accuracy and retrieve more suitable questions. Further exploration of user interface design could also make the quiz-generation system more intuitive and user-friendly, encouraging broader adoption in educational settings. Building a framework for user feedback integration can enhance the system's adaptiveness to individual learning preferences.

References

[1] C. Alalouch, "Cognitive styles, gender, and student academic performance in engineering education," *Education Sciences*, vol. 11, no. 9, p. 502, 2021.

[2] D. Malandrino, I. Manno, G. Palmieri, V. Scarano, and G. Filatrella, "How quiz-based tools can improve students' engagement and participation in the classroom," in *2014 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, 2014, pp. 379–386.

[3] P. Johanes and L. Lagerstrom, "Adaptive learning: The premise, promise, and pitfalls," in *2017 ASEE Annual Conference & Exposition*, 2017.

[4] R. M. Clark and A. Kaw, "Adaptive learning in a numerical methods course for engineers: Evaluation in blended and flipped classrooms," *Computer Applications in Engineering Education*, vol. 28, no. 1, pp. 62–79, 2020.

[5] Y. Walter, "Embracing the future of artificial intelligence in the classroom: the relevance of ai literacy, prompt engineering, and critical thinking in modern education," *International Journal of Educational Technology in Higher Education*, vol. 21, no. 1, p. 15, 2024.

[6] T. Pham, T. B. Nguyen, S. Ha, and N. T. N. Ngoc, "Digital transformation in engineering education: Exploring the potential of ai-assisted learning," *Australasian Journal of Educational Technology*, vol. 39, no. 5, pp. 1–19, 2023.

[7] UNESCO, F. Miao, and W. Holmes, *Guidance for Generative AI in Education and Research*. UNESCO, 2023.

[8] W. Nwanne, "Comparing gpt-3.5 & gpt-4: A thought framework on when to use each model," *Tech Community*, 2024, published Mar 18, 2024, 06:15 AM. [Online]. Available: https://techcommunity.microsoft.com/

[9] J. Doughty, Z. Wan, A. Bompelli, J. Qayum, T. Wang, J. Zhang, Y. Zheng, A. Doyle, P. Sridhar, A. Agarwal *et al.*, "A comparative study of ai-generated (gpt-4) and human-crafted mcqs in programming education," in *Proceedings of the 26th Australasian Computing Education Conference*, 2024, pp. 114–123.

[10] T. Phung, V.-A. Pădurean, J. Cambronero, S. Gulwani, T. Kohn, R. Majumdar, A. Singla, and G. Soares, "Generative ai for programming education: Benchmarking chatgpt, gpt-4, and human tutors," in *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 2*, 2023, pp. 41–42.

[11] O. Mendelevitch, "Avoiding hallucinations in llm-powered applications," *Vectara Blog*, 2023, retrieved from https://vectara.com/blog/avoiding-hallucinations-in-llm-powered-applications/.

[12] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[13] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu,

L. Yang, W. Zhang, and B. Cui, "Retrieval-augmented generation for ai-generated content: A survey," *arXiv preprint arXiv:2402.19473*, 2024.

[14] Y. Huang and J. Huang, "A survey on retrieval-augmented text generation for large language models," *arXiv preprint arXiv:2404.10981*, 2024.

[15] B. Das, M. Majumder, S. Phadikar, and A. A. Sekh, "Automatic question generation and answer assessment: a survey," *Research and Practice in Technology Enhanced Learning*, vol. 16, no. 1, p. 5, 2021.

[16] C. Diwan, S. Srinivasa, G. Suri, S. Agarwal, and P. Ram, "Ai-based learning content generation and learning pathway augmentation to increase learner engagement," *Computers and Education: Artificial Intelligence*, vol. 4, p. 100110, 2023.

[17] N. Meißner, S. Speth, J. Kieslinger, and S. Becker, "Evalquiz–llm-based automated generation of self-assessment quizzes in software engineering education," in *Software Engineering im Unterricht der Hochschulen 2024*. Gesellschaft für Informatik eV, 2024, pp. 53–64.

