A Comprehensive Survey and Taxonomy on Large Language Model-Based Knowledge Tracing

Sunwoo Park and Hyeoncheol Kim(✉)

Korea University, Seoul, Republic of Korea
{sunwoosan,harrykim}@korea.ac.kr

Abstract. Large language models (LLMs) have significant potential for intelligent tutoring systems (ITS), particularly in knowledge tracing (KT). Many current studies exhibit diverse approaches to LLM-based KT. However, despite the growing body of research, there is a lack of a consistent taxonomy for integrating LLMs into KT. In response, this study proposes a systematic taxonomy that categorizes the various roles LLMs can play in KT into three categories: LLM-enhanced, LLM-integrated, and LLM-standalone. Using this taxonomy, we systematically review and analyze studies published over the past three years that incorporate LLMs into knowledge tracing. Our analysis reveals that the role of LLMs, their strengths and weaknesses, and the type of data used, metrics vary across these categories. We also discuss the major challenges faced by each taxonomy, including optimizing feature fusion, handling real-time and unstructured inputs, designing effective prompts, and ensuring explainability. This comprehensive review provides a conceptual foundation and directions for future research in ITS driven by generative AI.

Keywords: knowledge tracing · Large Language Models · Intelligent tutoring system · Taxonomy · Personalized Education

1 Introduction

1.1 Knowledge Tracing and Large Language Models

Knowledge tracing (KT) is a key technique in intelligent tutoring systems (ITS) that continuously tracks and predicts learners' knowledge states [21,36]. Although early KT research often simplified knowledge into binary states, more recent methods capture nuanced representations of each concept and leverage advanced AI for enhanced predictive accuracy [23,30,31,34,37,44].

In particular, the recent development of LLMs, which shows excellent performance in the understanding and generation of natural languages, provides new opportunities for KT. This is because it supports integration of narrative answers, activity logs, and conversation data beyond the existing binary

S. Graf and A. Markos (Eds.): ITS 2025, LNCS 15723, pp. 246–258, 2026.
https://doi.org/10.1007/978-3-031-98281-1_20

answer input method, and can reflect complex learning states through fine-tuning [2,7,8,10,12,25,48]. Many researchers have hypothesized and experimentally verified that LLMs can improve KT models by interpreting students' open-ended answers, predicting question difficulty, and identifying misunderstandings in answers, and published related papers [11,13–15,17,27,35,45].

1.2 Motivation and Contributions for a Taxonomy of LLM-Based KT

Motivation. The convergence of LLMs and KT is a critical turning point in the evolution of ITSs. In recent studies, various KT methodologies using LLMs have been attempted, which can be broadly divided into several categories. Some studies focus on the use of LLMs as a feature extractor to improve the performance of existing KT models [19,22,28,42]. Other studies integrate existing KT models into LLMs to directly manage and update learners' knowledge states [16,20,24,43]. Another study does not use traditional KT, but uses LLMs as a standalone KT model that directly predicts student status [15,17,27].

This variety of roles for LLMs has created ambiguity in defining their function within systematic KT research. Because there is no clear consensus for the researcher, it becomes difficult to compare methods or accumulate findings between studies. Now, researchers usually use interchangeable terms like "LLM-based KT," "GPT-based tutoring," or "knowledge tracking with LLMs" to describe similar concepts.

In other domains, researchers have introduced taxonomies to bring order to such ambiguity [1,4,9,33,40,41]. For example, in reinforcement learning, a recent review proposes a taxonomy of LLM-based RL integrations to categorize how the two techniques interact [32]. Similarly, in software engineering, researchers have outlined a taxonomy for LLM-integrated applications to establish common terminology and design dimensions [47]. Given the ongoing development of KT research, there is a pressing need for a clear and standardized taxonomy to guide the integration of LLMs. Without such a classification, the diverse streams and categories of research become difficult to delineate, leading to fragmentation and a lack of clarity regarding the contribution of each study.

To address this issue, we propose a comprehensive taxonomy that systematically categorizes LLM-based KT approaches, specifying the ways in which LLMs can be incorporated into existing KT models and the specific roles they can fulfill. By offering precise definitions and well-defined categories, our taxonomy facilitates meaningful comparisons across different methods, standardizes the terminology used in the field, and promotes more effective communication and collaboration among researchers.

