# Integrating LLMs and RAG techniques into mathematics learning for engineering students

1st Mohamed Malek Gritli
*ESPRIT School of Engineering, Tunisia*
Ariana, Tunisia
malek.gritli@esprit.tn

2nd Mohamed Anis Ben Lasmer
*ESPRIT School of Engineering, Tunisia*
Ariana, Tunisia
mohamedanis.benlasmar@esprit.tn

3rd Mohamed Hedi Riahi
*ESPRIT School of Engineering, Tunisia*
Ariana, Tunisia
mohamedhedi.riahi@esprit.tn

4th Lotfi Ncib
*Ridcha Data*
Paris, France
lotfi.ncib@ridchadata.com

2024 Artificial Intelligence Revolutions (AIR) | 979-8-3503-7788-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/AIR63653.2024.00021

***Abstract*—In this study, we present a novel pipeline designed to enhance the capabilities of large language models (LLMs) for solving mathematical problems ( exercises, course) using a retrieval-augmented generation (RAG) methodology. Recognizing that a straightforward RAG approach is inadequate for significantly improving the reasoning abilities of LLMs, we propose an advanced method to address these limitations. Our approach integrates sophisticated retrieval techniques with generative models to provide more accurate and contextually relevant solutions to mathematical problems. By combining these elements, we aim to improve the problem-solving efficacy of LLMs, making them more effective tools for educational purposes and beyond. This research demonstrates the potential of enhanced RAG methodologies in advancing the application of LLMs in mathematics education, offering a promising pathway for overcoming existing challenges in the field.**

***Index Terms*—pipeline, large language models, mathematical exercises, retrieval-augmented generation**

## I. Introduction

Mathematics is an indispensable component of engineering education, playing a pivotal role in design, simulation, optimization, and decision-making. However, learning mathematics often presents significant challenges for engineering students for several reasons.

Firstly, mathematics employs an abstract and rigorous language that is frequently detached from students' everyday experiences, making it more difficult to grasp concepts and apply them to concrete problems. Additionally, mathematics curricula in engineering schools are typically intense and demanding, with a rapid pace of learning that can overwhelm students, sometimes hindering their ability to keep up with the coursework. Effective learning of mathematics also requires specific study methods, such as regular practice, problem-solving, and research-based learning. Some students may struggle to master these methods, which can impede their success. These difficulties can negatively impact engineering students, depriving them of the essential understanding needed to tackle more advanced concepts and succeed in their projects [10]–[12].

These challenges are common among engineering students. By providing appropriate resources and developing specific skills, students can be better equipped to succeed in their studies and future careers. There are various methods and approaches to addressing the problems faced by mathematics students and developing effective solutions. In our work, our solution is based on natural language processing (NLP) and specifically, we propose the use of LLMs and RAG techniques to enhance mathematics learning in engineering schools.

Our solution offers support for problem-solving and course comprehension. By leveraging LLMs, students can solve complex mathematical problems by receiving detailed explanations and step-by-step solutions. Students can ask specific questions about course concepts or exercises and receive personalized explanations. Moreover, we provide quick access to resources for students using the RAG technique, which combines information retrieval systems with generative models to deliver accurate answers from a vast database of documents and academic resources. This enables students to rapidly access relevant information for their studies [10]–[13]..

Finally, we will develop an educational chatbot that integrates LLM and RAG techniques to assist students in real time, answering their questions on mathematical topics, providing examples, course theorems, and guiding them through problem-solving steps.

In summary, integrating LLMs and RAG techniques into mathematics learning for engineering students represents a promising solution for overcoming current challenges and enhancing their academic and professional success.

979-8-3503-7788-0/24/$31.00 ©2024 IEEE
DOI 10.1109/AIR63653.2024.00021

In this work, we begin with an overview of LLMs and RAG methodologies. We then present our methodology, exploring different combinations of LLMs and RAG approaches. Following this, we introduce the platform of our solution, showcasing its application and providing a comprehensive evaluation.

## II. Overview of LLM and RAG

LLMs have emerged as advanced artificial intelligence systems capable of processing and generating text with coherent communication, generalizing across multiple tasks. These models have recently showcased remarkable capabilities in natural language processing and other fields. This progress is largely due to significant breakthroughs in language models, particularly with transformers, enhanced computational capabilities, and the availability of large-scale training data [1]–[3].

LLMs have applications across a wide range of domains, including business, education, medicine, law, research, and more. However, integrating these models is not always straightforward. It requires a deep understanding of their mechanisms, training processes, evaluation methods for real-world effectiveness, and awareness of their potential and challenges.

