MerryQuery: A Trustworthy LLM-Powered Tool Providing Personalized Support for Educators and Students

Benyamin Tabarsi[1], **Aditya Basarkar**[1], **Xukun Liu**[2], **Dongkuan (DK) Xu**[1], **Tiffany Barnes**[1]

[1]North Carolina State University
[2]Northwestern University
{btaghiz, avbasark, dxu27, tmbarnes}@ncsu.edu, xukunliu2025@u.northwestern.edu

Abstract

The potential of Large Language Models (LLMs) in education is not trivial, but concerns about academic misconduct, misinformation, and overreliance limit their adoption. To address these issues, we introduce MerryQuery, an AI-powered educational assistant using Retrieval-Augmented Generation (RAG), to provide contextually relevant, course-specific responses. MerryQuery features guided dialogues and source citations to ensure trust and improve student learning. Additionally, it enables instructors to monitor student interactions, customize response granularity, and input multimodal materials without compromising data fidelity. By meeting both student and instructor needs, MerryQuery offers a responsible way to integrate LLMs into educational settings.

Tool Prototype — https://mq.benyamintabarsi.com
User Guide — https://exploremq.benyamintabarsi.com

Introduction

Large Language Models' (LLMs) growing popularity has undeniably influenced education, although the scope and nature of their impact are under debate. Since LLMs were not originally developed for educational contexts, they often lack functionalities critical to learning environments, such as mechanisms to prevent academic misconduct, misinformation, and student overreliance. These inherent limitations have led to skepticism and confusion among educators (Sheard et al. 2024) and students (Zdravkova, Dalipi, and Ahlgren 2023; Grande, Kiesler, and Francisco R 2024), hindering LLMs' integration into educational settings.

To address these challenges, we have developed MerryQuery, an AI-powered educational platform designed to reconcile the needs of instructors and students. MerryQuery aims to offer functionalities that position it as a compelling alternative to widely used tools like ChatGPT. For educators, it provides access to student interaction logs and control over the granularity of responses, which facilitates oversight and alignment with pedagogical objectives. For students, it offers transparency by citing sources, encouraging critical thinking, and the verification of information accuracy. Additionally, MerryQuery's step-by-step response scaffolding mimics the approach human tutors use for complex tasks.

From a technical standpoint, MerryQuery leverages advanced AI methodologies such as RAG (Lewis et al. 2020) to ensure that its responses are not only contextually relevant but also grounded in course materials. Moreover, MerryQuery's integration of multimodal data processing allows the handling of complex learning resources, including documents with embedded images and tables, avoiding data loss or misinterpretation issues in LLM-generated content.

Related Work

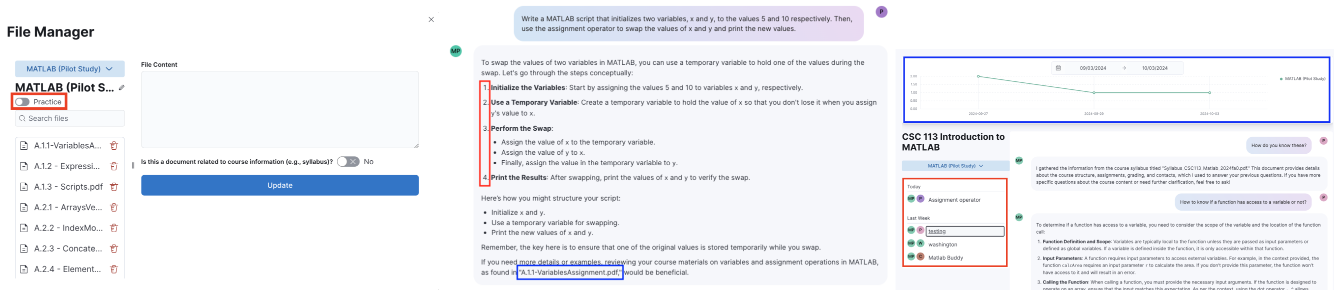
Parallel to the increasing adoption of LLMs, the number of AI-driven assistants for academic support is rising too (Liffiton et al. 2023; Liu et al. 2024). Zhong et al. introduced a RAG-based conversational agent to enhance collaborative problem-solving, which improved individual and group academic performance but had limited impact on fostering originality, possibly due to overreliance (Zhong et al. 2024). Kazemitabaar et al. developed CodeAid, an LLM-based programming assistant that provides conceptual support and code annotation without giving direct solutions. Despite achieving a high correctness and helpfulness rate, it struggled with complex tasks (Kazemitabaar et al. 2024).

While MerryQuery integrates several functionalities found in these tools, it offers features that are either absent in the existing literature or rarely found in combination. This includes support for courses beyond programming, customizable response granularity, source citation, an intuitive interface, a comprehensive logging system for researchers, and chat history access for instructors.

System Description

This section describes the components and architecture of MerryQuery. A high-level pipeline of MerryQuery is presented in Figure 2. The system uses OpenAI's API for natural language generation and LangChain to implement RAG. MongoDB stores interaction logs, document embeddings, and course data, and React.js creates a responsive and user-friendly front-end interface. Currently, MerryQuery operates on GPT-4o for generating responses.

Source Citation. MerryQuery leverages a RAG model to generate responses with relevant course material citations, as demonstrated in Figure 1b, highlighted by the blue box. This allows students to verify information, improving transparency and trustworthiness.