Objectives and Contributions. The primary goal of this study is to present a taxonomy that positions diverse LLM-based KT approaches within three integration levels: LLM-Enhanced, LLM-Integrated, and LLM-Standalone. We also compare datasets, domain contexts, and performance metrics across LLM-based KT models to establish a coherent basis for assessing disparate studies.

Through the introduction of standardized terminology and a unified taxonomy, we address existing ambiguities in naming conventions, thereby promoting consistency, reproducibility, and methodological clarity. Ultimately, this taxonomy is designed to deepen theoretical understanding, streamline model development, and enhance the effective integration of LLMs into KT research and practice.

2 Survey Methodology

2.1 Literature Search and Scope

For this study, we conducted a systematic search in IEEE Xplore, ACM Digital Library, Scopus, and Google Scholar using keywords, such as "LLMs," "GPT," "LLaMA," with "KT" or "LLMs and student modeling," "student prediction," and so on. The search period was set from 2022 to 2024. The primary search resulted in 39 papers, and after excluding duplicates and short workshop abstracts, we further filtered out papers that did not explicitly integrate LLMs into KT work or lacked empirical evaluation of KT performance.

Additionally, we did not define LLMs as studies that used simple language models such as BERT, Transformer, or Attention structures. In this study, LLMs refers to a large, resource-consuming language model that can process prompts based on fine-tuning. Therefore, we excluded KT-related studies that utilized simple language model structures. This reflects our intention to identify recent trends by analyzing the surge of related research since the emergence of ChatGPT in November 2022. Ultimately, 25 studies met the above criteria and were included. The 25 selected studies have a small literature size, but this should be taken into account the fact that LLM-based KT is a recently formed research field and practical factors such as data accessibility and implementation complexity.

2.2 Classification Scheme

To systematically analyze the key dimensions of LLM-based KT, we propose a taxonomy that centers on the extent and manner in which LLMs are incorporated into traditional KT pipelines. Building on this classification, we further consider how various studies utilize different data modalities and evaluate model performance via commonly used metrics such as AUC (Area Under the Curve), ACC (Accuracy) and F1-Score. By comparing these elements holistically, our taxonomy highlights distinct methodological strengths, weaknesses, and domain-specific characteristics, providing a coherent basis for both quantitative assessment and the identification of emerging trends in LLM-based KT research.

3 Results

3.1 A New Taxonomy: LLM-Enhanced KT, LLM-Integrated KT, LLM-Standalone KT

As shown in Fig. 1, building on the diverse body of work reviewed for LLM-Enhanced, LLM-Integrated, and LLM-Standalone KT, this section provides a

detailed analysis that emphasizes how each approach is implemented, why certain studies are categorized in each manner, and where the boundaries of these categories may become blurred. We further discuss cases where existing studies do not perfectly fit into a single category, thus underscoring the need for flexibility in applying this taxonomy.

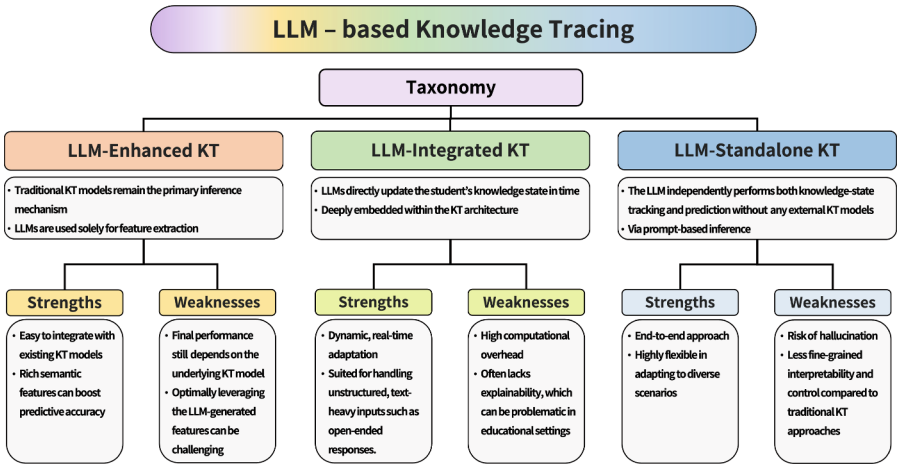

Fig. 1. Three Taxonomies for LLM-Based Knowledge Tracing.

