TutorLLM: Customizing Learning Recommendations with Knowledge Tracing and Retrieval-Augmented Generation

Zhaoxing LI
zhaoxing.li@soton.ac.uk
University of Southampton
Southampton, UK

Vahid Yazdanpanah
University of Southampton
Southampton, UK
v.yazdanpanah@soton.ac.uk

Jindi Wang
Durham University
Durham, UK
jindi.wang@durham.ac.uk

Wen Gu
Japan Advanced Institute of Science and Technology
Nomi, Japan
wgu@jaist.ac.jp

Lei Shi
Newcastle University
Newcastle upon Tyne, UK
lei.shi@newcastle.ac.uk

Alexandra I. Cristea
Durham University
Durham, UK
alexandra.i.cristea@durham.ac.uk

Sarah Kiden
University of Southampton
Southampton, UK
sk3r24@soton.ac.uk

Sebastian Stein
University of Southampton
Southampton, UK
ss2@ecs.soton.ac.uk

ABSTRACT

The integration of AI in education offers significant potential to enhance learning efficiency. Large Language Models (LLMs), such as ChatGPT, Gemini, and Llama, allow students to query a wide range of topics, providing unprecedented flexibility. However, LLMs face challenges, such as handling varying content relevance and lack of personalization. To address these challenges, we propose TutorLLM, a personalized learning recommender LLM system based on Knowledge Tracing (KT) and Retrieval-Augmented Generation (RAG). The novelty of TutorLLM lies in its unique combination of KT and RAG techniques with LLMs, which enables dynamic retrieval of context-specific knowledge and provides personalized learning recommendations based on the student's personal learning state. Specifically, this integration allows TutorLLM to tailor responses based on individual learning states predicted by the Multi-Features with Latent Relations BERT-based KT (MLFBK) model and to enhance response accuracy with a Scraper model. The evaluation includes user assessment questionnaires and performance metrics, demonstrating a 10% improvement in user satisfaction and a 5% increase in quiz scores compared to using general LLMs alone.

CCS CONCEPTS

• **Applied computing** → **E-learning**; • **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

Learning Recommender System, Large Language Models, Personalized Learning, Knowledge Tracing

ACM Reference Format:
Zhaoxing LI, Vahid Yazdanpanah, Jindi Wang, Wen Gu, Lei Shi, Alexandra I. Cristea, Sarah Kiden, and Sebastian Stein. 2024. TutorLLM: Customizing Learning Recommendations with Knowledge Tracing and Retrieval-Augmented Generation. In *Proceedings of 18th ACM Conference on Recommender Systems (RecSys 2024).* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

AI techniques are increasingly affecting various aspects of daily life, notably in educational environments. AI offers significant opportunities to enhance both the learning process and efficiency for students. Prominent among these AI applications are Large Language Models (LLMs), such as ChatGPT[1], Gemini [2], and Llama [3], which allow students to query a wide array of topics, thus offering unprecedented flexibility in learning. Unlike traditional search engines, LLMs enable students to ask nuanced and complex questions, engage in conversational interactions, and seek clarifications through follow-up questions, enhancing the depth and effectiveness of the learning experience. Despite these advantages, current LLMs face several challenges. These include generating inaccurate information (commonly referred to as "hallucinations"), lack of personalization, and varying content relevance [1, 10, 25]. Specifically, these models often struggle with problems requiring high-level logical and mathematical problems, such as solving complex equations or providing step-by-step logical reasoning, and fail to tailor responses to individual learning levels, sometimes providing overly generalized answers or requiring extensive prompting to yield useful information [25].

RecSys 2024, October 14–18, 2024, Bari, Italy

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

[1] https://chatgpt.com/
[2] https://gemini.google.com/app
[3] https://llama.meta.com/

In contrast, traditional educational recommender systems utilize technologies such as Knowledge Tracing (KT) to track the learning trajectory of students and offer personalized recommendations. KT methods leverage historical interaction data to predict future learning actions [9]. Despite their utility, these systems often fall short in terms of linguistic versatility and adaptability when compared to LLMs. Their responses are typically confined to a pre-defined set within their databases, limiting their ability to respond dynamically to a wide range of queries [25]. This limitation restricts the system's ability to adapt to individual learning needs in real-time, thereby reducing the effectiveness and engagement of the educational experience.

