HIGH SCHOOL EDITION
Journal of Student Research
Volume 14 Issue 1 (2025)

Knowledge Retrieval-Based Intelligent Question and Answer Generation Framework for Education

Anay Pardasani[1], Karen Maleski[#] and Sayantan Roy[#]

[1]Prospect High School, USA
[#]Advisor

ABSTRACT

The advent of Large Language Models (LLMs) has revolutionized educational technology, particularly in automated question generation. However, generating high-quality, contextually relevant examination questions requires substantial domain knowledge and understanding of educational standards. This paper presents a novel framework leveraging Retrieval Augmented Generation (RAG) to enhance LLMs' capability in generating and answering examination questions across diverse academic subjects. Our research addresses three primary challenges in implementing such systems: ensuring comprehensive subject coverage, maintaining educational standards in generated questions along with answers and developing robust evaluation metrics. The proposed system intelligently retrieves subject-specific content from a curated knowledge base and transforms it into examination questions along with answers tailored to different difficulty levels and assessment formats. Our experimental results demonstrate that the RAG-enhanced framework significantly outperforms traditional neural approaches in generating relevant and academically sound questions. We validate our findings through both automated metrics and expert educator evaluations. This research contributes to the growing field of AI-assisted education by providing a scalable solution for generating high-quality practice questions, potentially revolutionizing how students prepare for examinations.

Introduction

In this modern age, when each and every student and educator is becoming more and more dependent on digitization, the task of preparing quality, customized study materials has never been of higher priority. However, the creation of serious and realistic educational content still remains a complex process, as it requires good enough command over subject matter and academic standards. Traditionally, creating questions and answers was done manually by educators, being very labor-intensive and thus potentially limiting the variation and accessibility of quality study resources for students.Recent breakthroughs in AI and LLMs will definitely change this setup. These models can process large volumes of text, identify patterns, and create human-like language responses; therefore, they might provide a possible solution for automatically generating educational questions and answers. But as powerful as these LLMs might be, they often struggle to generate questions and answers that are consistently accurate and aligned with specific academic standards.

This paper proposes a new approach, RAG, which couples the linguistic acumen of large language models with the accuracy of a hand-curated knowledge base. The proposed RAG-based system will do much more than generate simple questions and answers but will ensure that these remain contextually relevant and academically aligned at various difficulty levels. This framework will probably assure a far more reliable means of producing high-quality, customized content to meet a range of educational needs by smartly retrieving particular information from the knowledge base and generating questions and answers based on the data retrieved.We believe that in this regard, our work addresses three major challenges regarding the deployment of such technology: complete coverage of the subject matter, quality questions and answers according to education standards, and valid metrics for its evaluation. A set of exhaustive experimentation along with educator expert feedback will show us that our RAG-enhanced framework can

effectively create academically sound questions and answers that help students in more lively and truly meaningful learning experiences.

In the end, this paper ushers in a critical development within the realm of educational AI to show how high-quality, customized content—including both questions and answers—can be opened up for easier access, both for learners and instructors alike. This approach opens doors to a new phase: one where wiser and more personalized learning tools finally meet advanced needs for both students and teachers.

Methods

Here we have developed a RAG (Retrieval Augmented Generation) based bilingual question answering system. This framework leverages the strengths of retrieval and generation based models, to improve the response according to the context.