LLM-Enhanced KT: Leveraging LLMs as Feature Generators. As depicted in Fig. 2, the LLM-Enhanced approach retains traditional KT models as the core inference engine while incorporating LLMs to augment feature extraction. In this taxonomy, LLMs primarily serve as feature generators, transforming unstructured data, such as open-ended student responses or problem text, into structured representations like semantic embeddings or difficulty estimates. These enriched features are subsequently fed into the existing KT pipeline, enhancing predictive accuracy without altering the core structure of traditional KT models.

The distinguishing feature of LLM-Enhanced KT, in contrast to the other categories, is that LLMs are used only in the initial phase of feature extraction. The generated text embeddings or semantic vectors are then processed by the existing KT models to update the student's knowledge state. This integration boosts prediction accuracy with minimal modifications to the existing algorithms, as shown in Fig. 1 [6,19,22,26,28,29,42,46].

However, this model's performance is intrinsically tied to the quality of the underlying KT model. If the base KT model is inadequately tuned, the features generated by the LLMs may not be fully leveraged, thus limiting the model's capacity to enhance predictive accuracy. Future research in this domain could

explore incorporating multimodal data, such as audio, video, or sensor data, to further expand the feature extraction capabilities of LLMs, thereby enabling richer and more robust representations of student behavior.

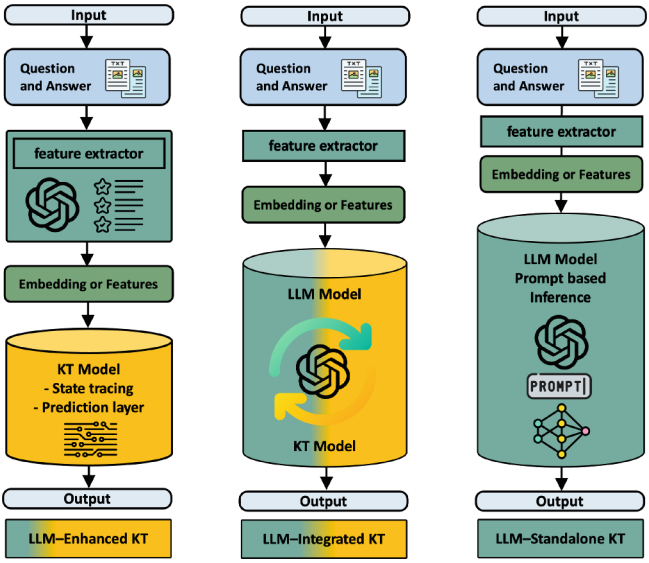

Fig. 2. Overview of LLM-based KT Architectures.

LLM-Integrated KT: Dynamic Knowledge State Updates. Figure 2 illustrates the more sophisticated nature of the LLM-Integrated approach, where LLMs play a central role in dynamically updating the student's knowledge state [3,5,16,18,20,24,38,39,43].

LLMs in this taxonomy are not merely employed as feature extractors; instead, they are deeply embedded into the KT architecture, directly influencing the student's knowledge state based interactions. In this model, LLMs continuously analyze incoming student responses and update the knowledge state. This process is dynamic, with LLMs evaluating the student's current knowledge and adjusting predictions based on their evolving performance.

However, the dynamic processing requirements of LLMs present significant computational challenges. The need to process large volumes of student interaction data rapidly can result in performance bottlenecks, especially in large-scale applications. Furthermore, LLM-based models often suffer from opacity, with limited interpretability of their decision-making processes. This lack of transparency is a critical issue in educational settings, where understanding the rationale behind feedback is crucial for both educators and students.

To address these challenges, future research should focus on optimizing the efficiency of LLMs for dynamic applications. Techniques such as model distillation, pruning, or quantization could be explored to reduce computational overhead while preserving performance. Additionally, improving the transparency and explainability of LLM-based predictions will be essential to enhance user trust and facilitate the broader adoption of these models in educational contexts.