To bridge the gap between the adaptability of LLMs and the personalized approach of educational recommender systems, we propose a novel framework: ***the Personalized Educational Recommender LLM System (TutorLLM), based on Knowledge Tracing (KT) and Retrieval-Augmented Generation (RAG)***. To the best of our knowledge, we are the first to incorporate KT technology into an LLM to achieve a personalized recommendation learning framework. The TutorLLM comprises three integral components. The first component is a KT model, which is developed based on the Multi-Features with Latent Relations BERT Knowledge Tracing (MLFBK) model [15]. This component not only gathers data on student interactions and performance but also assimilates information from dialogues with LLMs, offering insights into student capabilities, learning states, and the complexities of the knowledge being acquired. The second is a Scraper. This component collects text content during online learning sessions to provide context-specific background knowledge that enhances the relevance and accuracy of the LLM's responses. The third component is an RAG Enhanced LLM. This component, utilizing the GPT-4 API, integrates inputs from both the Scraper and the KT module to deliver precise, personalized responses and learning content recommendations. Additionally, for even more tailored interaction, students can manually upload learning materials. Figure 1 shows the overall model of the architecture. The text content provided by Scraper mitigates hallucinations by dynamically retrieving and incorporating context-specific knowledge from relevant course materials, ensuring accurate and reliable information. Additionally, the integration of knowledge tracing allows TutorLLM to deliver highly personalized recommendations and responses, tailored to each student's learning progress and needs.

To implement and evaluate this integrated approach, we developed a Chrome browser plugin that serves as an interface for students to interact directly with TutorLLM. Students could engage with TutorLLM by asking questions during online learning sessions. After the course, TutorLLM will provide students with personalized study material recommendations. Students can then decide whether to pursue further study based on these recommendations. Our evaluation, which involved a two-week field study with 30 undergraduate students in an online linear algebra course, used a crossover design to compare the effectiveness and user satisfaction between the general LLM approach and TutorLLM. Results showed a 10% increase in user satisfaction for TutorLLM users compared to the general LLM approach, as measured by the System Usability Scale (SUS), and a 5% improvement in academic performance, based on quiz scores.

2 RELATED WORK

Large Language Models (LLMs), which have advanced natural language processing (NLP) and understanding (NLU), have significantly impacted various fields, including education [12]. These models, trained on vast text data, can generate human-like text, comprehend complex queries, and provide detailed explanations [5]. Examples like OpenAI's GPT-3 and GPT-4 are used in educational tools for tutoring, answering questions, and generating study materials [4, 8]. However, LLMs can produce incorrect or misleading information ("hallucinations"), often lack personalized responses tailored to individual users' learning levels, and sometimes require significant prompting to be useful [23, 25].

Educational Recommender Systems provide personalized learning experiences by tailoring educational content based on students' needs, preferences, and progress [20]. Traditional systems use collaborative filtering, content-based filtering, or hybrid approaches to suggest relevant resources. Combining these systems with AI technologies like LLMs and KT models can significantly enhance personalization and performance in learning outcomes [6, 26].

Knowledge Tracing monitors and predicts students' knowledge states over time by tracking interactions with learning materials and assessments [9, 19]. It enables personalized recommendations and targeted interventions to address the knowledge gaps of the students. Traditional methods like Bayesian Knowledge Tracing (BKT) [18] and Deep Knowledge Tracing (DKT) [19] have been widely used in Educational Recommender Systems, leveraging historical data to predict future performance and learning needs [16, 17]. Recent models, such as Multi-Features with Latent Relations BERT Knowledge Tracing (MLFBK), enhance the accuracy and depth of predictions [7, 13, 15, 22].