RAG is a novel paradigm proposed by Patrick Lewis et al. in 2021 to enhance the effectiveness of LLMs and address their knowledge limitations. RAG ensures the generation of reliable outputs by incorporating up-to-date and accurate knowledge. Pre-trained LLMs often struggle with knowledge-intensive NLP tasks due to their limited capacity to integrate current, domain-specific knowledge and context, which restricts their applicability in real-world scenarios where timely and accurate responses are crucial. RAG models address this issue by combining pre-trained parametric memory with non-parametric memory for language generation [2], [5], [7]. The parametric knowledge of LLMs, stored in their weights and acquired during pretraining, remains static. In contrast, RAG's retrieval-based approach allows language models to access the latest information without the need for retraining. RAG operates by taking an input and retrieving a set of relevant documents from a specified source. These documents, concatenated with the original input prompt, provide context for the text generator, ultimately producing the final output.

The adaptability of RAG is particularly valuable in scenarios where facts may change over time. This includes domains such as the medical field, where knowledge is continually evolving with new research findings and discoveries; legal research, where laws and regulations are frequently updated and amended; and customer support, where accurate and timely assistance leads to higher satisfaction levels [6]–[8].

## III. Methodology

### A. Solution

In the ever-evolving landscape of education, technology continues to play a crucial role in enhancing learning experiences. Today, we are excited to introduce MathBot, an innovative chatbot designed to assist students in mastering mathematics. Whether you are grappling with complex algebraic equations, seeking to understand intricate calculus concepts, or simply looking for practice problems to sharpen your skills, MathBot is here to help. To train our MathBot, we use a database comprising various course chapters, tutorial exercises and their solutions, as well as past exams with their corresponding answers. The format of all these documents is pdf.

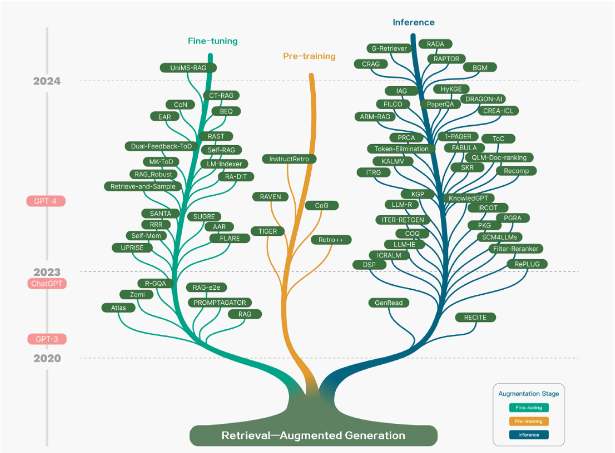


Fig. 1. Retrieval-augmented generation (RAG) [2], [4]

MathBot offers two primary modes of interaction. The first mode allows users to ask the LLM specific questions about the lesson, for example, we ask the chatbot to explain concepts such as probability laws, distribution functions, and other key topics in mathematics. In this scenario, the RAG technique enhances the performance of the LLM by retrieving relevant information to support the search for the most accurate answer, while also providing illustrative examples. The second mode involves utilizing the LLM to correct mathematical exercises. When the exercise is present in the database, a straightforward RAG approach can effectively assist the LLM in providing the correct answer along with a well-structured, step-by-step solution. However, if the exercise is novel and not included in the database, relying solely on a basic RAG methodology may prove insufficient for enhancing the LLM's performance. In such cases, a more sophisticated approach is necessary to ensure the LLM can adequately address and solve these unfamiliar problems.

Thus, we have identified two distinct interaction scenarios for MathBot:

1) Asking the chatbot specific questions related to the lesson.
2) Requesting the LLM to correct math exercises.

MathBot is designed to effectively address both scenarios, providing students with the support they need to excel in their mathematical studies.

### *B. Testing RAG, CRAG, and Hybrid RAG Methodologies*

By default, LLMs lack reasoning capabilities for math-related problems. To address this limitation, we explored RAG and its variants, such as Corrective RAG and Hybrid RAG. To benchmark their effectiveness, we configured and tested three RAG methodologies as follows [2], [4]:

