Large Language Models in Education

Rupasinghe T. T. V. N.
Dept. of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka.
e17297@eng.pdn.ac.lk

Manohora H. T.
Dept. of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka.
e17206@eng.pdn.ac.lk

Kalpana M. W. V.
Dept. of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka.
e17148@eng.pdn.ac.lk

Dr. Damayanthi Herath
Dept. of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka.
damayanthiherath@eng.pdn.ac.lk

Prof. Roshan G. Ragel
Dept. of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka.
roshanr@eng.pdn.ac.lk

Dr. Isuru Nawinne
Dept. of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka.
isurunawinne@eng.pdn.ac.lk

Dr. Shamane Siriwardhana
Dept. of Computer Engineering
University of Peradeniya
Peradeniya, Sri Lanka.
gshasiri@gmail.com

I. Introduction

In the last few years, artificial intelligence has advanced significantly, particularly in the fields of generative AI and large language models (LLMs). These cutting-edge models have proven to be exceptionally capable of reading, writing, and producing content that is human-like, opening up new horizons in creativity and invention. The rise of generative AI has sparked a lot of curiosity about the possible uses it could have in a variety of industries. These generative AI models provide fascinating prospects in the field of education [1], where technology integration is becoming more common. Intelligent tutoring systems have been sought after in education for a long time to improve and customize students' learning experiences. The effectiveness of generative AI paired with the current trends in educational technology use offer a potent synergy that has the potential to completely change the educational environment [2]. It offers great potential for involving students, encouraging creativity, and facilitating individualized learning journeys for them as generative AI models may produce tailored content, adaptive learning materials, and interactive simulations. The dawn of a new era in education is on the horizon, driven by the revolutionary potential of generative AI [3], as we observe the convergence of cutting-edge technology and educational methodology.

This article explores how LLM might be used in education as the basis for creating a smart tutor that is both affordable and successful. We aim to develop a cutting-edge teaching tool that can adapt to the demands of each individual student, deliver interactive and interesting learning experiences, and provide individualized assistance and support by harnessing the power of LLMs.

Meaningless prompts are input text or queries in the context of LLMs that lack precise and logical context. LLMs are created to produce responses that are meaningful and pertinent to the input given. However, if the prompt is unclear or inconsistent, the results could be devoid of information that is important or pertinent. We can use certain strategies to this issue to perform prompt reduction and prompt filtering. Additionally, LLMs are extremely resource-intensive models that need a lot of processing power and time to provide results. Because of this, running queries on LLMs can be expensive in terms of both time and computer resources, especially when done at scale. There are numerous ways to lower the cost and increase the effectiveness of the cost effective intelligent tutor. These techniques are covered here.

The main goal of this is to use LLMs to their full potential to develop an intelligent tutor who can successfully support students in their learning process. We intend to lower computing costs while retaining the quality and accuracy of the tutoring system by employing strategies like prompt caching [6] and prompt filtering. Keeping and utilizing previously created responses for particular prompts is known as prompt caching. The already computed response is fetched from cache and returned directly, as opposed to repeatedly executing the LLM inference process for the same prompt. On the other hand, prompt filtering entails picking or changing prompts in a selected manner to raise the standard and applicability of the results produced. The suggested method uses a cache handler to maximize cost-efficiency and improve user experience by delivering prompt-specific material.

Computer architecture has been selected as the subdomain inside the education domain. Our cost effective intelligent tutor will make use of the wealth of learning resources, including the PDFs, slides, and videos, to give students thorough and individualized instruction. However, if the same procedures

are followed in the deployment of the tutor to the subdomain, we can include any other subdomain. The methodology we employ also includes an idea like vector similarity. We aim to build a mapping system that connects user searches with pertinent course materials by exploiting the idea of vector similarity. By doing this, it gives the LLM greater memory and a unique retrieval-based system, allowing it to access and use more information for producing more thorough and accurate responses.