3 IMPLEMENTATION OF TUTORLLM

TutorLLM consists of three main components: the Scraper Model for collecting educational content, the KT for predicting students' learning states, and the RAG based LLM for dynamically retrieving information and tailoring personalized responses.

Overall Methodology. The motivation of our approach is to enable LLMs to comprehend the student's learning state (or knowledge master state) and the specific content of the ongoing course, thereby furnishing contextualized responses and tailored learning content recommendations. Thus, our method is structured into three distinct components. Firstly, the KT component utilizes an algorithm to trace students' learning state. This encompasses tracing skill mastery, ability profiles, problem difficulty, and predicting the next most probable action for the student. We employ MLFBK as our KT method within this component. Secondly, the Scraper function is designed to gather and organize the text information from the online course platform, including captions and subtitles of videos embedded in the web pages, into a background knowledge base. After receiving student action sequence data from the KT component, the TutorLLM first synthesizes the student's current learning status, focusing on identifying weak knowledge areas and predicting the student's potential next actions. Finally, the model gives personalized answers and recommendations based on the background knowledge base and students' learning state. Additionally, the model recommends additional learning materials to

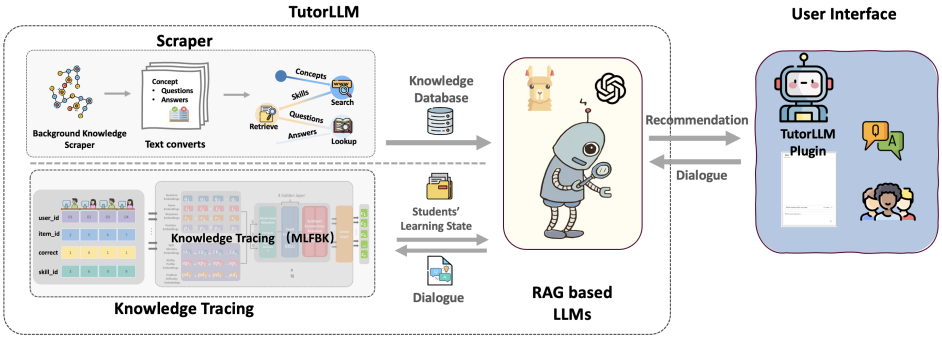

Figure 1: Overall architecture of TutorLLM.

students upon request or after each study session. Based on the above ideas, we built a Chrome browser plug-in. When students open an online course website, they can open our TutorLLM by clicking the button on the right-hand side of the address bar. Figure 2 shows the interface of the TutorLLM. Detailed insights into the functionalities of these three components are provided below:

Scraper Model: The Scraper component of TutorLLM was developed to autonomously collect and organize textual content from online course web pages. It extracts text information, including crucial captions and subtitles, to build a rich background knowledge base. This functionality is built on the Reader API from Jina AI[4]. In operation, the Scraper dynamically interacts with educational websites as a Chrome plugin activated by students. Upon visiting a relevant page, the Scraper processes the content, converting Uniform Resource Locators (URLs) to LLM-friendly inputs that include structured text and contextual captions. This ensures that every piece of extracted content is optimized for use by the large language model, providing accurate, up-to-date information that reflects the current scope of the course materials.

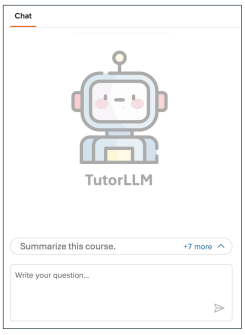

Figure 2: User Interface of the TutorLLM.

Knowledge Tracing Model: For the knowledge tracing part, we use MLFBK[15] to incorporate multi-feature embedding and latent Relations to capture students' learning state. MLFBK consists of three parts: embedding, BERT-based architecture, and correctness sequence output.