LLM-Standalone KT: Autonomous Knowledge State Prediction. As shown in Fig. 2, the LLM-Standalone approach represents a significant departure from traditional KT models by completely removing the need for external KT components. Instead, LLMs autonomously track and predict the student's knowledge state via prompt-based inference. This model leverages LLMs to handle both knowledge state tracking and performance prediction, offering a fully integrated, end-to-end solution. By processing immediate student input, the LLMs can provide real-time feedback and predictions, making this model particularly effective in dynamic and fast-paced educational environments. Unlike traditional KT models, the LLM-Standalone approach does not rely on external modules but instead employs self-supervised learning techniques to predict outcomes based on previous interactions. The model benefits from LLMs' ability to generalize knowledge from extensive pre-trained corpora [11,13–15,17,27,35,45].

However, one of the significant challenges with LLM-Standalone KT is the phenomenon of hallucinations, where the model generates predictions or feedback that are not grounded in the provided input, leading to inaccurate or misleading outputs. Additionally, unlike traditional KT models, which offer explicit control over specific knowledge components, such as skill-level estimation, LLM-Standalone KT lacks the same level of granularity and interpretability. This limitation can hinder the model's ability to provide fine-grained, skill-specific tracking of student progress.

Therefore, future research should focus on mitigating the hallucination problem by incorporating techniques such as calibration or confidence scoring, which could ensure more reliable predictions. Additionally, exploring methods to enhance control over specific competencies within the LLMs would allow for more precise tracking of student knowledge, making it possible to achieve a finer level of granularity in performance prediction and analysis.

3.2 Cross-Comparative Analysis

Datasets and Domains. Datasets are fundamental to the architecture and performance of KT models, as they define the input features and the scope of inference. Figure 3 provides a comparative analysis of the three LLM-based KT approaches based on the datasets used and the domains they target. The choice of dataset influences the model's ability to generalize and accurately predict learners' knowledge states.

In the LLM-Enhanced KT model, LLMs are incorporated into existing KT architectures to augment prediction capabilities by adding semantic features

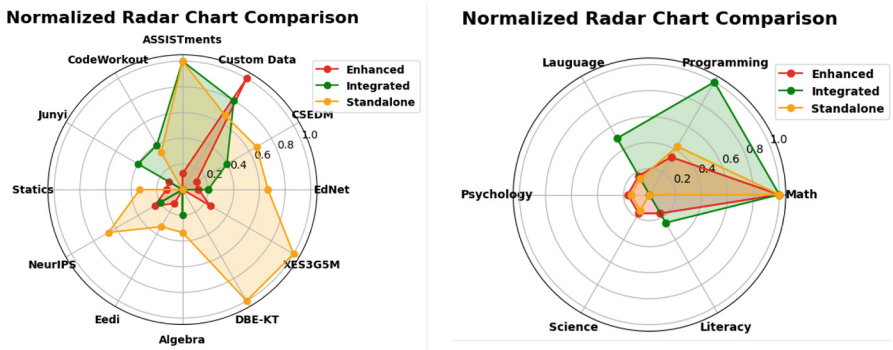

Fig. 3. Comparative Evaluation of LLM-Enhanced, Integrated, and Standalone KT Models Across Datasets and Domains. The left panel illustrates datasets, and the right panel illustrates targeted domains for each model type.

extracted from unstructured data. Given the nature of this model, it typically utilizes custom datasets, which may be collected from diverse sources or tailored to specific domains. The richness of the dataset plays a critical role in ensuring that the LLM can effectively capture nuanced semantic information, thus enhancing the model's performance.

LLM-Integrated KT, in contrast, relies heavily on its rich, domain-specific datasets, as LLMs are highly adept at understanding text. However, because LLMs directly interact with the KT model, handling noisy or unbalanced data becomes a key challenge. Consequently, pre-processing and balancing of the data are crucial steps for achieving optimal model performance. This approach often leverages large, structured educational datasets such as ASSISTments and EdNet, which are designed to provide detailed and diverse learning data.

The LLM-Standalone KT model operates independently of traditional KT modules, relying solely on LLMs for both tracking and prediction. This approach typically uses structured datasets, such as ASSISTments or XES3GSM, which could be prompt-based learning. The structured nature of these datasets helps stabilize model performance by simplifying the underlying complexity and making it easier to interpret the predictions. By using these well-organized datasets, LLM-Standalone systems can focus on leveraging prompt-based inference without being overwhelmed by the intricacies of unstructured data.

