Towards Modeling Learner Performance with Large Language Models

Seyed Parsa Neshaei*
EPFL
seyed.neshaei@epfl.ch

Richard Lee Davis*
EPFL
richard.davis@epfl.ch

Adam Hazimeh
EPFL
adam.hazimeh@epfl.ch

Bojan Lazarevski
EPFL
bojan.lazarevski@epfl.ch

Pierre Dillenbourg
EPFL
pierre.dillenbourg@epfl.ch

Tanja Käser
EPFL
tanja.kaeser@epfl.ch

ABSTRACT

This paper investigates whether the pattern recognition and sequence modeling capabilities of LLMs can be extended to the domain of knowledge tracing, a critical component in the development of intelligent tutoring systems (ITSs) that tailor educational experiences by predicting learner performance over time. In an empirical evaluation across multiple real-world datasets, we compare two approaches to using LLMs for this task, zero-shot prompting and model fine-tuning, with existing, non-LLM approaches to knowledge tracing. While LLM-based approaches do not achieve state-of-the-art performance, fine-tuned LLMs surpass the performance of naive baseline models and perform on par with standard Bayesian Knowledge Tracing approaches across multiple metrics. These findings suggest that the pattern recognition capabilities of LLMs can be used to model complex learning trajectories, opening a novel avenue for applying LLMs to educational contexts[1].

Keywords

large language models, knowledge tracing, student modeling

1. INTRODUCTION

Pre-trained large language models (LLMs), such as BERT [12] and GPT-3 [4], are transformer-based neural networks containing large numbers of parameters (e.g., from 110 million in BERT to 175 billion in GPT-3) trained on massive amounts of natural language data. These models have demonstrated an impressive ability for zero-shot and few-shot learning, which is the ability to generalize to novel tasks when provided with a handful of examples or task instructions in natural language. For example, LLMs have matched or exceeded the performance of bespoke models on several benchmarks, such as computer programming [9], medicine [43], and mathematics [22], demonstrating their ability to leverage learned language patterns across a variety of domains.

In the domain of education, one of the emerging uses of LLMs is integration into intelligent tutoring systems (ITSs) [16, 3]. Recent work suggests that LLMs are capable of providing personalized learning experiences to students in multiple domains [40] and in various curricula and topics, including programming [5], biology [35], math [61], or chemistry [53].

So far, the actual work of modeling students' knowledge and performance has not been handled by LLMs directly. Instead, this task is accomplished by knowledge tracing models that track how students learn over time through their interactions with the ITS. Knowledge tracing models are typically applied to the question-answering part of an ITS and model the evolution of student knowledge based on their responses to previous questions [2, 11, 37].

Knowledge tracing models in the EDM literature can be categorized broadly into distinct families, each with unique approaches to modeling student learning. The first family of approaches models student knowledge using Markov models, as typified by Bayesian Knowledge Tracing (BKT) [3]. Variations of BKT have been developed to incorporate additional dimensions such as student assistance received [23], task difficulty [31], and individualized adjustments to learning and error probabilities [58], enhancing the model's ability to personalize learning assessments and make predictions.

A second family of approaches uses logistic regression to model learning and predict performance. Logistic regression predictions take the form of $p(a_{s,t+1} = 1|q_{s,t+1}, \mathbf{x}_{s,1:t}) = \sigma(\mathbf{w}^T \Phi(q_{s,t+1}, \mathbf{x}_{s,1:t}))$ where $a_{s,t+1}$ is the response prediction of learner s at timestep $t+1$, $q_{s,t+1}$ is the question of learner s at timestep $t+1$, $\mathbf{x}_{s,1:t}$ is all of the data for learner s up to timestep t, σ is the logistic function, $\mathbf{w} \in \mathbb{R}^d$ is a trainable weight vector, and $\Phi(q_{s,t+1}, \mathbf{x}_{s,1:t})$ is is a vector of d features representing the learner, question q_{t+1}, and history $\mathbf{x}_{1:t}$. Different feature vectors Φ are associated with different models. For example, Performance Factors Analysis (PFA) [32] creates a feature vector that includes the difficulty of each knowledge component k associated with question q_{t+1} and the number of correct and incorrect answers on each

*Equal contribution

[1]Our code and links to data are available on https://github.com/spneshaei/LLM-Learner-Modeling

S. P. Neshaei, R. L. Davis, A. Hazimeh, B. Lazarevski, P. Dillenbourg, and T. Käser. Towards modeling learner performance with large language models. In B. Paaßen and C. D. Epp, editors, *Proceedings of the 17th International Conference on Educational Data Mining*, pages 759–768, Atlanta, Georgia, USA, July 2024. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.12729942

knowledge component k by learner s prior to timestep $t+1$. Other approaches in this family include Learning Factors Analysis (LFA) [7], DAS3H (which includes continuous time window features) [10], and Best-LR (which is similar to PFA but includes the total prior number of correct and wrong responses in the features) [14].

A third family of knowledge tracing approaches utilizes deep neural networks to model student performance. This approach was pioneered by Piech et al. with the Deep Knowledge Tracing (DKT) model [33], which used a recurrent neural network architecture to process student activity and jointly model skill acquisition at each timestep. Recent advancements in knowledge tracing include the introduction of dynamic memory models like DKVMN [59] and SKVMN [1], which incorporate next skill input and LSTM integration, respectively; EERNN [41] and EKT [24], which utilize exercise textual content; and SAKT [30], which uses self-attentive transformer models for knowledge tracing.