[4]https://github.com/jina-ai/reader

Within the embedding part, there are two components: Multi-Features embedding and Latent-Relations embedding. In the multi-feature embedding component, four distinct features, including student_id, skill_id, question_id and response_id are integrated. The latent relations embedding component employs a feature engineering method to extract three significant relations: skill mastery, ability profile, and problem difficulty. Skill mastery is modeled based on Bayesian Knowledge Tracing, while the ability profile is encoded using past performance in various time intervals and updated via the K-means algorithm [21]. Problem difficulty is quantified on a scale of 1 to 10 (where 1 is easy and 10 is extremely difficult), derived from success rates. Additionally, the overall model integration encompasses a combination of embeddings such as question, item, response, skill mastery, ability profile, and problem difficulty embeddings, culminating in creating a final input embedding.

In the BERT-based architecture, encoder blocks leverage a pre-LN Transformer architecture [23], incorporating monotonic convolutional multi-head attention followed by fully connected layers with LeakyReLU activation. Monotonic multi-head attention, in conjunction with mixed attention and monotonic attention, is utilized for sequence data representation.

In the correctness sequence output part, we initially acquire the prediction regarding the student's sequence action data. Subsequently, the prediction action will append to the end of the student's historical action sequence, generating the complete output sequence.

RAG Large Language Model: The large language model component, powered by the GPT-4 API, utilizes a Retrieval-Augmented Generation (RAG) [14] approach to produce responses.

Upon activation, the model receives a processed input from the knowledge tracing component, which includes detailed insights about the student's current mastery of various skills and predicted next actions. The retrieval process begins with the model querying the background knowledge base, which has been populated by the Scraper model with data from online course materials. This ensures that the responses are not only based on generic information but are enriched with up-to-date, course-specific content that enhances learning effectiveness.

The LLM then synthesizes the information from the knowledge base with the student's specific learning context to generate responses. These responses could range from direct answers to

queries, explanations of complex topics, or hints to guide problem-solving. Furthermore, the model actively offers study recommendations based on the student's learning progress and areas of difficulty, which could include suggestions for revisiting certain topics or advancing to new content based on the student's readiness.

4 USER STUDY

Participants & Study Setup. To evaluate the effectiveness of the TutorLLM educational tool on student learning outcomes, 30 first-year undergraduate students from XXX University (anonymous for review) enrolled in a linear algebra module were randomly allocated to one of three groups in a controlled experiment. The control group exclusively used general LLMs (the general web version of chatGPT4) for the duration of the study. Experimental Group 1 was exposed to general LLMs during the first week, followed by TutorLLM in the second week, whereas Experimental Group 2 utilized TutorLLM throughout both weeks. The key objective was to determine if integrating TutorLLM could enhance learning performance compared to the general LLMs.

Student Performance. Over a two-week period, students explore a new segment of linear algebra each day. They utilize our TutorLLM to address their queries daily during their studies. At the course's conclusion, TutorLLM suggests pertinent educational resources and exercises for further learning. Students have the autonomy to decide whether to engage with these additional materials, unlike those in the control group who use a general LLM model and do not receive such recommendations. Each day culminates with a quiz testing the knowledge acquired, leading up to a comprehensive exam covering all topics at the end of the two weeks. The assessment framework includes 15 tests overall, with each daily test comprising 10 questions and the potential to score up to 100 points. These daily quizzes and the final comprehensive test together determine the students' overall performance.

User Study. A questionnaire was administered to students in Group 1 and Group 2 to evaluate the effectiveness of the TutorLLM and its impact on student satisfaction. The survey was designed to gather quantitative and qualitative data on students' experiences with the TutorLLM. It incorporated a System Usability Scale (SUS) [2] and a User Experience (UX) [24] questionnaire to provide a comprehensive assessment. SUS, a validated tool, measures user perceptions of usability, focusing on user-friendliness and comprehensibility, with higher SUS scores indicating a more intuitive interface and enhanced user control. UX was measured from three perspectives, which are User Satisfaction (US), Comfort Level (CL) and Continue Willingness (CW). US was evaluated using a 5-point Likert scale to assess the overall satisfaction with the tool, where a positive US score reflects the assistant's ability to meet or exceed user expectations regarding usefulness and responsiveness. CL was rated on a 5-point scale to gauge user comfort during initial textual interactions on online platforms, providing critical insights for optimizing chat interfaces and enhancing interaction experiences [3]. CW, measured on a 5-point scale, captures users' willingness to continue interactions after the initial conversation [11].