As illustrated on the right side of Fig. 3, all three approaches are commonly applied to math and programming data, which feature well-defined question-answer structures. These domains are heavily studied within KT due to their clarity and predictable response patterns. However, for domains rich in text, the LLM-Integrated model proves to be more versatile, offering the flexibility to handle diverse and less structured inputs. On the other hand, the LLM-Enhanced and LLM-Standalone models tend to focus on more structured, question-oriented datasets, which are better suited to their inherent design.

Performance Metrics. The increasing reliance on LLMs in KT models necessitates the use of appropriate performance metrics to evaluate the effectiveness and accuracy of these models. Figure 4 compares the performance metrics commonly used to assess LLM-based KT approaches.

In the LLM-Enhanced KT model, traditional metrics such as AUC and ACC are frequently used to evaluate the prediction accuracy of the enhanced system. These metrics are commonly employed in KT models to assess their ability to rank and predict learner responses. Since LLM-Enhanced models focus on improving the predictions of existing KT frameworks, they rely on these conventional metrics to measure their performance in comparison to baseline models.

For LLM-Integrated KT, the dynamic and interactive nature of the model requires additional evaluation metrics beyond traditional ones. In particular, CodeBLEU and Dist-1 are used to assess the performance of these models. CodeBLEU is designed to evaluate the accuracy and semantic consistency of generated code, which is especially useful when the KT model interacts with code-related data. Dist-1, on the other hand, evaluates the diversity of the generated output, which is important for assessing the model's ability to capture and predict a variety of learner solutions. These metrics ensure that the LLM-Integrated model can handle complex, varied learner inputs and track their progress effectively.

The LLM-Standalone KT model, relying solely on LLMs for inference, employs a combination of AUC, ACC, and F1-Score. The inclusion of the F1-Score is particularly important, as it balances precision and recall, making it ideal for evaluating performance in contexts where predictions may be imbalanced, such as when dealing with unstructured text data. F1-Score helps determine how well the model can detect incorrect answers, especially in cases where misclassifications may have significant consequences. Given that prompt design plays a crucial role in the reliability of LLM-Standalone models, balancing precision and recall is essential for ensuring that predictions are both accurate and reliable.

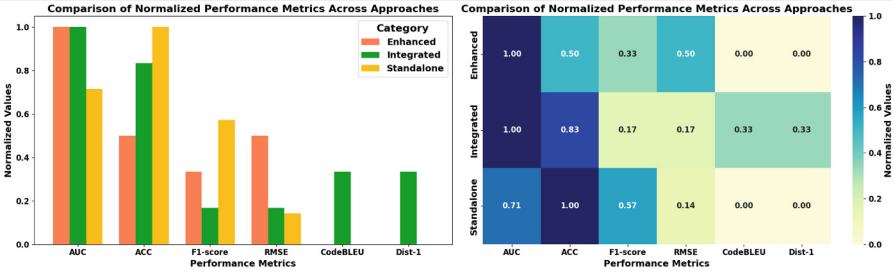

Fig. 4. Comparison of Performance Metrics Used to Evaluate LLM-based KT. The left chart shows the frequency of metric usage, while the right chart visualizes the metric values normalized by frequency.

The comparative evaluation of these performance metrics highlights the nuanced differences between the three approaches. While AUC and ACC remain the most commonly used metrics across all approaches, LLM-Integrated models require more specialized metrics to handle the complexities of interactive, dynamic learning environments. The addition of metrics like CodeBLEU and Dist-1 reflects the need for a more granular understanding of the model's performance in diverse and open-ended contexts. Conversely, LLM-Standalone models prioritize balance metrics, such as F1-Score, to address issues of imbalance and ensure that predictions maintain high reliability across a variety of learner inputs. This comprehensive evaluation underscores the importance of selecting the appropriate metrics to capture the full range of model performance, depending on the specific design and use case of the LLM-based KT system.

4 Discussion

The taxonomy introduced in this study aims to address a critical gap in the evolving landscape of KT research, particularly with respect to the integration of LLMs. While previous research has explored LLMs applications across a variety of domains, there remains a lack of a standardized taxonomy to systematically categorize these applications within KT. Our taxonomy, consisting of LLM-Enhanced, LLM-Integrated, and LLM-Standalone approaches, provides a structured way to assess and understand the various strategies for embedding LLMs into KT systems. This classification offers both theoretical insights and practical guidance for future implementations of LLM-based solutions in educational technology.

