RAMO: Retrieval-Augmented Generation for Enhancing MOOCs Recommendations

Jiarui Rao[1,†], Jionghao Lin[1]

[1]Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213

Abstract

Massive Open Online Courses (MOOCs) have significantly enhanced educational accessibility by offering a wide variety of courses and breaking down traditional barriers related to geography, finance, and time. However, students often face difficulties navigating the vast selection of courses, especially when exploring new fields of study. Driven by this challenge, researchers have been exploring course recommender systems to offer tailored guidance that aligns with individual learning preferences and career aspirations. These systems face particular challenges in effectively addressing the "cold start" problem for new users. Recent advancements in recommender systems suggest integrating large language models (LLMs) into the recommendation process to enhance personalized recommendations and address the "cold start" problem. Motivated by these advancements, our study introduces RAMO (Retrieval-Augmented Generation for MOOCs), a system specifically designed to overcome the "cold start" challenges of traditional course recommender systems. The RAMO system leverages the capabilities of LLMs, along with Retrieval-Augmented Generation (RAG)-facilitated contextual understanding, to provide course recommendations through a conversational interface, aiming to enhance the e-learning experience.

Keywords

Retrieval-Augmented Generation (RAG), Personalized Learning, Recommender Systems, Artificial Intelligence

1. Introduction

Massive Open Online Courses (MOOCs) gently facilitate access to learning for a diverse global audience [1]. By providing an extensive range of courses through an easily accessible online platform, MOOCs not only enhance individual learning and development but also enrich the broader educational community [2]. However, the diverse categories of courses across disciplines can often overwhelm students when they step into new fields of study [3]. Selecting the right courses that align with both personal interests and academic requirements is crucial, as improper choices may lead to wasted time, and resources, and a lack of fulfillment in one's educational journey (Generated by AI Tool ChatGPT) .

To resolve this, researchers have developed course recommender systems using advanced algorithms to offer tailored guidance that aligns with individual learning preferences [4]. Many existing implementations of recommendation systems have demonstrated significant benefits, such as enhancing personalized learning experiences and improving student engagement, as highlighted by a recent study [5]. However, these systems also face critical limitations, particularly the "*cold start*" problem, which occurs when trying to make recommendations for new users with limited historical data [6]. Though previous research proposed a more complex framework—a novel meta-learning heterogeneous information networks approach [7]—to address the "*cold start*" recommendation issue, the approach faces the challenge of high computational complexity, which is not scalable for large-scale MOOCs platforms.

In response to address the limitations of prior work in recommendation systems, where the recommendations lack sufficient personalization and interaction with users, researchers have proposed integrating large language models (LLMs) into course recommendations [8]. This approach enhances recommendation accuracy and personalization by leveraging user history and conversational prompts. Recent frameworks like GPT4Rec [9] and Chat-Rec [10] demonstrated the potential of LLMs in improving course alignment with learners' interests and interaction. However, LLMs can sometimes generate misleading or outdated information. To counteract these shortcomings, one possible solution is the integration of Retrieval-Augmented Generation (RAG) with LLMs [11].

RAG [12] is a process that optimizes the output of LLMs by extending their robust capabilities to cater specifically to distinct domains or an organization's internal knowledge base, eliminating the need for retraining the model [13]. The use of RAG in recommendation systems enhances the adaptability of LLMs, ensuring that recommendations remain current and contextually relevant [11]. This advancement paves the way for more precise and targeted course recommendations that adapt to changes in educational content and learner preferences. Despite these improvements, there is a noticeable gap in research specifically focused on using LLMs in course recommender systems, particularly in addressing the "*cold start*" problem where the system lacks a user's profile. Thus, our study aims to investigate the potential of LLMs, particularly those enhanced by RAG, in providing course recommendations tailored to individual user needs. We introduce a course recommender system, **RAMO** (**R**etrieval-**A**ugmented **G**eneration for **MO**OCs), which employs a RAG-based LLM model (refer to Figure 1). RAMO leverages RAG's advantage to improve the quality of course recommendations, addressing and mitigating common issues associated with LLMs especially in "*cold start*" problem.

2. Related Works

2.1. Course Recommender Systems

Course recommender systems are essential in educational technology, helping students choose courses that align with their interests and academic goals. Many prior studies have employed collaborative filtering methods to build course recommender systems [14, 15, 16]. For instance, Schafer

Educational Datamining '24 Human-Centric eXplainable AI in Education and Leveraging Large Language Models for Next-Generation Educational Technologies Workshop Joint Proceedings, July 13, 2024, Atlanta, GA

*Corresponding author.