5 RESULTS

Test Results. We evaluated the student performance of students across three control groups through 15 tests, including a final examination. The groups comprised students utilizing general LLMs, a hybrid LLM approach, and TutorLLM throughout the study. Our initial analysis revealed that the TutorLLM group achieved the highest overall mean score of 74.48, followed by the hybrid LLM group at 72.81 and the general LLM group at 71.97. To determine the statistical significance of the differences observed in the final exam scores among the different groups, we conducted an analysis of variance (ANOVA). This analysis produced an F-statistic of 0.795 and a p-value of 0.462, indicating no significant variance in performance across the groups. Figure 3 displays the daily mean scores for each study group. Although the differences were not statistically significant, there was a discernible trend toward improved performance in the TutorLLM group. These results suggest that additional research, potentially involving a larger sample size, different study designs, or a longer test duration, is necessary to definitively ascertain the effectiveness of integrating TutorLLM in enhancing academic performance.

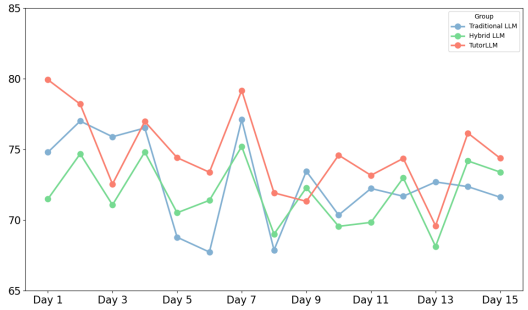

Figure 3: Daily Mean Scores of User Performance Across Different Study Groups

Usability and User Experience. From the usability perspective, the SUS results indicated a generally positive user experience, with an average SUS score of 76.35 and a median score of 79, reflecting high usability. The scores showed moderate variability, with a standard deviation of 14.46, and the majority of scores ranged between 61.89 and 90.81, demonstrating consistency in positive evaluations. The Shapiro-Wilk test confirmed the normality of the score distribution ($p = 0.721$), supporting the validity of the usability assessment. A bootstrap analysis estimated the median's 95% confidence interval to be between 68.5 and 83, further validating the positive user feedback on the TutorLLM's usability.

From the user experience perspective, the analysis of CL, CW, and US showed average scores of 3.50 for CL, 3.40 for CW, and 3.61 for US, with respective standard deviations of 1.00, 0.99, and 0.82. Correlation analysis revealed moderate positive correlations among the metrics CL and CW at 0.53, CL and US at 0.45, and CW and US at 0.40, suggesting that users felt more comfortable and satisfied with the app and were more willing to continue using it.

6 DISCUSSION

While our study did not show statistically significant improvements in academic performance with TutorLLM, the increased engagement and higher average scores suggest its potential benefits. We monitored the duration of LLM usage among students. Users of TutorLLM spent 36% more time engaging with the system compared to those using general LLMs. This increased engagement suggests that students were more satisfied with TutorLLM, making them more willing to invest additional time in the system. Additionally, the learning materials recommended by our TutorLLM appear to have a positive impact on students' learning performance to a certain extent. These findings underscore the importance of further refining AI-driven educational tools to better adapt to individual learning needs and contexts. It also could redefine personalized learning, making educational interactions more effective and engaging through tailored content and intelligent response systems.

7 CONCLUSION

In this paper, we introduced TutorLLM, a novel framework that integrates Large LLMs with KT and RAG to enhance personalized learning. Our main contributions include being the first to combine LLMs with KT to improve personalization. We demonstrated the practical application of TutorLLM through a Chrome browser plugin and validated its effectiveness in a two-week study with undergraduate students. Future research should focus on refining TutorLLM's personalization features, testing its effectiveness across various disciplines, and addressing challenges such as integrating existing technologies, ensuring data privacy, and providing educator training.