LLM-Enhanced systems primarily augment existing KT pipelines by integrating LLMs as additional feature extractors. These systems leverage the semantic power of LLMs to extract meaningful insights from unstructured data such as textual feedback or open-ended responses. This augmentation is particularly valuable in scenarios where traditional KT models are effective at tracking overall learner mastery but may struggle to capture subtle semantic cues embedded in learners' interactions. By enhancing these models with LLM-derived features, the system can capture more nuanced information, potentially improving prediction accuracy and learner modeling.

In contrast, LLM-Integrated systems rely on dynamic adaptation, making them suitable for learning environments that require constant updates and interpretations of learner data. These models are particularly effective in contexts where interactions are frequent and rich, such as in dialogue-based learning or interactive simulations. However, their reliance on dynamic data also makes them susceptible to noise and inconsistencies, such as unbalanced inputs in conversations or feedback loops that may distort learning assessments. This vulnerability highlights the challenge of managing and filtering noisy data in systems that depend on continuous feedback from learners.

On the other hand, LLM-Standalone approaches propose a more autonomous solution, where the LLM itself drives the entire knowledge tracing process without reliance on traditional KT models. This approach offers the potential for a

unified, prompt-driven solution capable of independently extracting both semantic and predictive signals from learner data. However, the scalability of this approach is limited when dealing with complex, nuanced, or domain-specific content. As such, LLM-Standalone models may work well in well-defined domains with clear, structured input but struggle in environments where content is more ambiguous or not sufficiently predefined.

Despite the promise of these approaches, there is a clear lack of standardized benchmark datasets tailored to LLM-based KT. The scattered nature of existing datasets across various domains complicates direct comparisons of model performance. The development of domain-specific benchmarks is crucial not only for evaluating the effectiveness of LLM-based solutions but also for enabling fair, consistent comparisons between different LLM-KT models. These benchmarks should also include fixed performance metrics to ensure the reliability and reproducibility of results, offering a foundation for further research and refinement.

The ambiguity surrounding the roles of LLMs in some hybrid systems further complicates the landscape. In cases where LLMs serve dual roles—both enhancing feature extraction and directly updating knowledge states—there is potential overlap between the categories of LLM-Enhanced and LLM-Integrated models. This gray area suggests the need for more fine-grained sub-classifications as the field matures, allowing for a more precise understanding of how LLMs can be utilized in different KT contexts.

Looking forward, further research is needed to empirically test the effectiveness of each category under controlled experimental conditions. Standardized experiments could assess the relative contributions of LLM-driven features versus traditional KT representations, thereby clarifying when LLM-Enhanced methods are sufficient and when more integrated or standalone approaches are warranted. Additionally, exploring the compatibility of different LLM architectures, such as GPT-based systems versus other model classes, could yield insights into the optimal use of LLMs in real-time feedback loops and structured item-response data. By situating these investigations within the taxonomy framework, researchers can gain a deeper understanding of how LLMs can be best integrated into KT systems, ultimately leading to the advancement of intelligent tutoring technologies.

5 Conclusion

Each of the three approaches to LLM-based Knowledge Tracing presents distinct benefits and challenges. The LLM-Enhanced model excels in augmenting existing systems by incorporating richer semantic features, improving the overall prediction accuracy of traditional KT methods. On the other hand, the LLM-Integrated model introduces a higher level of adaptability by enabling dynamic updates to learners' knowledge states, making it well-suited for interactive environments that require constant feedback. However, its reliance on continuous data inputs also introduces potential vulnerabilities, particularly in noisy or unbalanced learning contexts. Meanwhile, the LLM-Standalone approach offers

a more simplified, autonomous solution that reduces the dependency on traditional KT systems. Despite its potential for scalability and real-time predictions, it faces limitations when applied to complex or undefined content areas.

To fully harness the potential of LLMs in educational contexts, future research must address several key challenges. Optimizing computational efficiency will be crucial, particularly for real-time systems that need to handle large volumes of dynamic data. Additionally, improving model interpretability will help bridge the gap between the complex inner workings of LLMs and the practical needs of educators and learners. Finally, ensuring the reliability and scalability of LLM-based KT models will be vital for their widespread adoption and effectiveness across diverse educational environments. By tackling these challenges, future work can unlock the full promise of LLMs, advancing the field of intelligent tutoring systems and personalized learning.