The results of this have the potential to significantly alter current educational paradigms. We want to develop a cost-effective intelligent tutoring system that maximizes accessibility, personalisation, and cost-effectiveness in education by utilizing the capabilities of LLMs. We foresee an intelligent tutor that provides students with a seamless learning experience by adopting prompt caching, prompt filtering, and mapping course materials.

II. Literature Review

A. Cost Reduction Strategies for Integrating Language Model APIs

Integrating large language models (LLMs) in education has garnered significant attention and has shown tremendous potential in revolutionizing traditional teaching and learning methods [4]. As LLMs have advanced in sophistication and capability, researchers have begun exploring their application in developing intelligent tutoring systems that offer personalized and cost-effective educational experiences [5]. In the current landscape, the cost of utilizing Language Model APIs (LM APIs) has witnessed a notable increase owing to the surge in demand, consequently facilitating enhanced efficiency and convenience in daily life. As a result, individuals and organizations are actively seeking means to optimize their expenditures by implementing diverse system enhancements. Particularly within the industrial sector, the integration of LM APIs has become instrumental in driving business success. Within this context, numerous tutoring systems have embraced LM APIs as a pivotal component of their operations, albeit at a substantial financial investment.

B. Cost Reduction

Cost reduction is crucial in developing intelligent tutors, as it ensures wider accessibility and scalability. Implementing prompt cache handlers and prompt filtering techniques has gained traction in this regard. Prompt caching involves storing precomputed LLM outputs to mitigate computational overhead during runtime, resulting in faster response times and reduced computational costs. The intelligent tutor can efficiently retrieve relevant information without repeated computations by strategically managing and updating the prompt cache The fundamental idea is using a local cache, such as a database, to keep track of results when submitting queries to a Language Model API (LLM API). This method involves a preliminary check to see if a similar query has already been used. If so, the response can be directly obtained from the cache. Only when a matching query cannot be found in the cache, the LLM API is called. Significant cost reductions can be achieved by using a completion cache, especially in situations where similar queries are regularly used. For instance, in the case of a search engine powered by an LLM API, the completion cache effectively enables resolving all the users' requests if they are simultaneously searching for the same or related keywords [6].

The size of the prompt has a major impact on the cost of using LLM APIs, as the cost of an LLM query increases linearly with prompt size. As a result, prompt adaptation of a method for lowering the prompt's size is a natural way to lessen the costs associated with using the LLM API. Prompt selection is used as an example of prompt adaptation, where a smaller subset of examples is kept instead of using a prompt that uses many examples to demonstrate task performance. This method not only yields a shorter prompt, but it also lowers expenses. However, deciding which set of examples to keep is the best for each request presents a fascinating difficulty [6].

Concatenating many queries is another efficient method for cost reduction. To process individual requests, a LLM API must be repeatedly prompted with the same request. By delivering the prompt to the LLM API only once and enabling it to handle many queries, the basic idea behind query concatenation is to avoid unnecessary prompt processing. This is accomplished by combining several searches into one and expressly telling the LLM API to perform several queries at once in the prompt. This technique helps maximize the use of LLM API resources while lowering the expenses associated with timely submissions [6].

C. Implementing resource recommendations using Vector Similarity and Sentence Transforms.

Natural language processing embeddings are a powerful technology that allows texts to be represented as vectors in a multidimensional space [4], [5]. These vector representations efficiently capture the complex semantic and syntactic relationships that exist between texts, making it easier to complete a variety of tasks like text classification, sentiment analysis, and search. Transformer architectures are frequently used in the acquisition of embeddings, two prominent examples are BERT and RoBERT [6]. These designs use large amounts of textual data to obtain precise vector representations of words.

Sentence transformers and vector similarity are essential elements of similarity search. Sentence embeddings can be produced to capture semantic meaning by using pre-trained models. The core of phrases is encoded using these embeddings, which act as multidimensional representations. The creation of an index follows which improves search efficiency. A target text is encoded during search and contrasted with saved embeddings using similarity metrics like cosine similarity [10]. This method makes it possible to efficiently retrieve words or documents that are semantically comparable, as well as finding uses for it in information retrieval, recommendation systems, and various natural language processing tasks.