†These authors contributed equally.

jiaruira@andrew.cmu.edu (J. Rao); jionghal@andrew.cmu.edu (J. Lin)

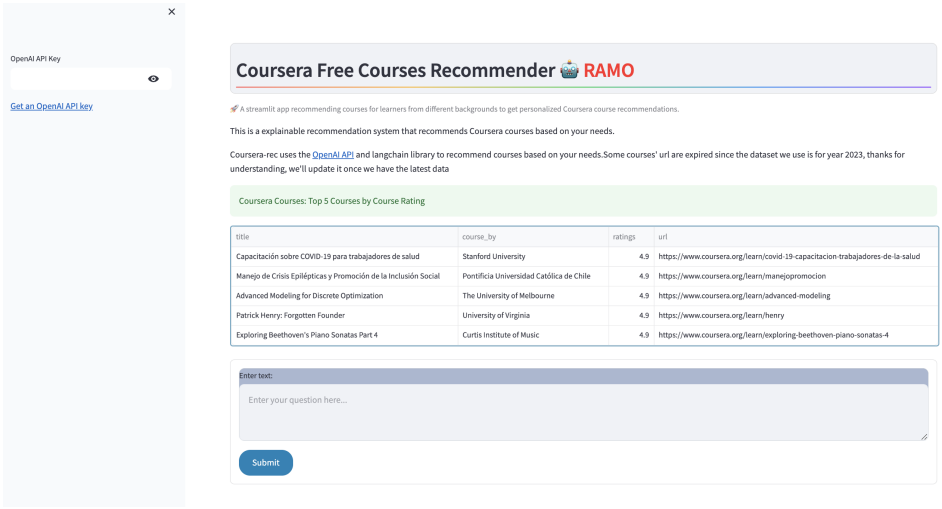

Figure 1: Interface of the Retrieval-Augmented Generation for MOOCs (RAMO) system

et al. [15] proposed a recommender system that suggested courses based on the preferences of similar users. A more recent example by Koren et al. [16] developed advanced collaborative filtering techniques to enhance course recommendation accuracy. However, a significant issue arises when recommending courses for new users, as there is no historical data available for these individuals—this is known as the "*cold start*" problem [17]. To address this challenge, a recent study by Wu et al. [17] leveraged large language models (LLMs), which utilize extensive pre-trained knowledge from web datasets, demonstrating potential in overcoming the cold start problem. Despite the advancements in LLMs, their integration into course recommendation systems remains largely unexplored, presenting an opportunity for future research to innovate and improve student course selection processes.

2.2. Large Language Models in Education

Large language models (LLMs) like ChatGPT, trained on extensive datasets, have the ability to generate human-like text and respond to questions with exceptional precision [18, 19]. Many studies have highlighted the potential of LLMs in educational applications, leveraging their capabilities to enhance various aspects of teaching and learning. For example, Kabir and Lin [20] developed an adaptive practicing system utilizing ChatGPT to generate personalized questions and feedback, demonstrating LLMs' potential in facilitating tailored educational interactions. Researchers investigated multiple GPT models on their ability to generate tailored learning materials and provide instant feedback on student errors, enhancing personalized learning experiences [21]. Huber et al. [22] demonstrated the use of LLMs in creating interactive, conversational systems that assist both students and teachers by providing adaptive learning support and resources. Moreover, LLMs are also used in generating automatic feedback for students [23, 24], handling sparse learner performance data [25] from intelligent tutoring systems, predicting learning performance [26], and supporting tutor training session [27].

2.3. Retrieval-Augmented Generation in Education

Retrieval-augmented generationn (RAG) has emerged as a significant technique to enhance the effectiveness of educational tools powered by LLMs. For example, a study [28] integrated textbook content into LLM prompts via RAG improved the quality of responses in interactive question-answering (QA) scenarios for middle-school math students, and demonstrated that students generally prefer responses generated by RAGs. RAG has also been employed in programming education to generate improved feedback for student's completion of coding tasks [29], by incorporating transcriptions of lecture recordings and using timestamps as meta-information, RAG reduces hallucinations and ensures the use of accurate technical terms. Moreover, RAG has been utilized to assess novice math tutors' use of social-emotional learning strategies [30], they proved that RAG-enhanced prompts demonstrated more accurate and cost-effective performance compared to other prompting strategies by providing relevant external content. This application highlights the potential of RAG in developing personalized tutor training programs and enhancing the overall effectiveness of tutored learning.

While traditional course recommender systems have laid the groundwork for personalized education, the integration of LLMs and techniques such as RAG offers unprecedented opportunities for enhancing educational experiences. These advanced methods address limitations of earlier approaches and pave the way for more sophisticated and effective educational tools, inspiring us to utilize RAG in developing our course recommender system.