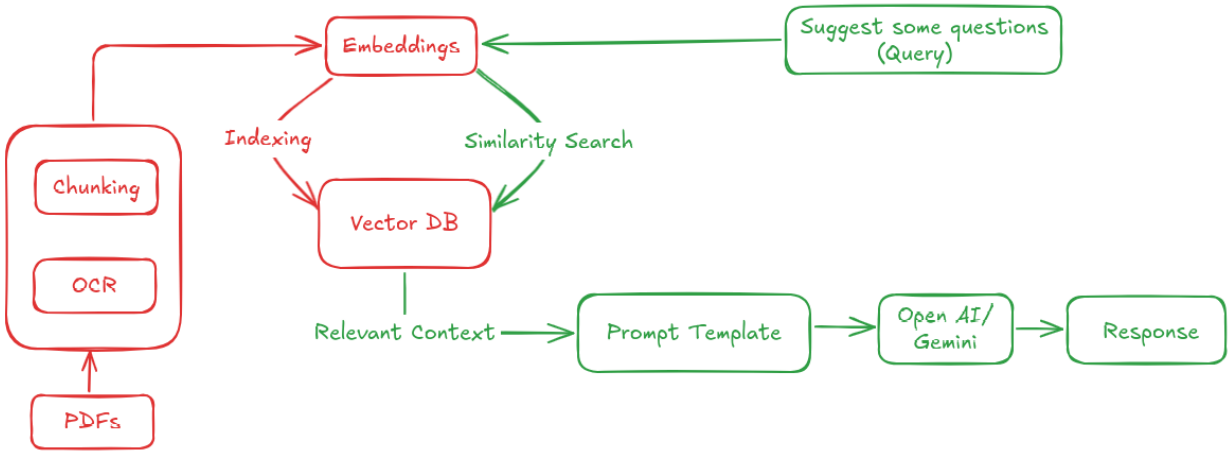

Figure 1. The architecture diagram of Q&A RAG system.

Data Preparation and Extraction

First, Optical Character Recognition (OCR) is performed on the PDFs to convert them into machine-readable text. Using LlamaParse, we efficiently extract figures, tables, and complex elements from these documents. After extraction, basic Natural Language Processing (NLP) techniques are applied to clean and preprocess the text. The cleaned text is saved in separate text files, with multilingual content stored distinctly to facilitate better processing and retrieval later on.

An essential step in the pipeline is data chunking, which involves breaking down large documents into smaller sections based on predefined rules. We explore various chunking methods such as fixed-size chunking, context-aware chunking, recursive chunking, and semantic chunking. Each method has its trade-offs, balancing between preserving context and improving retrieval accuracy. We employ LlamaIndex's sentence splitter for chunking with overlapping sections of fixed size. The overlap ensures contextual continuity, leading to better retrieval of relevant information, as evidenced in other RAG implementations.

Vectorization of Text

Post-chunking, we transform text into vector representations using multilingual BERT (mBERT), a transformer-based model that is designed to work with text in multiple languages. mBERT is particularly advantageous for it's self-

attention mechanism that captures nuanced semantics across different languages. The text is encoded into dense vectors of fixed dimensions, typically 768 or 1536, which are then utilized for retrieval and semantic search.

```
{
  "_id": 1,
  "sentence": "Organisms that convert decaying matter into nutrients",
  "vector": [0.23, 0.89, -0.10, 0.54, ...]  // vector representation
}
```

This structure is often referred to as a "node," where the vector is the primary representation, and additional metadata like a sentence or text snippet is stored alongside it. Furthermore, we can enrich these nodes by adding annotations, such as subject references, chapter numbers, or categories. This metadata becomes highly useful during search or retrieval processes, allowing us to filter or rank results based on specific criteria.

Retrieval

Once the vectors are stored, custom functions, like cosine similarity, are implemented to compare vectors and compute their similarities. Cosine similarity measures how close two vectors are in a multi-dimensional space, helping in tasks like identifying similar sentences or documents. This process, where embeddings are stored as arrays in nodes and then indexed for fast retrieval, is known as vector indexing. It enables efficient similarity searches.

After the retrieval setup is completed, the next stage is to use large language models (LLMs) to generate responses to user queries. The query is initially encoded into vectors using the same embedding paradigm, like mBERT (Multilingual BERT), when a user submits it. The goal of vectorization is to transform the query into a format that is easily comparable to the vectors kept in the database for retrieval based on similarity. We also attach the following prompt template to it.

‘’’

You are an assistant for question-answering tasks. Use the following pieces of retrieved context to answer the question.If you don't know the answer, just say that you don't know. Use three sentences maximum and keep the answer concise.
Question: {question}
Context: {context}
Answer:

‘’’

The LlamaIndex query engine is used to retrieve the most relevant data. To find the context that most closely resembles the user's query, it does a cosine similarity search across all of the MongoDB nodes. The pertinent context is returned by the search, together with details about the source nodes that yielded the closest matches. For this experiment we are using the top-k (k=1) hyperparameter.

Following its retrieval, the pertinent context serves as the language model's input. A prompt template with the context connected organizes the user's inquiry and includes the MongoDB data. After that, a large language model (LLM) is fed the enriched prompt to generate responses. This context is essential since it gives the LLM background information, which enhances the response's relevance and precision.