D. Proposed Framework

Within the education domain, our research focuses on the subdomain of computer architecture. This subject is integral to understanding computer systems' fundamental principles and design. We utilize various learning materials, including PDFs, slides, and videos, to develop an intelligent tutor tailored to computer architecture. These materials serve as the foundation for the intelligent tutor's knowledge base, enabling it to provide comprehensive and contextually appropriate guidance to students.

Mapping user queries to relevant course material is a crucial component of our intelligent tutor. To accomplish this, we employ the concept of vector similarity, which allows us to establish semantic connections between user queries and the content of the computer architecture course materials. By measuring the similarity between the vector representations of user queries and the available course materials, the intelligent tutor can dynamically generate URLs that lead to web pages specifically aligned with the user's information needs. This mapping process ensures that students have access to supplementary resources for further reading and exploration, enhancing their understanding and mastery of computer architecture concepts. Fig. 1 shows the architectural diagram of the proposed framework for addressing the problem.

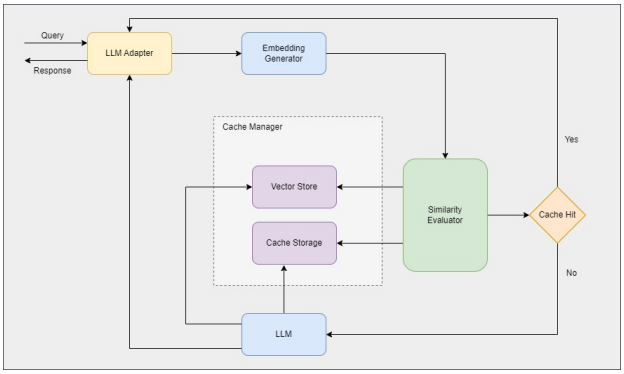

Fig. 1. Architectural Diagram of Cost Effective Intelligent Tutor

In summary, the literature supports using large language models in education, particularly in developing intelligent tutoring systems. Researchers have made significant strides in creating efficient and cost-effective tutoring systems by implementing cost-reduction techniques such as prompt cache handling and prompt filtering. Our research contributes to this field by focusing on the domain of education and the subdomain of computer architecture, utilizing available course materials, and employing vector similarity to establish a mapping system for prompt-based exploration. Integrating these approaches in an intelligent tutor can transform traditional educational practices, providing students with personalized and contextually relevant learning experiences.

REFERENCES

[1] D. Baidoo-Anu and L. Owusu Ansah, "Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning," SSRN 4337484, 2023.

[2] C.H. Chang and G. Kidman, "The rise of generative artificial intelligence (AI) language models-challenges and opportunities for geographical and environmental education," International Research in Geographical and Environmental Education, vol. 32, no. 2, pp. 85-89, 2023.

[3] H. Yu and Y. Guo, "Generative artificial intelligence empowers educational reform: current status, issues, and prospects," in Frontiers in Education, vol. 8, p. 1183162, June 2023.

[4] C. Cao, "Scaffolding CS1 Courses with a Large Language Model-Powered Intelligent Tutoring System," in Proceedings of the International Conference on Intelligent User Interfaces (IUI), Mar. 2023, pp. 229-232, doi: 10.1145/3581754.3584111.

[5] Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E. and Krusche, S., 2023. ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 103, p.102274.

[6] L. Chen, M. Zaharia, and J. Zou, "FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance," May 2023, [Online]. Available: http://arxiv.org/abs/2305.05176

[7] O. Levy and Y. Goldberg, "Dependency-Based Word Embeddings.

[8] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger, "From Word Embeddings To Document Distances.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Oct. 2018, [Online]. Available: http://arxiv.org/abs/1810.04805.

[10] C. Aguerrebere, I. Bhati, M. Hildebrand, M. Tepper, and T. Willke, "Similarity search in the blink of an eye with compressed indices," Apr. 2023, [Online]. Available: http://arxiv.org/abs/2304.04759.