Quantitative Evaluation of Using Large Language Models and Retrieval-Augmented Generation in Computer Science Education

Kevin Shukang Wang
wskksw@student.ubc.ca
University of British Columbia
Kelowna, BC, Canada

Ramon Lawrence
ramon.lawrence@ubc.ca
University of British Columbia
Kelowna, BC, Canada

Abstract

Generative artificial intelligence (GenAI) is transforming Computer Science education, and every instructor is reflecting on how AI will impact their courses. Instructors must determine how students may use AI for course activities and what AI systems they will support and encourage students to use. This task is challenging with the proliferation of large language models (LLMs) and related AI systems. The contribution of this work is an experimental evaluation of the performance of multiple open-source and commercial LLMs utilizing retrieval-augmented generation in answering questions for computer science courses and a cost-benefit analysis for instructors when determining what systems to use. A key factor is the time an instructor has to maintain their supported AI systems and the most effective activities for improving their performance. The paper offers recommendations for deploying, using, and enhancing AI in educational settings.

CCS Concepts

• **Social and professional topics** → **Computing education**; • **Applied computing** → **Education**; *E-learning*.

Keywords

artificial intelligence, question answering, retrieval-augmented generation, large language model, human-in-the-loop

ACM Reference Format:
Kevin Shukang Wang and Ramon Lawrence. 2025. Quantitative Evaluation of Using Large Language Models and Retrieval-Augmented Generation in Computer Science Education. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE TS 2025), February 26-March 1, 2025, Pittsburgh, PA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3641554.3701917

1 Introduction

The widespread use of large language models (LLMs) and chatbots has generated extensive conversations on how they can be best utilized in Computer Science education. The capabilities of these systems are vast and continually improving with the ability to answer a diverse range of questions and conduct interactive help sessions for students. The risks include incorrect answers (hallucinations), privacy, bias, and cost. It is challenging for instructors to integrate AI into their courses, and this is exacerbated by the many open-source and commercial systems available. Determining when to use a commercial cloud product such as ChatGPT or develop and host custom solutions as described in prior research [8, 11, 14, 17] is a hard decision. This research evaluates open-source and commercial systems on computer science questions to determine their performance and cost. The goal is to help instructors decide whether to avoid explicitly supporting AI systems and rely on students to use what is publicly available, build and host their own AI system, or choose from the commercial products available.

There is limited data on key considerations for using, developing, and deploying AI technologies in education such as:

- **Model Selection:** Comparative data between cost-effective models and larger, more powerful ones (e.g., GPT-3.5 vs. GPT-4) and open-source models like Gemma and Phi.
- **Data Security:** Ensuring data privacy protection, complying with regulations, and maintaining trust.
- **Infrastructure Costs:** Associated costs with different models vary significantly, affecting the feasibility of their deployment based on factors like inference speed and computational requirements.
- **Viability of Local Models:** The practicality of using local models for educational use cases needs investigation to understand the potential benefits and limitations.

Other critical aspects are utilizing retrieval-augmented generation (RAG) and the instructor's role in interacting with and supporting the AI ("human-in-the-loop"). RAG provides local context (materials, assignments, deadlines, course policies) to the AI allowing for a broader range of questions to be answered. There are multiple ways to implement RAG either in commercial products or custom-developed. Instructors need to know how much time to invest in providing resources to AI to improve question answering. It is also critical to determine if the instructor has a role in validating/reviewing AI answers and how feedback can be provided to the system for continual improvement. Factors such as cost-effectiveness, speed, consistency of answer quality, and effectiveness play a role. By addressing these factors, educators and developers can make informed decisions about the most suitable AI technologies for enhancing educational outcomes.

This paper answers the following research questions:

- **RQ1:** What is the question-answering performance of a variety of open-source (e.g., Llama3 Instruct 70B, Phi3, Gemma) and commercial large language models (e.g., GPT-3.5, GPT-4) for Computer Science courses?

SIGCSE TS 2025, February 26-March 1, 2025, Pittsburgh, PA, USA

ACM ISBN 979-8-4007-0531-1/25/02
https://doi.org/10.1145/3641554.3701917

- **RQ2:** Does retrieval-augmented generation (RAG) improve the performance of both local and commercial language models in educational applications?
- **RQ3:** What approaches can instructors use to improve AI performance while being effective with their time?
- **RQ4:** What are the cost-benefit tradeoffs for deploying local LLMs versus commercial systems in educational settings?