3. Method

3.1. Dataset

In this study, we utilized the "Coursera Courses Dataset 2021"[1] from Kaggle. The dataset, scraped from Coursera's publicly available information in September 2021, contains a variety of courses that feature comprehensive details such as

[1] https://www.kaggle.com/datasets/khusheekapoor/coursera-courses-dataset-2021

skill requirements, difficulty levels, and direct course links. It provides a robust knowledge base for our RAMO system, enabling it to suggest courses tailored to students' specific skills and educational needs. This dataset effectively supports our objective to enhance accessibility and personalized learning through course recommendations. We first cleaned the dataset to remove meaningless symbols and duplicate rows, and it has 3,342 non-duplicate courses in total after data-cleaning, with 6 columns:

- **Course Name**: The title of the course.
- **University**: The institution offering the course.
- **Difficulty Level**: The level of complexity of the course content.
- **Rating**: The average rating given by learners.
- **URL**: The web address where the course can be accessed.
- **Description**: A brief overview of what the course covers.
- **Skills**: The specific abilities or knowledge areas that the course aims to develop.

3.2. Recommendation System Design

3.2.1. Prompt Design

The "cold start" problem, where systems lack user historical data, is a significant challenge in recommendation systems. Both traditional course recommender algorithums like content-based and collaborative-filtering algorithms and LLM-based system recommendation systems struggle with this issue. However, our RAG-based solution addresses this by using a 'prompt template' in the back-end. This template guides RAMO to generate relevant responses even when no user-specific data is available, as detailed in Table 1. The RAMO system can provide meaningful recommendations from the outset, unlike non-RAG-based recommender systems, which lack a retrieval process and prompt-based customization. The prompt to our retriever (i.e., to retrieve the relevant docs from the databases) is called the 'prompt template', which is shown in Table 1. The prompt to our generator is composed with three parts: 1) `User Question`, 2) `Prompt Template`, and 3) `Search Results` (the context of the retrieved relevant documents). We also added the uplifting adverb '*fantastic*' to the prompt template, to elevate it with Emotional Intelligence since ChatGPT is designed to recognize patterns in language, including those associated with emotions [31].

Table 1
Overview of interaction prompt structure

Prompt Template *You are a fantastic Coursera course recommender. Use the following pieces of context to answer the question and recommend relevant courses to the user. If the user doesn't specify their requirements, you can just recommend some courses that are most popular in the system based on their ratings and difficulty levels. You only need to provide the course title to the user. Also, please pay attention to how many courses the user wants you to recommend. If you don't know the answer, just say "I don't know".*
Context Retrieved course data
User Question User's specific question to the generator

3.2.2. Integration of RAG approach

As shown in Table 2 below, we employed several LLMs to build our course recommender system. We provide a list of the LLM models we used, along with details on their associated costs and token limits. The token limit refers to the maximum number of tokens (a token represents about 3/4 of a word or four characters, according to Open AI [32]) that the model can process in a single input. While some models, like Llama 2 and Llama 3, are free to use on small-scale dataset, due to their open-source nature, others may incur costs based on usage or subscription plans [33].

Table 2
Cost and token limit of models we used

LLM Model	Output Cost	Token Limit
`GPT-3.5 Turbo`	0.50 per 1M tokens	4,096 tokens
`GPT-4`	30.00 per 1M tokens	8,192 tokens
`Llama-2`	Free	4,096 tokens
`Llama-3`	Free	8,000 tokens

We then leveraged the RAG approach to enhance the system's understanding of the user context. As shown in Figure **??**, RAG consists of two primary components: the *retriever* and the *generator*. The retriever aims to enhance the prompt templates, which 'augment' the retrieval process, tailoring it to specific user queries. The knowledge base used for the retrieval process can contain any format of course data (e.g., csv, pdf, and json), providing a flexible and rich source of information for generating responses and we used the largest MOOC platform—coursera's course dataset in csv format as the knowledge base. The dataset was transformed into text embeddings and stored in the vector database. These embeddings were then used to find high-quality, relevant information, which was incorporated into the prompt for the generator. Here we use OpenAI embedding model (`text-embedding-ada-002` [34]) to tokenize the course data and store the embeddings in vector store, considering its advantage over BERT (Bidirectional Encoder Representations from Transformers) [35], while OpenAI embeddings [34] offer better generalization and contextual understanding [36], making them more suitable for diverse educational content. The generator is powered by LLMs, which generate the textual contents based on the engineered prompts. To facilitate user's interaction with the system, we make the recommendation process to be completed via conversational manner.