REFERENCES

[1] Razvan Azamfirei, Sapna R Kudchadkar, and James Fackler. 2023. Large language models and the perils of their hallucinations. *Critical Care* 27, 1 (2023), 120.

[2] Aaron Bangor, Philip T Kortum, and James T Miller. 2008. An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction* 24, 6 (2008), 574–594.

[3] Susan Bergin and Ronan Reilly. 2005. The influence of motivation and comfort-level on learning to program. (2005).

[4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[5] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology* 15, 3 (2024), 1–45.

[6] Hsinchun Chen, Roger HL Chiang, and Veda C Storey. 2020. Big data analytics for educational recommender systems: A review and research agenda. *MIS quarterly* 44, 1 (2020), 135–169.

[7] Jiahao Chen, Zitao Liu, Shuyan Huang, Qiongqiong Liu, and Weiqi Luo. 2023. Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 14196–14204.

[8] Tsung-Hsien Chen, Minghao Li, Ying-Hsun Chien, et al. 2021. A survey of large language models. *arXiv preprint arXiv:2109.11601* (2021).

[9] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.

[10] Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines* 30, 4 (2020), 681–694.

[11] Mark Freiermuth and Douglas Jarrell. 2006. Willingness to communicate: can online chat help? 1. *International journal of applied linguistics* 16, 2 (2006), 189–212.

[12] Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. 2023. Performance of ChatGPT on USMLE: potential for AI-assisted medical education using large language models. *PLoS digital health* 2, 2 (2023), e0000198.

[13] Unggi Lee, Yonghyun Park, Yujin Kim, Seongyune Choi, and Hyeoncheol Kim. 2022. MonaCoBERT: Monotonic attention based ConvBERT for Knowledge Tracing. *arXiv preprint arXiv:2208.12615* (2022).

[14] 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.

[15] Zhaoxing Li, Mark Jacobsen, Lei Shi, Yunzhan Zhou, and Jindi Wang. 2023. Broader and Deeper: A Multi-Features with Latent Relations BERT Knowledge Tracing Model. In *European Conference on Technology Enhanced Learning*. Springer, 183–197.

[16] Qi Liu, Shuanghong Shen, Zhenya Huang, Enhong Chen, and Yonghe Zheng. 2021. A survey of knowledge tracing. *arXiv preprint arXiv:2105.15106* (2021).

[17] Himanshu Pandey and Rajesh Kumar Pandey. 2019. Machine learning based approaches for personalized recommender system: A survey. *Procedia computer science* 165 (2019), 551–559.

[18] Radek Pelánek. 2017. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User modeling and user-adapted interaction* 27 (2017), 313–350.

[19] Chris Piech, Joe Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Advances in neural information processing systems* 28 (2015).

[20] Francesco Ricci, Lior Rokach, Bracha Shapira, et al. 2011. Introduction to recommender systems handbook. *Recommender systems handbook* (2011), 1–35.

[21] Douglas Steinley. 2006. K-means clustering: a half-century synthesis. *Brit. J. Math. Statist. Psych.* 59, 1 (2006), 1–34.

[22] Zejie Tiana, Guangcong Zhengc, Brendan Flanaganb, Jiazhi Mic, and Hiroaki Ogatab. 2021. BEKT: Deep Knowledge Tracing with Bidirectional Encoder Representations from Transformers. In *Proceedings of the 29th International Conference on Computers in Education*.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[24] Arnold POS Vermeeren, Effie Lai-Chong Law, Virpi Roto, Marianna Obrist, Jettie Hoonhout, and Kaisa Väänänen-Vainio-Mattila. 2010. User experience evaluation methods: current state and development needs. In *Proceedings of the 6th Nordic conference on human-computer interaction: Extending boundaries*. 521–530.

[25] Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. 2024. Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology* 55, 1 (2024), 90–112.

[26] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* 52, 1 (2019), 1–38.