The contribution of the paper is an evaluation of open-source and commercial models on questions from multiple Computer Science courses including the effectiveness of RAG and techniques for instructors to improve the system. Recommendations are provided based on cost, answer quality, and performance.

2 Background

Large Language Models (LLMs) may transform education through personalized learning experiences and enhanced engagement [1, 13]. There have been many research studies on the effectiveness of question answering [3, 9, 12, 16] that show AI systems' ability to answer with high accuracy many questions in computer science. Prior research selected a few systems to test performance, but there have been limited comparisons between systems to help inform what system is the best for certain instructional use cases.

Systems were developed for particular courses or universities including Harvard's CS50 [8], Jill Watson [14], BARKPLUG [11], and ChatEd [17] that provides verified references to source materials used to generate the answer. Most systems use retrieval-augmented generation to provide course-specific content to improve question-answering capabilities. An advantage of using RAG is the scale of the data is much smaller compared to general systems with vast resources from the web and Wikipedia [2, 6] and can be curated to meet the specific needs of a course [7, 11, 17].

Prompt engineering is used to provide guardrails to restrict the domains and precision of answers and improve accuracy by effectively using provided context. Although the tools for constructing RAG AI systems are expanding and becoming easier to use, there is still an engineering challenge in building, deploying, and optimizing these systems for use in production. These systems offer the promise of 24-7 student help while reducing instructor workload.

2.1 Retrieval-Augmented Generation

There are multiple techniques and technologies for implementing retrieval-augmented generation. The essential component is documents are stored in a vector database that is queried for related content for a question. This content is provided to the LLM as context. The implementations of commercial systems are proprietary, while research systems may use the Langchain framework and select from a variety of vector databases [11, 17]. ChatGPT released an Assistant API in June 2024 that allows users to easily develop a domain-specific RAG system by uploading files and editing prompts. Although users have less control over the system, as a turn-key solution, it is important to compare with custom-developed systems.

Work in the AI community to improve RAG applications uses techniques such as semantic chunking, contextual compression, and alternate embeddings [5]. RAG is highly dependent on the retrieved documents' quality, and performance is improved by retrieving better document chunks that are more related to the question. There has been limited work on specific applications of these techniques for CS education use cases with most prior systems utilizing a recursive character splitter [11, 14, 17]. There has been no comparison between commercial systems using RAG for Q&A with custom RAG implementations deployed over various LLMs.

There are specific questions unique to the educational context. Since the course materials used for RAG are much smaller than required for general question answering, it is interesting to determine how best to use these materials and techniques to improve them. An instructor may upload the syllabus, all notes, assignments, etc. which may be less than 100 documents for a typical course. Uploaded course materials may suffer from a lack of details and scope to help with student questions. Guardrails restricting the LLM to using only provided materials may prevent the LLM from answering questions it could handle without RAG.

2.2 Human-in-the-loop

It is an open question on an instructor's role with curation and feedback on question answering. The "human-in-the-loop" approach in AI [10] involves human interaction and oversight to enhance decision-making, ensure accuracy, and address ethical considerations. An instructor can be involved in the process by:

(1) **Curating**: Uploading and curating resources for RAG.
(2) **Monitoring**: During operation, supervising AI outputs and correcting errors when necessary.
(3) **Feedback**: Analyzing AI performance and providing feedback for adapting to changing conditions or new data.

It takes time for instructors to curate resources and monitor AI outputs, and those activities reduce the time-saving benefits of AI automation. Systems can make this process more efficient by reporting on questions that have incorrect answers, allowing instructors to provide feedback to prevent future incorrect answers and inform an instructor when student questions are outside of the scope of the materials provided using RAG.

2.3 Evaluating LLMs

Evaluating Large Language Models (LLMs) has been performed with datasets like FEVER [15] and HotPotQA [19]. The Fact Extraction and Verification (FEVER) dataset evaluates a model's ability to verify claims against a set of facts from Wikipedia. HotPotQA focuses on multi-hop question answering, evaluating a model's ability to retrieve and reason over multiple documents [19]. There is no standardized data set and testing framework for education Q&A, which would be valuable as these generic testing frameworks do not capture the questions asked in education.