The interface of our recommender system is shown in Figure 1, where we listed 5 default courses based on their ratings in the dataset on the web page to make it more user-friendly. As for the implementation of the system, we use `GPT-3.5 Turbo`, selected for its robust integration with the `LangChain` [37] framework—a platform designed to streamline the implementation of language models in application-specific contexts. This setup allows the system to dynamically retrieve relevant documents and generate responses tailored to user inputs, as illustrated in the workflow in Figure 2.

3.2.3. Comparative Analysis

To evaluate the performance of our system, we conducted a series of tests by providing different prompts representing various user needs to RAMO. This allowed us to explore

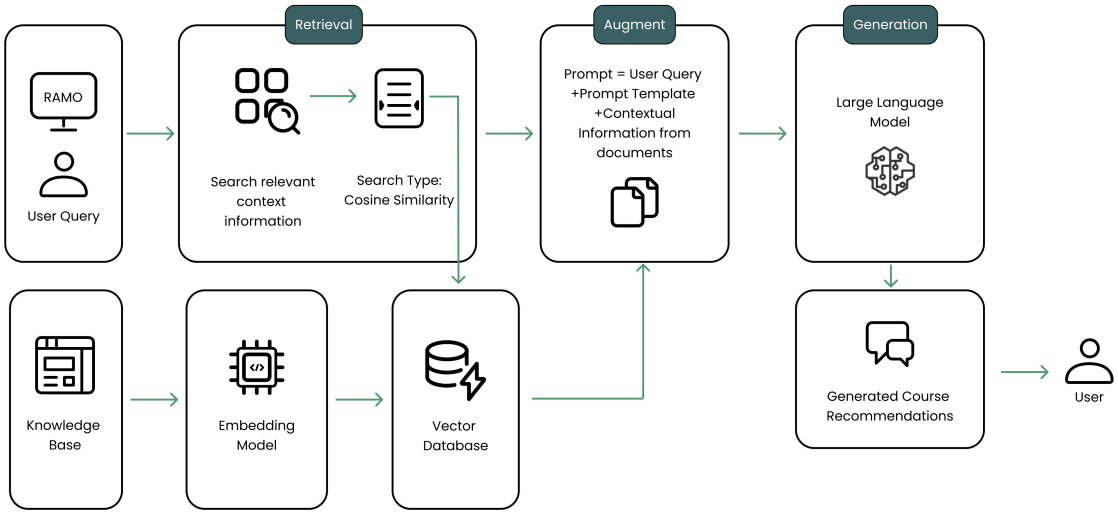

Figure 2: Workflow for the RAMO System

its ability to deliver course recommendations based on the outputs generated in response to varied user prompts.

LLM vs. Non-LLM. We explored both the relevance of the recommended courses to the user's interests and responding time (the time it takes to generate a response) of the LLM-based recommender system compared to non-LLM course recommender systems (e.g., course recommender system using collaborative filtering and content-based approaches), focusing particularly on their ability to address the "*cold start*" problem. This problem occurs when the user lacks specific requirements on what skills they want to learn, and the system lacks data on the new user.

LLM vs. LLM with RAG. We further examined the performance of a standard LLM recommender system (without RAG and without using a dataset as a knowledge base) versus an RAG-enhanced LLM recommender system by testing different prompt templates for the retriever and various user queries for the generator to ascertain improvements in system performance and recommendation personalization.

To explore the performance of our course recommender system, we focused on comparing the relevance of the recommended courses to different prompts by varying prompt templates and user-specific requirements.

4. Results

4.1. LLM vs. Non-LLM

We compared RAMO with a traditional course recommendation system built by the content-based and collaborative filtering using the same dataset[2]. During this comparison, we focused on the "*cold start*" problem. The "*cold start*" problem is especially pertinent in the context of an e-learning platform for tutor training, such as tutor training platform [38]. When new tutors join the platform, they are encouraged to complete various training courses to enhance their tutoring skills. Given the wide range of courses available, new tutors may feel overwhelmed when deciding where to begin their learning journey. In such scenarios, they may ask general questions such as, "*What can I learn today since I am a new tutor onboarding to this platform?*" They do not have prior course completions or specific learning preferences logged in the system, making it challenging for the recommendation system to personalize suggestions based on historical data. When prompted with "*I am a new user*", the traditional recommender system failed to generate a recommendation because its algorithm relies on the cosine similarity of the descriptive texts of the user's desired learning topic and the database items, and there are no courses with similar title or description as the phrase '*new user*'. In contrast, both our standard LLM and the RAG-enhanced LLM system can provide relevant course suggestions for the new user, with the LLM offering more detailed descriptions based on its internal knowledge base and RAG offering more customized outputs based on its external knowledge base and the prompt template we designed. The comparative results for both the standard and RAG-based recommender systems are displayed in Figure 3.