References

1. Cao, Y., et al.: Survey on large language model-enhanced reinforcement learning: concept, taxonomy, and methods. IEEE Trans. Neural Netw. Learn. Syst., 1–21 (2024). https://doi.org/10.1109/tnnls.2024.3497992. http://dx.doi.org/10.1109/TNNLS.2024.3497992
2. Chowdhery, A., et al.: PaLM: scaling language modeling with pathways. J. Mach. Learn. Res. **24**(240), 1–113 (2023)
3. Duan, Z., Fernandez, N., Hicks, A., Lan, A.: Test case-informed knowledge tracing for open-ended coding tasks. In: Proceedings of the 15th International Learning Analytics and Knowledge Conference, pp. 238–248 (2025)
4. Feng, Z., et al.: Trends in integration of knowledge and large language models: a survey and taxonomy of methods, benchmarks, and applications (2024). https://arxiv.org/abs/2311.05876
5. Fu, L., et al.: SINKT: a structure-aware inductive knowledge tracing model with large language model. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, pp. 632–642 (2024)
6. Guo, Y., et al.: Mitigating cold-start problems in knowledge tracing with large language models: an attribute-aware approach. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, pp. 727–736 (2024)
7. Han, Z., Gao, C., Liu, J., Zhang, J., Zhang, S.Q.: Parameter-efficient fine-tuning for large models: a comprehensive survey. arXiv preprint arXiv:2403.14608 (2024)
8. Hasan, S.M.: Multidimensional human activity recognition with large language model: a conceptual framework. arXiv preprint arXiv:2410.03546 (2024)
9. Jin, H., Huang, L., Cai, H., Yan, J., Li, B., Chen, H.: From LLMs to LLM-based agents for software engineering: a survey of current, challenges and future (2024). https://arxiv.org/abs/2408.02479
10. Jin, M., et al.: Time-LLM: time series forecasting by reprogramming large language models. arXiv preprint arXiv:2310.01728 (2023)
11. Jung, H., Yoo, J., Yoon, Y., Jang, Y.: CLST: cold-start mitigation in knowledge tracing by aligning a generative language model as a students' knowledge tracer. arXiv preprint arXiv:2406.10296 (2024)

12. Jyothy, S.N., Kolil, V.K., Raman, R., Achuthan, K.: Exploring large language models as an integrated tool for learning, teaching, and research through the fogg behavior model: a comprehensive mixed-methods analysis. Cogent Eng. **11**(1) (2024)
13. Kim, J., Chu, S., Wong, B., Yi, M.: Beyond right and wrong: mitigating cold start in knowledge tracing using large language model and option weight. arXiv preprint arXiv:2410.12872 (2024)
14. Lee, U., et al.: From prediction to application: language model-based code knowledge tracing with domain adaptive pre-training and automatic feedback system with pedagogical prompting for comprehensive programming education. arXiv preprint arXiv:2409.00323 (2024)
15. Lee, U., et al.: Language model can do knowledge tracing: simple but effective method to integrate language model and knowledge tracing task. arXiv preprint arXiv:2406.02893 (2024)
16. Lee, U., et al.: Difficulty-focused contrastive learning for knowledge tracing with a large language model-based difficulty prediction. arXiv preprint arXiv:2312.11890 (2023)
17. Li, H., et al.: Explainable few-shot knowledge tracing. arXiv preprint arXiv:2405.14391 (2024)
18. Li, Z., et al.: TutorLLM: customizing learning recommendations with knowledge tracing and retrieval-augmented generation. arXiv preprint arXiv:2502.15709 (2025)
19. Liang, Z., Yu, W., Rajpurohit, T., Clark, P., Zhang, X., Kaylan, A.: Let GPT be a math tutor: Teaching math word problem solvers with customized exercise generation. arXiv preprint arXiv:2305.14386 (2023)
20. Liu, N., Wang, Z., Baraniuk, R., Lan, A.: Open-ended knowledge tracing for computer science education. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (2022)
21. Liu, Q., Shen, S., Huang, Z., Chen, E., Zheng, Y.: A survey of knowledge tracing. arXiv preprint arXiv:2105.15106 (2021)
22. Liu, Z.: XES3G5M: a knowledge tracing benchmark dataset with auxiliary information. Adv. Neural. Inf. Process. Syst. **36**, 32958–32970 (2023)
23. Long, T., et al.: Automatical graph-based knowledge tracing. In: EDM (2022)
24. Makharia, R., et al.: AI tutor enhanced with prompt engineering and deep knowledge tracing. In: 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), vol. 2, pp. 1–6. IEEE (2024)
25. Mirchandani, S., et al.: Large language models as general pattern machines. arXiv preprint arXiv:2307.04721 (2023)
26. Moon, H., Davis, R., Neshaei, S.P., Dillenbourg, P.: Using large multimodal models to extract knowledge components for knowledge tracing from multimedia question information. arXiv preprint arXiv:2409.20167 (2024)
27. Neshaei, S.P., Davis, R.L., Hazimeh, A., Lazarevski, B., Dillenbourg, P., Käser, T.: Towards modeling learner performance with large language models. arXiv preprint arXiv:2403.14661 (2024)
28. Ni, L., et al.: Enhancing student performance prediction on learner sourced questions with SGNN-LLM synergy. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, pp. 23232–23240 (2024)
29. Ozyurt, Y., Feuerriegel, S., Sachan, M.: Automated knowledge concept annotation and question representation learning for knowledge tracing. arXiv preprint arXiv:2410.01727 (2024)

30. Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. arXiv preprint arXiv:1907.06837 (2019)
31. Piech, C., et al.: Deep knowledge tracing. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
32. Pternea, M., et al.: The RL/LLM taxonomy tree: Reviewing synergies between reinforcement learning and large language models. J. Artif. Intell. Res. **80**, 1525–1573 (2024). https://doi.org/10.1613/jair.1.15960. http://dx.doi.org/10.1613/jair.1.15960
33. Qu, G., Chen, Q., Wei, W., Lin, Z., Chen, X., Huang, K.: Mobile edge intelligence for large language models: a contemporary survey (2024). https://arxiv.org/abs/2407.18921
34. Salomons, N., Scassellati, B.: Time-dependant Bayesian knowledge tracing–robots that model user skills over time. Front. Robot. AI **10** (2024)
35. Scarlatos, A., Baker, R.S., Lan, A.: Exploring knowledge tracing in tutor-student dialogues using LLMs. In: Proceedings of the 15th International Learning Analytics and Knowledge Conference, pp. 249–259 (2025)
36. Shen, S., et al.: A survey of knowledge tracing: models, variants, and applications. IEEE Trans. Learn. Technol. (2024)
37. Wang, C., Sahebi, S.: Continuous personalized knowledge tracing: modeling long-term learning in online environments. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pp. 2616–2625 (2023)
38. Wang, D., Zhang, L., Zhao, Y., Zhang, Y., Yan, S., Hou, M.: Deep knowledge tracking integrating programming exercise difficulty and forgetting factors. In: International Conference on Intelligent Computing, pp. 192–203. Springer (2024)
39. Wang, Z., et al.: LLM-KT: aligning large language models with knowledge tracing using a plug-and-play instruction. arXiv preprint arXiv:2502.02945 (2025)
40. Weber, I.: Large language models as software components: a taxonomy for LLM-integrated applications (2024). https://arxiv.org/abs/2406.10300
41. Xi, Z., et al.: The rise and potential of large language model based agents: a survey (2023). https://arxiv.org/abs/2309.07864
42. Xia, J., Wang, H., Zhuge, Q., Sha, E.: Knowledge tracing model and student profile based on clustering-neural-network. Appl. Sci. **13**(9), 5220 (2023)
43. Yu, Y., Zhou, Y., Zhu, Y., Ye, Y., Chen, L., Chen, M.: ECKT: enhancing code knowledge tracing via large language models. In: Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 46 (2024)
44. Zanellati, A., Di Mitri, D., Gabbrielli, M., Levrini, O.: Hybrid models for knowledge tracing: a systematic literature review. IEEE Trans. Learn. Technol. (2024)
45. Zhan, B., et al.: Knowledge tracing as language processing: a large-scale autoregressive paradigm. In: International Conference on Artificial Intelligence in Education, pp. 177–191. Springer (2024)
46. Zhang, L., et al.: Predicting learning performance with large language models: a study in adult literacy. In: International Conference on Human-Computer Interaction, pp. 333–353. Springer (2024)
47. Zhang, Q., et al.: A survey on large language models for software engineering (2024). https://arxiv.org/abs/2312.15223
48. Zhao, W.X., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)