3 Methodology

Multiple open-source and commercial LLMs were evaluated on question data from four CS courses. These systems were tested:

- gemma2 (Published 2024-06-27)
- llama3 (Published 2024-04-28)
- phi3 (Published 2024-04-23)
- gpt-3.5-turbo-0125 (Published 2024-01-25)
- gpt-4o (Published 2024-05-13)
- gpt-4o-mini (Published 2024-07-18)

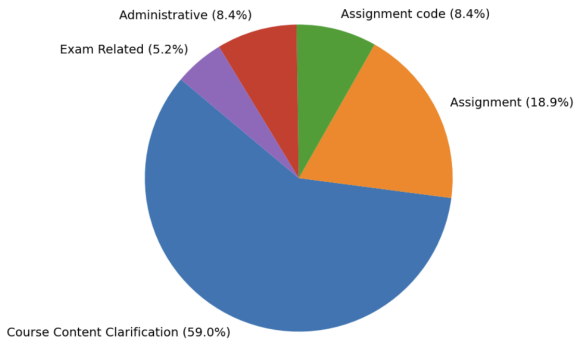

Figure 1: Distribution of Question Categories

Category
Content clarification
Question
What are some data models other than the relational model?
Ground Truth Answer
Some data models other than the relational model include the hierarchical model, object-oriented model, XML, graphs, key-value stores, and document models.

Figure 2: Example test case

The non-gpt systems were hosted on a dual Intel Xeon Platinum 8462Y+ with 1TB of RAM and an Nvidia RTX 6000 GPU. Local hosting costs were estimated using pricing for GPU renting where pricing is available at $1 per hour[1].

3.1 Evaluation Dataset

The evaluation framework assesses model performance using a dataset first generated by LLM and edited by instructors and teaching assistants. The questions are based on content of four courses: CS1, CS2, DB1 (intro), and DB2 (advanced). There are 241 test cases with 83% being general questions that do not require context. Each question has a category, question, and expected answer. The distribution of question categories is in Figure 1, and an example test case is in Figure 2.

3.2 Testing Framework

The testing framework was implemented in Python. For each test case, the code used an API to submit the question and retrieve the answer. The response time was measured. The answers were evaluated with several metrics:

- **TF-IDF based cosine similarity:** TF-IDF vectors are generated on both the ground truth and generated answer and cosine similarity computed between the two vectors.
- **Answer similarity:** This metric uses a MiniLM text embedding model implemented using MiniLM-L6-v2[2] to embed text strings and check similarity between strings based on semantic similarity instead of string similarity [18]. As the model employs deep self-attention mechanism, the embedding captures semantic similarity better when sentences may vary in length and wording.
- **Answer Correctness:** We evaluate answer correctness using the Ragas library in combination with the llama3:70b-instruct LLM base model. This process [4] involves comparing the generated answers to the ground truth by having the LLM categorize each statement as True Positive (statements present in both answer and ground truth), False Positive (statements in answer but not in ground truth), or False Negative (statements in ground truth but missing from answer). This LLM-as-a-judge approach has been validated to closely align with human evaluators [20].
- **Average inference speed:** to generate a response with lower values preferred.
- **Question cost:** is the estimated average cost to answer a question. This is the actual billed cost for GPT systems and is calculated for locally hosted systems by dividing query answer time by the cost per GPU hour.

Original chunk:
length, color, and filled), creates that object, and then displays its state (side, color, area, etc.). Assume users enter valid inputs. Create a new object using the clone method and compare the two objects using the compareTo method. Hint: Add “throws CloneNotSupportedException” to the header of the clone method as well as the main method in your test program. You will learn more about exception handling in next lecture and lab.

AI trimed chunk:
Object Cloning and Comparison: To create an object with specific attributes (e.g. length, color, filled), display its state, and compare it with a cloned object, you can utilize the `clone` and `compareTo` methods. Remember to add "throws CloneNotSupportedException" to the header of the `clone` method and the main method in your test program to handle exceptions.

Figure 3: Example of AI Edit

3.3 RAG Optimizations and Human Feedback

For commercial systems, course documents are uploaded as PDFs into the system. There is no user control of the RAG processes for these systems. For local hosted models, the base RAG system implementation encoded course documents using PGvector and OpenAI embeddings. Chunking is performed first by page, then by 1000 characters with a 20-character overlap using a recursive text splitter similar to prior work [11, 14, 17]. The prompt used in all systems is designed to incorporate relevant context effectively.