Regarding system performance, the traditional system typically took about 0.02 seconds longer than RAMO to generate responses according to the same user interest—a certain topic the user wants to learn, and this delay increased with the complexity of the user's input regarding relevant skills.

4.2. LLM vs. LLM with RAG

To explore how well our LLMs can provide personalized course recommendations, we used prompts that specified a particular skill to be learned. The non-RAG LLM (based on GPT-3.5) delivered detailed suggestions for relevant courses available on Coursera, utilizing its internal database of courses. In contrast, the recommendations from the RAG-enhanced LLM varied according to the specific prompt tem-

[2]https://www.kaggle.com/code/sagarbapodara/coursera-course-recommendation-system-webapp

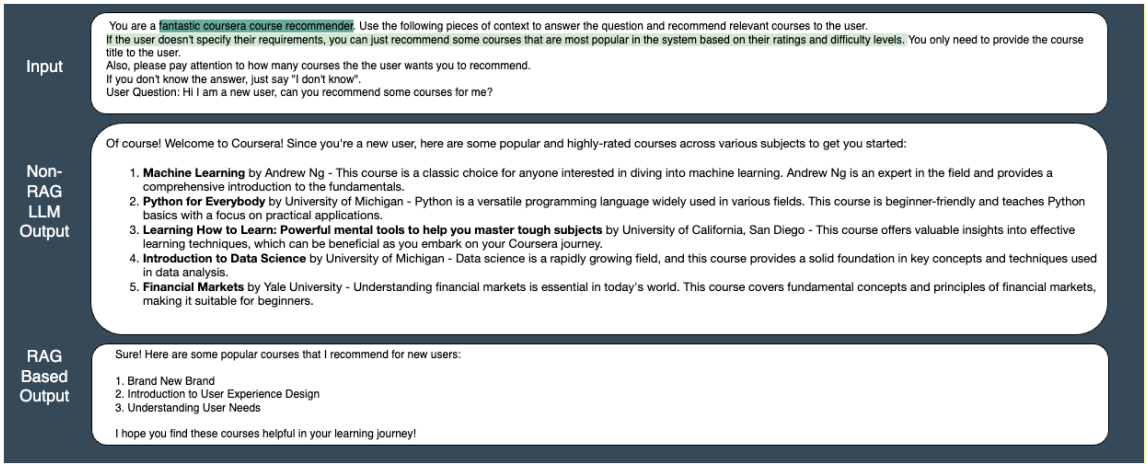

Figure 3: Sample output for a cold-start question on LLM vs RAG-LLM system

plate used by the retriever. This adaptability allows developers to tailor the quantity and detail of the courses recommended, showcasing the flexibility of the RAG approach. The user interface and the outcomes for a query focused on learning a specific skill are illustrated in Figure 4.

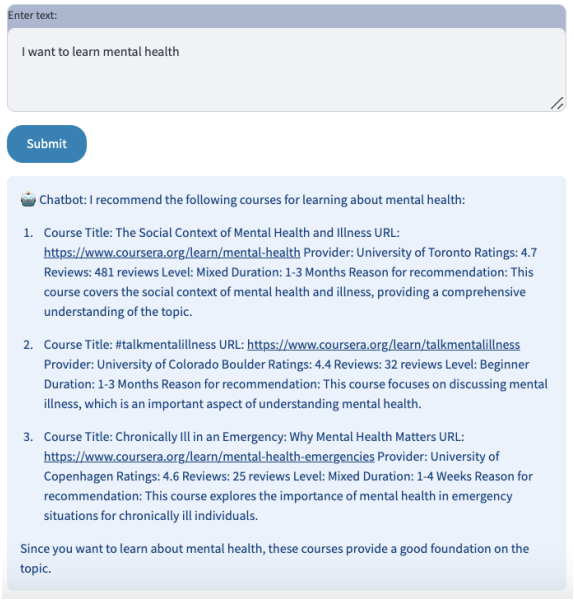

Figure 4: Output for a specific user question

We modified the retrieval prompts and generation queries to test the adaptability of our recommendation system. First, we conducted tests on various user queries using the same prompt template to compare the variations in output. The first module in Figure 5 illustrates the system's response to a "*cold start*" problem, while modules 2 through 6 demonstrate how the output varies based on user questions about the number of courses recommended and the level of detail provided, such as reasons for recommendations, URLs, and other specifics. For example, when user asks question like "*I want to learn python, can you recommend me some courses?*", RAMO can give the output to the user: "*Sure! Here are some recommended Python courses for you: 1. Introduction to Python 2. Crash Course on Python 3. First Python Program 4. Python Basics These courses cover a range of topics from basic syntax to building interactive applications. Happy learning!*" When the user changes their mind and decides to learn about another topic, RAMO can give relevant recommendations. The outputs consistently matched the user requirements in relevance, successfully retrieving the pertinent courses from the Coursera dataset, more examples could be found at Figure 5.