(a) Instructor File Manager (b) Student-AI Conversation (c) Student Chat History and Statistics

Figure 1: User Interface for Key Features of MerryQuery

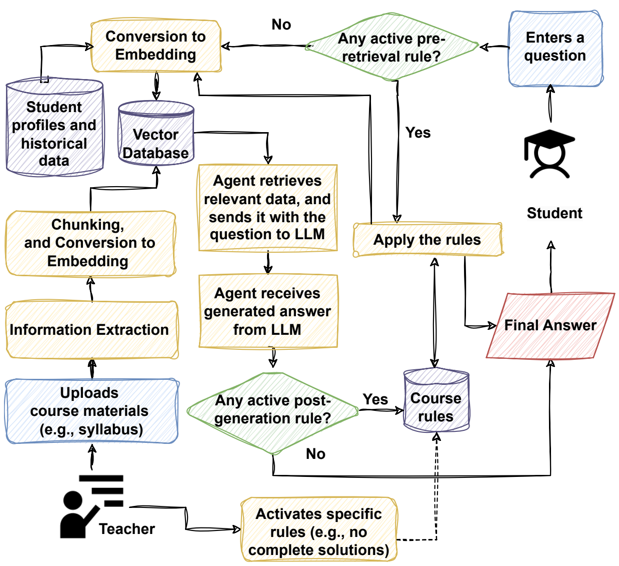


Figure 2: Pipeline of LLM-Powered Trustworthy Agents

Granularity Control: This feature, implemented through prompt engineering, allows instructors to adjust the level of assistance based on the course policy and timeline. As shown in Figure 1a, this feature offers two modes:

- **Practice Mode:** This mode supports scaffolded learning, providing hints and guidance without final answers. It encourages critical thinking and independent problem-solving, ideal during regular coursework.
- **Review Mode:** Designed for periods closer to deadlines or exams, this mode provides complete solutions with comprehensive, step-by-step explanations. It supports efficient review of materials when time is limited.

Guided Learning Dialogue Design. MerryQuery's conversation context, regardless of the mode in use, emphasizes step-by-step structure in responses to support problem-solving without undermining critical thinking. Figure 1b, highlighted by the red box, demonstrates this approach in practice mode, where the complete solution is withheld.

Chat History Monitoring. MerryQuery's chat logging system stores each student-AI interaction in MongoDB with a timestamp. These logs are accessible through the instructor interface, allowing them to review student queries, as illustrated in Figure 1c highlighted by the red box.

Usage Statistics. The platform also presents each course's real-time statistics, which are highlighted by a blue box in Figure 1c. This can help instructors monitor the frequency of conversations initiated in courses they offer.

Multimodal Data Processing. MerryQuery's current pipeline processes complex PDFs containing text, tables, and images. It first repairs PDFs using a custom function that applies garbage collection, compression, and cleanup techniques to ensure the files are properly formatted. Text is extracted using text mining techniques, while images are handled by a multi-step OCR pipeline. This includes image binarization, morphological transformations, and text extraction via Tesseract. The system then embeds both the extracted text and OCR-processed image data into a vectorized database to be retrieved through the RAG model.

Memory Mechanism for Coherent Conversations. As OpenAI's API is stateless, MerryQuery implements a custom memory mechanism to maintain context across a conversation. This mechanism identifies and retains relevant past interactions, ensuring a coherent conversation flow.

All of these features have been fully implemented and are under testing and improvement. Our initial results show that, except for minor refinements, the system performs effectively in providing accurate, course-specific assistance. Additional functionalities, such as automated grading, reinforcement learning from human feedback (RLHF), a comprehensive logging system for researchers, and personalized learning and teaching strategy generation, are under development and are expected to be integrated into MerryQuery within the next two months.

Bridging Disciplines for Advancing Learning

MerryQuery demonstrates how bridging traditionally separate disciplines can lead to meaningful advancements in everyday human experiences. By integrating LLMs, educational practices, and human-AI interaction, MerryQuery showcases the potential of interdisciplinary collaboration to build trust and enhance educational experiences for both students and educators. As further research is conducted, we plan to make MerryQuery publicly available to facilitate collaboration within the educational and research communities to gather feedback and continue refining its impact.

Acknowledgments

We would like to acknowledge the contributions of Homak Patel, Teddy Chen, and Aishwarya Radhakrishnan to MerryQuery's development.

References

Grande, V.; Kiesler, N.; and Francisco R, M. A. 2024. Student Perspectives on Using a Large Language Model (LLM) for an Assignment on Professional Ethics. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, 478–484.

Kazemitabaar, M.; Ye, R.; Wang, X.; Henley, A. Z.; Denny, P.; Craig, M.; and Grossman, T. 2024. Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, 1–20.

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

Liffiton, M.; Sheese, B. E.; Savelka, J.; and Denny, P. 2023. Codehelp: Using large language models with guardrails for scalable support in programming classes. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*, 1–11.

Liu, R.; Zenke, C.; Liu, C.; Holmes, A.; Thornton, P.; and Malan, D. J. 2024. Teaching CS50 with AI: leveraging generative artificial intelligence in computer science education. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 750–756.

Sheard, J.; Denny, P.; Hellas, A.; Leinonen, J.; Malmi, L.; and Simon. 2024. Instructor Perceptions of AI Code Generation Tools-A Multi-Institutional Interview Study. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, 1223–1229.

Zdravkova, K.; Dalipi, F.; and Ahlgren, F. 2023. Integration of Large Language Models into Higher Education: A Perspective from Learners. In *2023 International Symposium on Computers in Education (SIIE)*, 1–6. IEEE.

Zhong, X.; Xin, H.; Li, W.; Zhan, Z.; and Cheng, M.-h. 2024. The Design and application of RAG-based conversational agents for collaborative problem solving. In *Proceedings of the 2024 9th International Conference on Distance Education and Learning*, 62–68.