Two RAG optimizations are explored:

- **AI-assisted content curation:** have course contents automatically summarized by AI (see Figure 3)
- **Question reuse:** encodes answers to questions and stores them as content for use by RAG

Three RAG databases are evaluated:

- **Content database:** contains the course materials
- **AI-edited database:** has the course materials edited by AI
- **Content and question database:** has course materials and question-answer pairs

[1] https://www.runpod.io/gpu/6000-ada
[2] Model available at: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Two types of prompts are evaluated:

- **With heavy guardrails:** "RULES: 1) If you don't know the answer just say that "I don't know", do not try to make up an answer. 2) If you are unsure of the answer, you shall PREFACE your answer with "I'm not sure, but this is what I think." 3) Try to be as concise as possible. 4) Do not use any other resources apart from the context provided to you."
- **With light guardrails**: "You are an educational assistant. Here are some rules for question answering: 1) Try to be as concise as possible. 2) Refer to context as you see fit."

ChatGPT's Assistant API controls all aspects of the documents used to answer queries and prompt engineering for providing context. Thus, the Assistant API prompt is adapted to remove the last rule regarding how to use context.

Testing the content and AI-edited database uses the question set. Testing the database containing question answers is designed to determine if instructors providing feedback on correct answers can help answer similar questions. These similar questions ($N = 723$) are generated using an LLM prompted to mimic student questions and are the test set of the experiment, not the original question set.

4 Results

4.1 LLM Performance

Table 1 contains data on the performance of non-RAG systems for **RQ1** on questions that do not require context. The numbers in bold are the best for that metric. The results show that gpt-4o-mini (followed closely by gpt-3.5) has the best cost/performance tradeoff with high performance according to the metrics (similarity, correctness, TF-IDF) while having the lowest cost. This outperforms locally hosted LLMs, even with the low cost of $1/hour for the GPU server and only costing by the time to answer the question set. Actual hosted costs would be higher as a production system would need to be available 24-7 and include costs related to system administration and maintenance. The most advanced models such as gpt-4o demonstrate higher performance than open-source models, but many scores are not significantly different and may not be noticeable to users. For inference speed, llama3:8b has similar performance to gpt-3.5, while llama3:70b takes significantly longer.

4.2 RAG Performance

Table 2 contains data on the performance of LLM systems for **RQ2** when utilizing course-specific data provided via RAG and a prompt with light guardrails. All systems use the same base RAG system except for ChatGPT's Assistant API. Course content improves performance for all models with a larger relative impact for the open-source models. GPT 3.5 has the best cost-performance tradeoff. Utilizing the Assistant API does not outperform locally hosted RAG and comes with a much higher cost. However, it does not require local hosting of the vector database. The local vector database costs are excluded since the database was hosted on the same GPU machine with minimal computational overhead. Adding context increases the response time for all systems.

The data for questions that require context is in Table 3. Without providing context, all LLMs have poor performance (average for TF-IDF=0.276, Similarity=0.639, Correctness=0.320). Providing context increases performance, but not to the level of general questions. This is often due to the context not providing sufficient information to answer the questions or failure to retrieve the appropriate content.

There is an interesting performance difference for individual courses as shown in Table 4. This experiment considers only questions that do not require context and averages the three metrics for all LLMs. The performance of CS1/CS2 is higher than DB1/DB2 courses as there are more training materials for CS1/CS2 available to the LLMs. Using RAG to provide course-specific context improves average scores for CS1 and CS2 marginally as most concepts in CS1/CS2 are well covered by LLMs. Performance for DB2 improved only slightly even though the context should have helped more significantly for an advanced course. Analyzing the test cases revealed that the uploaded materials often did not always have sufficient information to answer the question. The performance improvement for DB1 is due to the course context allowing the LLM to tailor general answers to how the course was taught. For example, an open-ended question like "What are the benefits and limitations of key-value stores?" has an answer specific to the level covered in the course compared to a generic answer.

4.3 Instructor Optimizations

RQ3 asked what time-effective approaches instructors can use to improve AI performance. The results from RAG versus non-RAG show that for general question answering the uploading of content has some impact on the ability of the LLM to answer questions with a clear benefit in answering context-specific questions about the course that would be contained in a syllabus. Students can get good answers to most questions without the instructor utilizing RAG. The most important content to provide is the syllabus and answers to common questions.