Figure 5: User questions and related outputs

Figure 6: Prompt templates and related outputs

We also utilized different retrieval prompt templates to explore how the output varies based on different prompts. Specifically, we used the same user question "*I want to learn python*", and altered the prompt templates to specify the number of recommended courses and the level of detail provided in the output, ranging from mere course titles to comprehensive descriptions that include titles, URLs, and rationales for each recommendation. The variations in the prompt templates and their corresponding outputs are illustrated in Figure 6. Here, red lines highlight changes in the number of courses recommended, blue lines detail the content of the courses—such as the inclusion of reasons for recommendations or just the course titles, ratings, and URLs—while green highlights how we addressed the "*cold-start*" problem, resulting in recommendations of the three most popular (based on course ratings) and easiest courses (based on its difficulty level), as depicted in the output module labeled 1 in Figure 6. The generated response in response to varied prompts underscores the system's robustness; for instance, when the template specifies "*recommend three courses at a time*", the output consistently includes exactly three courses. Similarly, if the prompt contains 'course URLs and titles', the system reliably appends this information to each recommended course, ensuring that the output meticulously adheres to the specified criteria.

5. Conclusions

In this study, we have demonstrated the application of LLMs as course recommender systems, particularly within MOOCs. Our findings confirm the potential of LLMs to deliver personalized course recommendations based on user's different requirements. We initially compared four LLMs, including GPT-3.5 Turbo and GPT-4. Ultimately, we selected GPT-3.5 as the back-end model for the RAMO system due to its comparable performance to GPT-4 at a lower cost. Although the Llama models are free to access, we found that the GPT models were significantly faster. Specifically, GPT-3.5 had an approximate response time of 3 seconds, whereas Llama 2 and Llama 3 took approximately 5 minutes and 8 minutes, respectively. Furthermore, the integration of RAG has enhanced the quality of recommendation outputs, as evidenced by the generated responses based on various user prompts, which are highly related to user's needs and all came from the knowledge base. Additionally, our system supports conversational interaction with users, which could be seamlessly integrated into numerous online educational platforms. Our use of open-source LLMs (e.g. Llama 2 and Llama 3 [33]) has also been validated, proving to be a cost-effective approach for broader deployment.

Limitations

As this study is ongoing, we have not yet conducted comprehensive evaluations of our recommender systems, including human evaluations or user studies. This is primarily due to the nascent stage of our research. Moreover, while many research projects on recommendation systems employ benchmarks to evaluate system adaptability, our study currently lacks such benchmarks because we do not possess a test dataset. The Coursera dataset we utilized includes only course data, lacking user profiles which are essential for evaluating the effectiveness of recommender systems across different time periods. If we had access to user data, including users' past course learning histories and their preferences, we could integrate this information with the course data to enhance our retrieval process. This integration would allow us to personalize recommendations more effectively, tailoring course suggestions to individual learning patterns and preferences. Incorporating detailed user data would enable RAMO to provide more accurate and relevant recommendations, improving user satisfaction and engagement. It would also allow for longitudinal studies to track how users' interactions with the system evolve over time and how well the recommendations align with their long-term learning goals.

Future Work

We plan to undertake several further steps to advance our research. *Firstly*, we aim to conduct thorough evaluations and tests to validate the efficacy and reliability of our recommender systems. This will involve integrating user studies and utilizing real user data once our systems are deployed on our e-learning platform. Such measures will enable us to robustly measure performance and refine our approach. *Secondly*, we will focus on enhancing system performance, considering scalability and the potential to expand our technology to encompass a broader range of educational tools and platforms. These efforts will ensure that our recommender systems not only meet current educational needs but also adapt to future demands and technological advancements. *Thirdly*, we could deploy RAMO on our own e-learning platform, and then have

the opportunity to gather comprehensive user data and utilize our own course dataset rather than Coursera's. This deployment would allow us to conduct extensive testing and validation, further proving the eligibility and effectiveness of the LLM for recommending courses. With access to real-time user data, we could continuously refine our algorithms, making the system more adaptive and responsive to users' evolving needs.