The system has been tested using both Gemini and OpenAI APIs to generate answers. Gemini is another LLM, and OpenAI refers to popular models like GPT-3.5 or GPT-4. The results from these models are compared to assess factors such as response quality, coherence, and relevance to the query. Both APIs offer different strengths: OpenAI’s models are widely regarded for their advanced understanding of context and fluency in generation, while Gemini provide competitive performance based on the specific tasks.

Results

To evaluate the retrieval-augmented generation (RAG) system's performance, we employed the RAGAs metric, which provides insights into various aspects of response quality. We have created an evaluation dataset from a separate set of pdfs. It contains 100 questions along with ground truth for each of them. Being a two step process, for retrieval we have context precision and context recall as metrics and for generation we have faithfulness and relevancy.

Faithfulness evaluates the generated response's factual coherence with the provided context. The response and recovered context are used to calculate it. Higher values on the scale of 0 to 1 indicate greater consistency in the response.If the generated answer's statements can all be deduced from the context, it is deemed accurate. First, a set of claims from the generated response is identified in order to ascertain this. Then, in order to determine whether these statements may be deduced from the context, they are cross-checked with it.

Example: Question:"How can acceleration be determined using a velocity-time graph?" Low Faithful Answer: "The slope of the velocity-time curve gives acceleration and the area under the curve gives distance." High Faithful Answer: "The slope of the velocity-time curve gives acceleration and the area under the curve gives displacement". If we calculate faithfulness using the first response, we first break it into a set of statements (S) and for each statement LLM will determine a binary verdict(V).

Statement 1:"The slope of velocity-time curve gives acceleration" , Verdict :"Yes"

Statement 2: "the area under the curve gives distance", Verdict "No"

Faithfulness = |V|/|S = ½= 0.5

Answer Relevancy is used to find out how the generated answers fair against the prompt. Higher ratings signify greater relevance, whereas lower values are assigned to answers that are lacking or contain repetitive information. Here from the generated response we ask the LLM to synthesize a question. The cosine similarity of this generated question with the original questions gives this metric

Answer relevancy = (1 / N) * ∑ (cos(E_g_i, E_0)) for i from 1 to N

Context Recall is the number of relevant documents retrieved, that helps the LLM in answering the questions. It penalizes when redundant information is fetched. Recall is just the ability to remember crucial details. Calculating recall always requires a reference to compare against because it is about not missing anything.
The value ranges from 0 -1. Higher value shows signs of good recall. Here we find out whether sentences in ground truth (GT) can be attributed to the retrieved context.

Context Recall = (No of sentences in GT that can be attributed to retrieved context)/no of sentences in GT

The percentage of relevant chunks in the retrieved_contexts is measured using a statistic called Context Precision. It is computed as the average precision@k for every context piece. The ratio of the number of pertinent chunks at rank k to the total number of chunks at rank k is known as precision@k.

Context Precision@K = (∑ (Precision@k * v_k) for k from 1 to K) / (Total number of relevant items in the top K results)

Table 1. Comparison of Metrics between 3 different LLMs

Metric	GPT-4	GPT 4o mini	Gemini 1.5

Faithfulness	0.6516	0.7044	0.4583
Answer relevancy	0.9779	0.9838	0.1201
Context Recall	0.6154	0.6575	0.6875
Context Precision	0.8756	0.7676	0.5000

Discussion

The performance of the three models—GPT-4, GPT-4o mini, and Gemini 1.5—on the question-and-answer dataset demonstrates significant variations in several key metrics: faithfulness, answer relevancy, context recall, and context precision. These metrics are essential in evaluating the quality of responses in a retrieval-augmented generation (RAG) framework. Each model's architecture and training objectives contribute to these differences.

In terms of metrics, Open AI model outperforms gemini. This disparity can largely be attributed to the differences in model architecture and training methodologies between OpenAI and Google's Gemini. OpenAI's ChatGPT models, built on the Generative Pre-trained Transformer (GPT) architecture, leverage extensive pre-training on vast datasets and attention mechanisms, which allow them to maintain context over multiple interactions and generate more contextually relevant responses. This high level of pre-training and fine-tuning gives the OpenAI models an edge in both faithfulness to the source material and relevancy in responses.