[18] S. H. A. Faruqui, N. Tasnim, I. I. Basith, S. Obeidat, and F. Yildiz, "Integrating ai in higher education: Protocol for a pilot study with'samcares: An adaptive learning hub'," *arXiv preprint arXiv:2405.00330*, 2024.

[19] X. Xu, Y. Chen, and J. Miao, "Opportunities, challenges, and future directions of large language models, including chatgpt in medical education: a systematic scoping review," *Journal of Educational Evaluation for Health Professions*, vol. 21, 2024.

[20] M. Balfaqih and Z. Balfagih, "Ai-enhanced engineering education: Customization, adaptive learning, and real-time data analysis," in *AI-Enhanced Teaching Methods*. IGI Global, 2024, pp. 108–131.

[21] P. Bhargava and V. Ng, "Commonsense knowledge reasoning and generation with pre-trained language models: A survey," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, 2022, pp. 12 317–12 325.

[22] J. Bailey, "Ai in education: The leap into a new era of machine intelligence carries risks and challenges, but also plenty of promise," *Technology*, 2022. [Online]. Available: https://www.educationnext.org

[23] N. S. Raj and V. G. Renumol, "A systematic literature review on adaptive content recommenders in personalized learning environments from 2015 to 2020," *Journal of Computer Education*, vol. 9, pp. 113–148, 2022. [Online]. Available: https://doi.org/10.1007/s40692-021-00199-4

[24] K. Sawarkar, A. Mangal, and S. R. Solanki, "Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers," *arXiv preprint arXiv:2404.07220*, 2024.

[25] S. Ramlochan. (2023) Improving large language models with retrieval augmented generation. Redefining AI Conversations: How Retrieval Augmented Generation is supercharging Large Language Models for a smarter future. [Online]. Available: https://promptengineering.org

[26] C. Ryu, S. Lee, S. Pang, C. Choi, H. Choi, M. Min, and J.-Y. Sohn, "Retrieval-based evaluation for llms: A case study in korean legal qa," in *Proceedings of the Natural Legal Language Processing Workshop 2023*, 2023, pp. 132–137.

[27] J. Hayes, "Retrieval augmented generation: Making generative ai safe, trustworthy, and more relevant," *Vectara Blog*, 2023, retrieved from https://vectara.com/blog/retrieval-augmented-generation-making-generative-ai-safe-trustworthy-more-relevant/.

[28] O. Mendelevitch, "Retrieval augmented generation (rag) done right: Retrieval," October 2023. [Online]. Available: https://vectara.com/blog/retrieval-augmented-generation-rag-done-right-retrieval/

[29] Vectara. (2024) Data ingestion. [Online]. Available: https://docs.vectara.com/docs/learn/data-ingestion

[30] S. Sen, "Truchain recorder for langchain applications. build and evaluate llm apps with llamaindex and trulens," *TruLens*, 2023, retrieved from https://medium.com/trulens/build-and-evaluate-llm-apps-with-llamaindex-and-trulens-fd6bb4d86aca.

[31] TruLens, "Trulens documentation," Website, 2024, retrieved from https://www.trulens.org/trulens_eval/api/app/truchain/.

[32] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[33] C.-H. Chiang and H.-y. Lee, "A closer look into automatic evaluation using large language models," *arXiv preprint arXiv:2310.05657*, 2023.

[34] A. Liusie, P. Manakul, and M. J. Gales, "Zero-shot nlg evaluation through pairware comparisons with llms," *arXiv preprint arXiv:2307.07889*, 2023.

[35] S. Zhang, R. S. Shuttleworth, Z. Chin, P. Lantigua, S. Surbehera, G. Hunter, D. Austin, Y. Hicke, L. Tang, S. Karnik *et al.*, "Automatically answering and generating machine learning final exams," 2022.

[36] V. Raina and M. Gales, "Question difficulty ranking for multiple-choice reading comprehension," *arXiv preprint arXiv:2404.10704*, 2024.