To evaluate the effectiveness of our LLM-based course recommendation system, we plan to conduct a comprehensive experiment that includes quantitative metrics, user studies, and personalization improvements. Our experiment aims to assess both the relevancy of the recommendations and the satisfaction of the users with the recommended courses.

We will utilize several quantitative metrics to evaluate the performance of the recommendation system. Key metrics include post-test performance, measured by the improvement in students' scores from pre-test to post-test after tutoring sessions, and course completion rate, which compares the rate of course completion between students who follow the system's recommendations and those who do not. Additionally, engagement rate will be tracked by monitoring whether students continue engaging with the lesson without dropping out midway. User satisfaction will also be assessed through feedback collected after each lesson via a thumbs-up or thumbs-down system and detailed surveys. To gather qualitative insights into the system's effectiveness and user experience, we will conduct user studies. These will involve satisfaction surveys completed by students following each lesson to gauge their satisfaction with the course content and the relevance of the recommendations, as well as focus group discussions to explore students' experiences in more depth and gather suggestions for improvement.

Acknowledgments

We extend our sincere gratitude to Chenfei Lou, a current software engineer at X (former twitter), for his invaluable guidance in developing our demo. We also thank Sandy Zhao, a current master's student in the CMU METALs program, for her excellent assistance in generating the wonderful diagram. Additionally, we appreciate Yuting Wang, an undergraduate student at CMU, for her help in refining the design in this paper.

References

[1] M. H. Baturay, An overview of the world of moocs, Procedia - Social and Behavioral Sciences 174 (2015) 427–433. doi:https://doi.org/10.1016/j.sbspro.2015.01.685, international Conference on New Horizons in Education, INTE 2014, 25-27 June 2014, Paris, France.

[2] N. M. Castillo, J. Lee, F. T. Zahra, D. A. Wagner, Moocs for development: Trends, challenges, and opportunities, Information Technologies & International Development 11 (2015) pp–35.

[3] J. Knox, Digital culture clash:"massive" education in the e-learning and digital cultures mooc, Distance Education 35 (2014) 164–177.

[4] C. Romero, S. Ventura, Educational data mining: a review of the state of the art, IEEE Transactions on Systems, Man, and Cybernetics, Part C (applications and reviews) 40 (2010) 601–618.

[5] Z. Gulzar, A. A. Leema, G. Deepak, Pcrs: Personalized course recommender system based on hybrid approach, Procedia Computer Science 125 (2018) 518–524.

[6] J. Jeevamol, V. Renumol, An ontology-based hybrid e-learning content recommender system for alleviating the cold-start problem, Education and Information Technologies 26 (2021) 4993–5022.

[7] Y. Lu, Y. Fang, C. Shi, Meta-learning on heterogeneous information networks for cold-start recommendation, in: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 2020, pp. 1563–1573.

[8] T. E. Kolb, A. Wagne, M. Sertkan, J. Neidhardt, Potentials of combining local knowledge and llms for recommender systems, in: # PLACEHOLDER_PARENT_METADATA_VALUE#, volume 3560, CEUR-WS. org, 2023, pp. 61–64.

[9] J. Li, W. Zhang, T. Wang, G. Xiong, A. Lu, G. Medioni, Gpt4rec: A generative framework for personalized recommendation and user interests interpretation, 2023. arXiv:2304.03879.

[10] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, J. Zhang, Chat-rec: Towards interactive and explainable llms-augmented recommender system, 2023. arXiv:2303.14524.

[11] H. Lyu, S. Jiang, H. Zeng, Y. Xia, Q. Wang, S. Zhang, R. Chen, C. Leung, J. Tang, J. Luo, Llm-rec: Personalized recommendation via prompting large language models, 2024. arXiv:2307.15780.

[12] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, Retrieval-augmented generation for large language models: A survey, arXiv preprint arXiv:2312.10997 (2023).

[13] Z. Feng, X. Feng, D. Zhao, M. Yang, B. Qin, Retrieval-generation synergy augmented large language models, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2024, pp. 11661–11665.

[14] J. Basilico, T. Hofmann, Unifying collaborative and content-based filtering, in: Proceedings of the twenty-first international conference on Machine learning, 2004, p. 9.

[15] J. B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in: The adaptive web: methods and strategies of web personalization, Springer, 2007, pp. 291–324.

[16] Y. Koren, S. Rendle, R. Bell, Advances in collaborative filtering, Recommender systems handbook (2021) 91–142.