Table 5 contains data comparing recursive chunking and AI trimming of uploaded content. The context-related metrics in the table are obtained from the Ragas testing framework [4]. This shows relative consistency in answer quality, but with significant context-related improvements. Utilizing smart AI editing of chunks improves readability for humans, although the impact on RAG performance is marginal. Instructor time spent on prompt engineering, chunking optimizations, and content curation has marginal impacts on performance for the time spent.

One key feature not well-supported by commercial systems is helping instructors analyze questions that are answered poorly when using the RAG content. This is especially important if the LLM has guardrails restricting its answers to the specified domain. Table 6 has data on the impact of guardrails and utilizing previously answered questions with RAG. In this experiment, the test case question-answer pairs were inserted into the RAG database (**Content and Question** in the table) as data sources. The test set used AI-generated similar question-answer pairs.

Using previously answered questions for RAG demonstrated excellent performance. Similar questions were mapped with high precision to instructor-answered questions and provided the necessary context. Utilizing instructor answers has the potential to significantly improve performance with reasonable effort as most instructors have a frequently asked set of questions for their courses.

Model	TF-IDF	Similarity	Correctness	Response Time (s)	Cost per 1000 queries
gpt-3.5-turbo-0125	0.466	0.844	0.531	**1.25**	$0.12
gpt-4o	0.454	0.840	**0.540**	2.25	$0.20
gpt-4o-mini	**0.467**	**0.849**	0.531	2.38	**$0.08**
llama3:70b	0.446	0.835	0.523	7.60	$2.10
llama3:8b	0.416	0.822	0.482	1.48	$0.41
phi3:14b	0.395	0.810	0.502	2.99	$0.83
phi3:3.8b	0.393	0.825	0.508	1.30	$0.36
gemma2:9b	0.383	0.807	0.514	1.66	$0.46

Table 1: Comparison of Non-RAG Systems

Model	TF-IDF	Similarity	Correctness	Response Time (s)	Cost per 1000 queries
gpt-3.5-turbo-0125	0.497	0.851	0.578	3.95	**$0.37**
gpt-4o	**0.525**	**0.854**	**0.588**	5.15	$5.48
gpt-4o assistantAPI	0.512	0.853	0.583	9.10	$52.07
gpt-4o-mini assistantAPI	0.485	0.850	0.544	13.04	$1.50
llama3:70b	0.511	0.837	0.576	7.29	$2.10
gemma2:9b	0.481	0.830	0.580	2.24	$0.62
llama3:8b	0.474	0.821	0.562	**2.00**	$0.56
phi3:14b	0.460	0.844	0.564	2.69	$0.74
phi3:3.8b	0.428	0.814	0.543	2.01	$0.56

Table 2: Comparison of RAG Systems

Model	TF-IDF	Similarity	Correctness
gpt-3.5-turbo-0125	0.416	0.714	0.519
gpt-4o	**0.448**	0.718	0.539
gpt-4o-mini	0.431	**0.749**	0.556
llama3:70b	0.428	0.724	0.522
llama3:8b	0.398	0.691	0.472
phi3:14b	0.386	0.674	0.482
phi3:3.8b	0.367	0.702	0.486
gemma2:9b	0.425	0.729	**0.557**

Table 3: Context Specific Questions with RAG

Course	Score	RAG Score
CS1	0.6000	0.6435
CS2	0.6131	0.6214
DB1	0.5645	0.6748
DB2	0.5715	0.5974

Table 4: Average Scores Across Different Courses

Guardrails preventing student questions in out-of-scope content or restricting the precision of answers did not have a significant impact on performance with the test set. However, with guardrails some LLMs provide no answers to some questions. This was not a factor for GPT systems, and the most affected system was phi3:3.8b. LLMs respond differently to prompt engineering guardrails, affecting the user experience based on how often these guardrails prevent effective answers.

4.4 Deployment Costs

A fundamental decision is whether to use locally hosted or commercial systems like ChatGPT (**RQ4**). As of July 2024, GPT-3.5 and GPT-4o-mini are low-cost options with high performance. The GPT systems get more expensive when using RAG. ChatGPT's Assistant API is about 10 times the cost of querying the same GPT model with a locally hosted RAG. RAG also increases response time. Local hosting of the RAG component has cost benefits and does not require the GPU resources required to run the LLM.

Building and maintaining a locally hosted solution has benefits for flexibility and privacy but requires engineering support. Hardware and energy costs are also significant factors when deploying an LLM. Even when scaled across multiple courses in an institutional infrastructure, the costs for locally running and supporting the LLM can be substantial.