- Naive RAG:
  In the Naive RAG configuration, we employed ChatGPT 3.5 as the LLM, integrated with the Text-Embedding-Ada-002 embedding model and VectorDB for retrieval. The dataset utilized for this setup includes various course chapters, tutorial exercises and their solutions, as well as past exams with their corresponding answers. This configuration aimed to enhance the LLM's performance in generating responses for math-related problems by leveraging retrieval-augmented generation techniques. The limitation of the Naive RAG configuration is that it relies on a fixed dataset consisting of course chapters, tutorial exercises, and past exams. This means that the LLM can only generate responses based on the information available within that dataset. If a student asks a question that is not covered in the dataset, the LLM may fail to provide an accurate or helpful answer.Additionally, the Naive RAG configuration employs a simple retrieval-augmented generation technique. In this setup, the LLM first retrieves relevant information from the dataset and then uses it to generate a response. However, the LLM may not always combine the retrieved information effectively to produce a coherent and informative answer.
- Corrective RAG:
  In the Corrective RAG setup, we utilized ChatGPT 3.5 as the LLM, complemented by the Text-Embedding-Ada-002 embedding model and VectorDB for retrieval purposes, using the same dataset as the Naive RAG configuration. This approach aimed to address the shortcomings of Naive RAG by integrating corrective strategies designed to enhance the accuracy and relevance of responses generated for math-related problems through retrieval-augmented generation techniques. While Corrective RAG addresses some of the key issues observed in the Naive RAG setup, it still faces several limitations. The effectiveness of the corrective strategies depends largely on the quality and completeness of the predefined corrective rules. If these rules do not comprehensively cover all possible problem scenarios or are inaccurately applied, the system may still produce incorrect or suboptimal responses. Moreover, even with the application of corrective strategies, the LLM can still encounter difficulties when dealing with complex or nuanced mathematical problems, especially those that require a deeper level of contextual understanding. In such cases, the system may fail to capture the underlying mathematical logic or relationships, limiting its ability to offer fully accurate solutions. Additionally, while Corrective RAG provides an improvement in terms of response quality, it may continue to struggle with advanced mathematical reasoning tasks that involve symbolic manipulations or require detailed, step-by-step problem-solving. The corrective mechanisms do not equip the system with sophisticated mathematical reasoning capabilities, which are necessary for more intricate problem sets. As a result, the model may still generate answers that are either incomplete or fail to fully address the user's question.
- Hybrid RAG: In the Hybrid RAG configuration, ChatGPT 3.5 served as the LLM, augmented by the Text-Embedding-Ada-002 embedding model and an additional BM25 model for retrieval. VectorDB was employed as the database source, the same dataset. This hybrid approach aimed to enhance the retrieval-augmented generation (RAG) method by integrating BM25, a relevance-based retrieval model, alongside traditional embedding techniques. The goal was to improve the accuracy and effectiveness of generating responses to math-related problems by leveraging a combined approach of parametric and non-parametric memory retrieval strategies. While Hybrid RAG combines the strengths of embedding-based and BM25 retrieval methods, it still faces several limitations. BM25, being a term-frequency-based model, may introduce biases towards frequently occurring terms. This could lead to retrieval of less relevant information in some cases. Embedding models can also exhibit biases, such as those related to representation gaps or cultural biases. Balancing the contributions of embedding-based and BM25 retrieval requires careful tuning of hyperparameters. Finding the optimal balance can be time-consuming and computationally expensive. Even with combined retrieval methods, the LLM may still struggle to fully understand complex mathematical concepts, especially those involving abstract reasoning or symbolic manipulation.

We evaluated these configurations using a customized evaluation pipeline (details provided later) and found that all methodologies failed to improve the answers; in fact, they worsened them.

## IV. Proposed New Architecture

In this work proposes a novel approach that integrates innovative ideas to enhance the effectiveness of these methodologies for solving math-related problems.

- Creating a Dataset:
  We utilize PDF math lessons as the basis for our dataset creation. For each lesson, such as those on Markov chains, we employ a powerful LLM for summarization. This process involves generating a concise summary and crafting example exercises directly related to the lesson content. Currently, each lesson in our dataset comprises a summary and one or two corresponding examples.

- Proposed Solution in Use Case:
  In handling user input, our system incorporates a robust mechanism for correction or reformulation. For instance, if a user queries ”what is the Markov chain,” our LLM corrects it to ”what is the Markov channel” and verifies if this is the intended query. This correction feature significantly enhances accuracy, especially in scenarios where users ask the chatbot about specific topics. However, for the second scenario where users ask the LLM to correct an exercise, while this correction capability is beneficial, it alone is not sufficient. Here, our chatbot identifies the main topic of the exercise, such as Markov chains, and ensures that the vector database includes a summary of this lesson. By leveraging chain-of-thought prompting, the chatbot accesses the vector database to retrieve the relevant lesson summary and examples stored in the chat history. This approach empowers the LLM to accurately formulate responses by referencing the pertinent lesson content. We present the pipeline of our proposed solution with detailed steps