[17] X. Wu, H. Zhou, Y. Shi, W. Yao, X. Huang, N. Liu, Could small language models serve as recommenders? towards data-centric cold-start recommendation, in: Proceedings of the ACM on Web Conference 2024, 2024, pp. 3566–3575.

[18] Hasan, How does chatgpt generate human-like text?, 2023. URL: https://dev.to/hasan048/how-does-chatgpt-generate-human-like-text-3ljj.

[19] ZDNet, What is chatgpt and why does it matter? here's everything you need to know, 2024. URL: https://www.zdnet.com/article/what-is-chatgpt-and-why-does-it-matter-heres-everything-you-need-to-k

[20] M. R. Kabir, F. Lin, An llm-powered adaptive practicing system, in: AIED 2023 workshop on Empowering Education with LLMs-the Next-Gen Interface and Content

Generation, AIED, 2023.
[21] S. Wang, T. Xu, H. Li, C. Zhang, J. Liang, J. Tang, P. S. Yu, Q. Wen, Large language models for education: A survey and outlook, arXiv preprint arXiv:2403.18105 (2024).
[22] S. E. Huber, K. Kiili, S. Nebel, R. M. Ryan, M. Sailer, M. Ninaus, Leveraging the potential of large language models in education through playful and game-based learning, Educational Psychology Review 36 (2024) 25.
[23] W. Dai, J. Lin, H. Jin, T. Li, Y.-S. Tsai, D. Gašević, G. Chen, Can large language models provide feedback to students? a case study on chatgpt, in: 2023 IEEE International Conference on Advanced Learning Technologies (ICALT), IEEE, 2023, pp. 323–325.
[24] W. Dai, Y.-S. Tsai, J. Lin, A. Aldino, F. Jin, T. Li, D. Gasevic, et al., Assessing the proficiency of large language models in automatic feedback generation: An evaluation study (????).
[25] L. Zhang, J. Lin, C. Borchers, M. Cao, X. Hu, 3dg: A framework for using generative ai for handling sparse learner performance data from intelligent tutoring systems, arXiv preprint arXiv:2402.01746 (2024).
[26] L. Zhang, J. Lin, C. Borchers, J. Sabatini, J. Hollander, M. Cao, X. Hu, Predicting learning performance with large language models: A study in adult literacy, arXiv preprint arXiv:2403.14668 (2024).
[27] J. Lin, Z. Han, D. R. Thomas, A. Gurung, S. Gupta, V. Aleven, K. R. Koedinger, How can i get it right? using gpt to rephrase incorrect trainee responses, arXiv preprint arXiv:2405.00970 (2024).
[28] Z. Levonian, C. Li, W. Zhu, A. Gade, O. Henkel, M.-E. Postle, W. Xing, Retrieval-augmented generation to improve math question-answering: Trade-offs between groundedness and human preference, arXiv preprint arXiv:2310.03184 (2023).
[29] S. Jacobs, S. Jaschke, Leveraging lecture content for improved feedback: Explorations with gpt-4 and retrieval augmented generation, arXiv preprint arXiv:2405.06681 (2024).
[30] J. Lin, A. Gurung, D. R. Thomas, E. Chen, C. Borchers, S. Gupta, K. R. Koedinger, et al., Improving assessment of tutoring practices using retrieval-augmented generation, arXiv preprint arXiv:2402.14594 (2024).
[31] R. Vinay, G. Spitale, N. Biller-Andorno, F. Germani, Emotional manipulation through prompt engineering amplifies disinformation generation in ai large language models, arXiv preprint arXiv:2403.03550 (2024).
[32] Maximum length - netdocuments, 2024. URL: https://support.netdocuments.com/s/article/Maximum-Length, accessed: 2024-07-05.
[33] Meta, Llama: Large language model meta ai, 2024. URL: https://llama.meta.com/.
[34] OpenAI, New and improved embedding model, 2024. URL: https://openai.com/index/new-and-improved-embedding-model/.
[35] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.
[36] B. Moradiya, The battle of language models: Openai vs. bert, 2023. URL: https://medium.com/@moradiyabhavik/the-battle-of-language-models-openai-vs-bert-ee46f4e5ef2f.
[37] O. Topsakal, T. C. Akinci, Creating large language model applications utilizing langchain: A primer on developing llm apps fast, in: International Conference on Applied Engineering and Natural Sciences, volume 1, 2023, pp. 1050–1056.
[38] J. Lin, D. R. Thomas, Z. Han, W. Tan, N. D. Nguyen, S. Gupta, E. Gatz, C. Tipper, K. R. Koedinger, Personalized learning squared (plus): Doubling math learning through ai-assisted tutoring (2023).