4.5 Recommendations

The experimental results show that both open-source and commercial LLMs are effective at answering computer science questions. The GPT versions had higher performance metrics, but by utilizing RAG effectively open-source models are competitive. Instructors can use locally hosted LLMs which have advantages for privacy and confidentiality. If data security is a concern, Gemma2 is an excellent option with or without RAG. However, this might change as it is one of the newest models at the time of this work.

The increased performance of RAG also significantly increases cost as more tokens are sent to the LLM. This is a primary factor for GPT but also impacts local models as it requires extra computation time and deployment of the vector database.

Chunking Method	TF-IDF	Time (s)	Similarity	Relevancy	Recall	Precision
AI trimmed	0.48	7.3	0.84	0.49	0.68	0.93
Recursively chunked	0.49	7.2	0.82	0.21	0.59	0.84

Table 5: Comparison of AI trimmed and traditional chunking methods for llama3:70b

Database	TF-IDF	Similarity	Correctness	Prompt Type
Content only	0.311621	0.677357	0.500571	Heavy guardrail
Content and Question	0.439530	0.737213	0.655102	Heavy guardrail
Content only	0.296113	0.664729	0.463972	Light guardrail
Content and Question	0.437722	0.736551	0.640519	Light guardrail

Table 6: Test results on similar questions with Llama3:70b

For instructors who do not have institutional infrastructure support, deploying and maintaining these systems is a significant cost and effort compared to utilizing GPT-4o with the Assistant API which is a packaged solution. Although its per query cost is the highest, the absolute amount is small when compared to costs of instructor time. Another consideration is whether it is valuable to support a course-specific LLM at all. For CS1/CS2 courses, except for course-specific questions mostly related to the syllabus, general LLMs provide high-quality answers and can be used easily by students without support. The effort of providing resources to an LLM may not be worth the instructor's time compared to the number of questions asked by students.

RAG in education has issues when retrieval fails, often due to insufficient relevant content, as unsuccessful retrieval can simultaneously degrade answer quality, increase response time, and raise operational costs. Thus it's very important to consider completeness of the data store before deploying RAG assistants.

5 Threats to Validity

The evaluation has some limitations that may affect the findings. The test cases were generated by teaching assistants with the assistance of large language models (LLMs). The questions are not student questions posed to the LLM to avoid issues with student privacy, so they may not fully capture the full range of student queries and should be interpreted with caution. The questions are representative of common questions asked in the courses. All course materials are used for the four CS courses, but more courses and a diversity of courses would be valuable to test. Standardized test cases and course materials would allow for rigorous, repeatable testing. This work will release the test set for use by others, which will hopefully contribute to the creation of a community test database.

LLMs are constantly changing, so the costs quoted and performance captured are a snapshot as of July 2024. However, the relative comparisons of GPT versus local LLM hosting will likely remain consistent. GPT systems may become even more cost-effective compared to local hosting as cloud providers benefit from economies of scale. The new release of GPT-4o-mini is a good example of the rapid improvement and lower costs of commercial LLMs.

The study focused solely on question-answering tasks. It does not encompass other applications of LLMs in education, such as tutoring, facilitating discussion posts, providing real-time feedback or help with coding, or supporting collaborative learning activities. Future studies could consider a broader range of use cases.

6 Conclusions and Future Work

The study evaluated large language models and retrieval-augmented generation for computer science questions. The key findings are:

- **LLM Performance**: Advanced models like GPT-4 outperform open-source models in Q&A tasks. However, the performance gap is not substantial suggesting that cost-effective and locally hosted models can be viable alternatives depending on specific needs.
- **Impact of RAG**: Implementing RAG enhances the ability of LLMs to answer context-specific questions accurately. This improvement is particularly noticeable in models with integrated course materials and pre-answered question databases and allows open-source models to close some of the gap with GPT-4.
- **Instructor Involvement**: Instructor content curation has mixed benefits for the time spent. High-priority resources are the syllabus and verified question answers.
- **Cost-Benefit Analysis**: ChatGPT systems have high per-query costs but relatively low absolute costs compared to local hosting. Supporting an LLM may have limited benefit to instructors depending on the volume of questions asked.

The RAG implementation, test cases, evaluation code, and results are at https://github.com/ubco-db/LLM_education_benchmark.