In contrast, Gemini 1.5, part of Google's evolving language model ecosystem, does not achieve the same level of accuracy in answer relevancy or precision. While Gemini is integrated into Google's extensive AI infrastructure, its architecture appears to be less optimized for conversational AI tasks that require detailed context tracking and precise response generation. This gap in performance could also stem from differences in training data and optimization techniques, with OpenAI potentially using a broader or more diverse dataset that enhances its model's adaptability to various contexts. Thus, while both OpenAI and Gemini have advanced architectures, OpenAI's refined approach to training and fine-tuning makes GPT-4 and GPT-4o mini more effective for high-precision tasks in retrieval-augmented generation applications.

Conclusion

In this paper we had built an end to end RAG pipeline with different LLMs for question and answer generation. With the advent of AI in the 90's, it has come a long way to help humans in their daily day-to-day tasks. The large language models performed well, but on the contrary we saw that the small language models are also equally effective in getting task-specific jobs done. This finding opens new avenues for further exploration with SLMs, emphasizing their potential to deliver efficient, scalable, and accessible AI solutions tailored to specific tasks and environments. As we look to the future, continuing to innovate with SLMs could greatly enhance the adaptability and affordability of AI in various domains.

Limitations

While our Retrieval-Augmented Generation (RAG) pipeline with large language models (LLMs) proved effective in generating question-and-answer pairs, certain limitations inherent to LLMs present challenges in ensuring reliability and scalability. A primary issue is hallucination, where LLMs sometimes produce factually incorrect or fabricated information despite access to relevant documents. Additionally, LLMs are highly resource-intensive, demanding substantial computational power, memory, and storage, which restricts deployment in real-time or resource-limited

environments. The effectiveness of an LLM in RAG systems also heavily depends on the quality of retrieved documents; irrelevant or incomplete retrieval can lead to off-base answers. Contextual drift is another limitation, where multi-turn responses can lose alignment with the original query, impacting coherence in longer interactions. Furthermore, LLMs often function as black boxes, making it difficult to interpret why certain responses were generated, which complicates debugging and refinement.

Acknowledgments

I am deeply grateful to Karen Maleski for generously taking the time to review my work. I would also like to extend my heartfelt thanks to Sayantan Roy for his mentorship and guidance throughout this project. His insights and encouragement helped me navigate the complexities of each iteration.

References

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *arXiv*. https://doi.org/10.48550/arXiv.1706.03762

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, Haofen Wang (2020). Retrieval Augmented Generation for knowledge intensive NLP tasks. https://doi.org/10.48550/arXiv.2005.11401

Shahul Es, Jithin James, Luis Espinosa-Anke, Steven Schockaert (2023). RAGAS: Automated evaluation for Retrieval Augmented Generation. https://doi.org/10.48550/arXiv.2309.15217

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2019). BERT: Pre Training of Deep Bidirectional Transformer for Language understanding. https://doi.org/10.48550/arXiv.1810.04805

P. Joshi, A. Gupta, P. Kumar and M. Sisodia, "Robust Multi Model RAG Pipeline For Documents Containing Text, Table & Images," 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2024, pp. 993-999, doi: 10.1109/ICAAIC60222.2024.10574972.

S. Kukreja, T. Kumar, V. Bharate, A. Purohit, A. Dasgupta and D. Guha, "Vector Databases and Vector Embeddings-Review," 2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP), Yogyakarta, Indonesia, 2023, pp. 231-236, doi: 10.1109/IWAIIP58158.2023.10462847.
Archit Parnami and Minwoo Lee (2022). Learning from Few Examples: A Summary of Approaches to Few-Shot Learning.https://doi.org/10.48550/arXiv.2203.04291

P. Omrani, A. Hosseini, K. Hooshanfar, Z. Ebrahimian, R. Toosi and M. Ali Akhaee, "Hybrid Retrieval-Augmented Generation Approach for LLMs Query Response Enhancement," *2024 10th International Conference on Web Research (ICWR)*, Tehran, Iran, Islamic Republic of, 2024, pp. 22-26, doi: 10.1109/ICWR61162.2024.10533345.

S. Selva Kumar, A. K. M. A. Khan, I. A. Banday, M. Gada and V. V. Shanbhag, "Overcoming LLM Challenges using RAG-Driven Precision in Coffee Leaf Disease Remediation," *2024 International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS)*, Bengaluru, India, 2024, pp. 1-6, doi: 10.1109/ICETCS61022.2024.10543859.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, Douglas C. Schmidt (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. https://doi.org/10.48550/arXiv.2302.11382