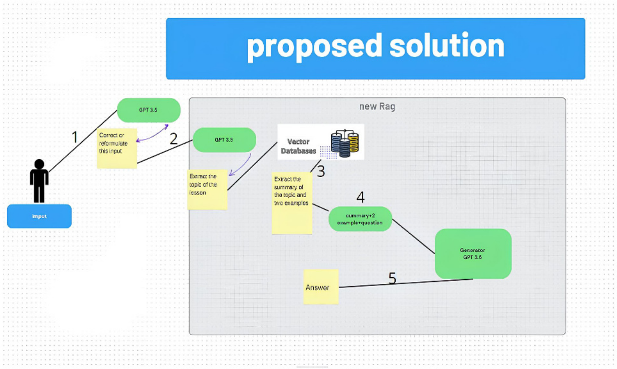
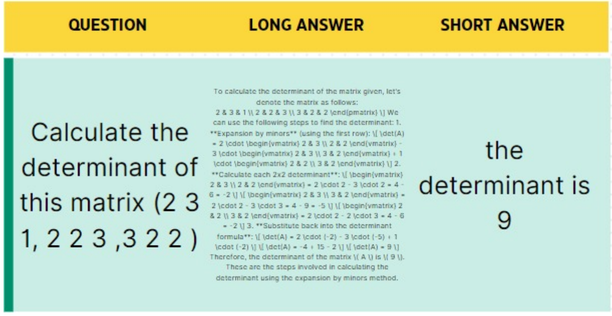

proposed solution
new Rag
GPT 3.5
1
Correct or reformulate this input
2
GPT 3.5
Vector Databases
Extract the topic of the lesson
3
Extract the summary of the topic and two examples
4
summary+2 example+question
Generator GPT 3.6
Input
Answer
5

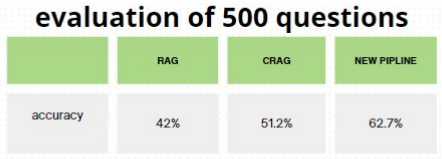
Fig. 2. Proposed Solution

- Creating a Pipeline to Evaluate Our MathBot:
  To evaluate our MathBot, we have developed a specialized pipeline that avoids using another LLM for assessment, as methods like RAG are unsuitable due to their limited reasoning capabilities. Instead, we have devised an evaluation pipeline that generates a dedicated dataset consisting of exercises and their correct answers. Each exercise question in this dataset is paired with both a detailed answer and a short answer. For instance, for a question like ”calculate the determinant of a matrix,” the detailed answer includes step-by-step calculations, while the short answer provides only the final result. Using an LLM, we generate the short answers for this dataset, leveraging its capability to extract straightforward results. During evaluation, we compare the short answers generated by our chatbot with the correct short answers in the dataset. Both the dataset and our chatbot’s responses are processed using the same model to extract short answers. This comparison enables us to assess the chatbot’s performance based on how accurately its generated short answers align with the correct answers in the dataset.



| QUESTION | LONG ANSWER | SHORT ANSWER |
|---|---|---|
| Calculate the determinant of this matrix (2 3 1, 2 2 3 ,3 2 2 ) | To calculate the determinant of the matrix given, let's denote the matrix as follows: 2 & 3 & 1 \\ 2 & 2 & 3 \\ 3 & 2 & 2 \end{pmatrix} \] We can use the following steps to find the determinant: 1. **Expansion by minors** (using the first row): \[ \det(A) = 2 \cdot \begin{vmatrix} 2 & 3 \\ 2 & 2 \end{vmatrix} - 3 \cdot \begin{vmatrix} 2 & 3 \\ 3 & 2 \end{vmatrix} + 1 \cdot \begin{vmatrix} 2 & 2 \\ 3 & 2 \end{vmatrix} \] 2. **Calculate each 2x2 determinant**: \[ \begin{vmatrix} 2 & 3 \\ 2 & 2 \end{vmatrix} = 2 \cdot 2 - 3 \cdot 2 = 4 - 6 = -2 \] \[ \begin{vmatrix} 2 & 3 \\ 3 & 2 \end{vmatrix} = 2 \cdot 2 - 3 \cdot 3 = 4 - 9 = -5 \] \[ \begin{vmatrix} 2 & 2 \\ 3 & 2 \end{vmatrix} = 2 \cdot 2 - 2 \cdot 3 = 4 - 6 = -2 \] 3. **Substitute back into the determinant formula**: \[ \det(A) = 2 \cdot (-2) - 3 \cdot (-5) + 1 \cdot (-2) \] \[ \det(A) = -4 + 15 - 2 \] \[ \det(A) = 9 \] Therefore, the determinant of the matrix \( A \) is \( 9 \). These are the steps involved in calculating the determinant using the expansion by minors method. | the determinant is 9 |

Fig. 3. Example of solution



evaluation of 500 questions

| | RAG | CRAG | NEW PIPLINE |
|---|---|---|---|
| accuracy | 42% | 51.2% | 62.7% |

Fig. 4. Proposed Solution in Use Case

- Choosing components:
  After careful evaluation, we selected ChatGPT 3.5 as our large language model (LLM) and opted for Text-Embedding-Ada-002 as the embedding model. We utilized VectorDB as the vector store and implemented LangChain as our chosen framework [1], [5].