For future work, a key goal would be to create and distribute public test data sets to allow for experimentation and comparison of approaches. Future research should investigate the application of LLMs and RAG beyond Q&A, such as supporting student discussions and providing personalized feedback. Developing a comprehensive set of metrics that accurately reflect the effectiveness of these applications will be essential for meaningful evaluations. Research should focus on helping instructors curate content and monitor performance to ensure that AI is integrated effectively when delivering a course.

References

[1] Paul Denny, James Prather, Brett A Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N Reeves, Eddie Antonio Santos, and Sami Sarsa. 2024. Computing education in the era of generative AI. *Commun. ACM* 67, 2 (2024), 56–67.

[2] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL] https://arxiv.org/abs/2312.10997

[3] Chuqin Geng, Yihan Zhang, Brigitte Pientka, and Xujie Si. 2023. Can ChatGPT Pass An Introductory Level Functional Language Programming Course? arXiv:2305.02230 [cs.CY] https://arxiv.org/abs/2305.02230

[4] Exploding Gradients. 2023. RAGAS: Retrieval Augmented Generation for Answer Synthesis. https://github.com/explodinggradients/ragas.

[5] LangChain. 2023. LangChain Documentation - Introduction. https://python.langchain.com/v0.1/docs/get_started/introduction/ Accessed: 2024-07-11.

[6] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[7] Chang Liu, Loc Hoang, Andrew Stolman, and Bo Wu. 2024. HiTA: A RAG-Based Educational Platform that Centers Educators in the Instructional Loop. In *International Conference on Artificial Intelligence in Education*. Springer, 405–412.

[8] Rongxin Liu, Carter Zenke, Charlie Liu, Andrew Holmes, Patrick Thornton, and David J. Malan. 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 (SIGCSE 2024)*. ACM, New York, NY, USA, 750–756. https://doi.org/10.1145/3626252.3630938

[9] Kamil Malinka, Martin Peresni, Anton Firc, Ondrej Hujnk, and Filip Janus. 2023. On the Educational Impact of ChatGPT: Is Artificial Intelligence Ready to Obtain a University Degree?. In *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education*. ACM, 47–53. https://doi.org/10.1145/3587102.3588827

[10] E. Mollick. 2024. *Co-Intelligence: Living and Working with AI*. Penguin Publishing Group. https://books.google.ca/books?id=r13gEAAAQBAJ

[11] Subash Neupane, Elias Hossain, Jason Keith, Himanshu Tripathi, Farbod Ghiasi, Noorbakhsh Amiri Golilarz, Amin Amirlatifi, Sudip Mittal, and Shahram Rahimi. 2024. From Questions to Insightful Answers: Building an Informed Chatbot for University Resources. arXiv:2405.08120 [cs.ET] https://arxiv.org/abs/2405.08120

[12] OpenAI. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[13] James Prather, Paul Denny, Juho Leinonen, Brett A Becker, Ibrahim Albluwi, Michelle Craig, Hieke Keuning, Natalie Kiesler, Tobias Kohn, Andrew Luxton-Reilly, et al. 2023. The robots are here: Navigating the generative ai revolution in computing education. In *Proceedings of the 2023 Working Group Reports on Innovation and Technology in Computer Science Education*. 108–159.

[14] Karan Taneja, Pratyusha Maiti, Sandeep Kakar, Pranav Guruprasad, Sanjeev Rao, and Ashok K. Goel. 2024. Jill Watson: A Virtual Teaching Assistant powered by ChatGPT. arXiv:2405.11070 [cs.AI] https://arxiv.org/abs/2405.11070

[15] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for Fact Extraction and VERification. arXiv:1803.05355 [cs.CL] https://arxiv.org/abs/1803.05355

[16] Kevin Wang, Seth Akins, Abdallah Mohammed, and Ramon Lawrence. 2023. Student Mastery or AI Deception? Analyzing ChatGPT's Assessment Proficiency and Evaluating Detection Strategies. arXiv:2311.16292 [cs.CY] https://arxiv.org/abs/2311.16292

[17] Kevin Wang, Jason Ramos, and Ramon Lawrence. 2023. ChatEd: A Chatbot Leveraging ChatGPT for an Enhanced Learning Experience in Higher Education. arXiv:2401.00052 [cs.CY] https://arxiv.org/abs/2401.00052

[18] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems* 33 (2020), 5776–5788.

[19] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. arXiv:1809.09600 [cs.CL] https://arxiv.org/abs/1809.09600

[20] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *Advances in Neural Information Processing Systems* 36 (2024).