Despite improvements in solving certain exercises and better answering capabilities, the LLM still lacks reasoning capabilities in some cases. To address this, the best solution is to finetune the LLM and implement our proposed approach.

## V. Conclusion

Our study demonstrated notable improvements in the LLM’s ability to tackle specific types of exercises and deliver accurate responses. Despite these advances, we identified limitations in the model’s reasoning abilities, particularly in addressing more complex and abstract problems. This suggests that while progress has been made, there is still room for enhancement to achieve robust, generalizable reasoning across diverse educational scenarios.

Looking ahead, the most promising direction involves fine-tuning the LLM and implementing the solutions proposed in this research to further elevate its performance and adaptability in educational contexts. Key areas of focus include refining our dataset creation processes to ensure comprehensive coverage of learning materials, optimizing interaction strategies to foster deeper engagement, and exploring the integration of advanced methodologies such as Corrective RAG and Hybrid RAG. These approaches have the potential to address specific challenges more effectively, such as improving retrieval accuracy, enhancing contextual understanding, and mitigating response errors.

By continually iterating and innovating on these techniques, we aim to extend the capabilities of LLMs in educational applications. This will not only improve the quality of learning experiences for students but also provide educators with more reliable, adaptive tools for facilitating personalized instruction. As the field of AI in education evolves, we are committed to pushing the boundaries of what LLMs can achieve, ultimately paving the way for more intelligent, responsive, and impactful educational technologies.

## REFERENCES

[1] N. Kandpal, H. Deng, A. Roberts, E. Wallace, and C. Raffel, “Large language models struggle to learn long-tail knowledge,” in International Conference on Machine Learning. PMLR, 2023, pp. 15 696–15 707.

[2] 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.

[3] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray et al., “Training language models to follow instructions with human feedback,” Advances in neural information processing systems, vol. 35, pp. 27 730–27 744,2022.

[4] P. Lewis, E. Perez, A. Piktus, et al., “Retrieval-augmented generation for knowledge intensive nlp tasks,” Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.

[5] Y. Gao, Y. Xiong, X. Gao, et al., “Retrieval-augmented generation for large language models: A survey,” arXiv preprint arXiv:2312.10997, 2023.

[6] H. Liu, D. Tam, M. Muqeeth, et al., “Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning,” Advances in Neural Information Processing Systems, vol. 35, pp. 1950–1965, 2022.

[7] M. Zhao, T. Lin, F. Mi, M. Jaggi, and H. Schütze, “Masking as an efficient alterna- tive to finetuning for pretrained language models,” arXiv preprint arXiv:2004.12406, 2020.

[8] Z. Fu, H. Yang, A. M.-C. So, W. Lam, L. Bing, and N. Collier, “On the effective- ness of parameter-efficient fine-tuning,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, 2023, pp. 12 799–12 807.

[9] A. Tang, L. Shen, Y. Luo, et al., “Parameter efficient multi-task model fusion with partial linearization,” arXiv preprint arXiv:2310.04742, 2023.

[10] H. Zhou, X. Wan, I. Vulić, and A. Korhonen, “Autopeft: Automatic configura- tion search for parameter-efficient fine-tuning,” Transactions of the Association for Computational Linguistics, vol. 12, pp. 525–542, 2024.

[11] Bui, T., Tran, O., Nguyen, P., Ho, B., Nguyen, L., Bui, T., and Quan, T. (2024, June). Cross-Data Knowledge Graph Construction for LLM-enabled Educational Question-Answering System: A Case Study at HCMUT. In Proceedings of the 1st ACM Workshop on AI-Powered Q & A Systems for Multimedia (pp. 36-43).

[12] Lan, Y. J., and Chen, N. S. (2024). Teachers’ agency in the era of LLM and generative AI. Educational Technology and Society, 27(1), I-XVIII.

[13] Orenstrakh, M. S., Karnalim, O., Suarez, C. A., and Liut, M. (2023). Detecting llm generated text in computing education: A comparative study for chatgpt cases. arXiv preprint arXiv:2307.07411.