The goal of the current paper is to explore whether LLMs might serve as the basis for a new family of methods for student performance prediction. In this proposed family, student interaction data would be represented by sequences of natural language tokens, and a complete dataset would form a kind of specialized sub-language. Modeling and prediction tasks would rely on an LLM's ability to recognize or learn the underlying linguistic structure in these sub-languages and extrapolate from seen to unseen data. This approach to student performance prediction is inspired by recent work showing that, under the right conditions, LLMs can act as general pattern machines [28], identifying patterns in sequences of tokens representing numerical data [17, 6, 60, 8]. These abilities have been demonstrated in multiple domains, including healthcare [25], finance [57, 26], and transportation [56]. However, to the best of our knowledge, there has been no work investigating the abilities of LLMs to model and predict time-series data about student performance in educational contexts.

Our primary research question, **RQ1**, is: *Can the general pattern matching abilities of LLMs be applied to the domain of student performance prediction?* To answer this question, we evaluate two distinct approaches to applying LLMs in this domain, zero-shot prediction and model fine-tuning. For each approach, we ask the question: **RQ2:** *How does the performance of the LLM on three real-world datasets compare to (a) naive baselines, (b) Markov approaches to knowledge tracing, (c) logistic regression approaches, and (d) deep learning approaches to knowledge tracing?* Finally, in the context of fine-tuning, we investigate the question: **RQ3:** *Is there a relationship between the scale of a fine-tuned model (in terms of parameter count) and its ability to predict student performance?*

In the current work, we selected three datasets from the literature which were previously used in knowledge tracing research. We evaluated our two LLM-based approaches on the data, as well as comparing them with naive baselines and a range of standard knowledge tracing approaches across a set of metrics. We found while the zero-shot approaches were not successful in capturing student knowledge over time, fine-tuned LLMs beat the baselines and exhibited similar or higher performance than standard Bayesian models for knowledge tracing. Our results shed light on the applicability of LLMs as general pattern machines to the task of knowledge tracing, and enhance EDM research by contributing to the research line of the effectiveness of LLMs in pedagogical scenarios.

2. EVALUATED MODELS

To compare how LLMs can reliably predict the students' answers to the questions and thus help in knowledge tracing, we compared their performance against well-established traditional approaches to knowledge tracing.

2.1 Problem Formulation

To model students' knowledge over time, we followed the approach proposed by Gervet et al. [14]. We define S as the sequence of answers provided by a student, and $S_{:i}$ as the sub-sequence of answers provided up to and before point i in sequence S. We also define K_i as the ID indicating the skill (i.e., knowledge component) of the current next question. At each point i, we aim to predict the correctness of the student response in the next step, using data from $S_{:i}$. For simplicity, we consider *correct* answer as 1, and *wrong* answer as 0.

2.2 Baselines and Previous Models

For our naive baselines, we used three approaches:

Mean Following [36], we found the mean of all the responses in any given dataset and constantly predicted the same value as a simple baseline. We also computed the (constant) probability of each prediction as the count of responses equal to the mean response, divided by the total number of responses in the whole training dataset.

Next as Previous (NaP) Taking idea from the approach used by [36], at each step i, we considered the mean of all previous responses in $S_{:i}$ as the next prediction. We also computed the probability of each prediction as the count of responses equal to the mean response in $S_{:i}$, divided by the total number of responses in $S_{:i}$.

Skillwise Next as Previous (NaP Skills) We extended the NaP baseline by only considering the mean of the previous responses in $S_{:i}$ with the same skill ID as of the current next question (K_i), as the next prediction. We also computed the probability of each prediction as the count of responses in $S_{:i}$ equal to the mean response and with skill ID K_i, divided by the total number of responses in $S_{:i}$ with skill ID K_i.

Additionally, we evaluated four other approaches to knowledge tracing, shown to be useful in the literature, and described briefly in the following:

Bayesian Knowledge Tracing (BKT) A standard BKT approach which modeled student knowledge as a latent variable, determined by four parameters per each skill: A) prior learning, B) probability of moving from the *not-knowledgable* state to *knowledgable* state for a skill, C) probability of applying a skill correctly by accident, and D) probability of applying a skill incorrectly by accident [36, 58].

Deep Knowledge Tracing (DKT) A canonical DKT model that utilized an RNN-based long short-term memory (LSTM) deep architecture [33] which helps to reduce information loss due to the integrated gate structure [36, 19, 18].

Best-Logistic Regression (Best-LR) A logistic regression model introduced by [14] which was trained on the total number of previous success and error counts in $S_{:i}$ at each step i.

Self-Attentive Knowledge Tracing (SAKT) A transformer-based model [46], first introduced by [30], which uses an embedding of information at point i and predicts a probability for correctness at each step using a one-neuron output layer.

2.3 Fine-tuning LLMs

We constructed a set of *prompt-completion* pairs to fine-tune LLMs for the task of knowledge tracing. Each *prompt* includes the details needed for prediction at step i, and the *completion* includes the answer to be predicted for the current question at each step.

Our approach to constructing input prompts was directly inspired by logistic regression approaches to knowledge tracing [14] which convert student interaction data into a set of features $\Phi(q_{s,t+1}, \mathbf{x}_{s,1:t})$. Where our approach differs is that we represent this set of features Φ using natural language, and pass the set of natural language features to an LLM which then generates a prediction.

Thus, for our *prompts*, we used two sets of features as inputs to our models, also among the feature sets used by [14], for each point i. The *completion* of each prompt is either "CORRECT" or "WRONG".

1) Minimal Prompt: This prompt includes the ID of the current question ($= A$), the total number of correct answers for all prior questions in $S_{:i}$ ($= B$), and the total number of wrong answers for all prior questions in $S_{:i}$ ($= C$). We thus formed our *prompt* at each step i accordingly as:

"*Total correct until now: B*
Total wrong until now: C
Current question ID: A
Student response: "

2) Extended Prompt: This prompt includes all of the features in the minimal prompt, as well as the following features: K_i, the total number of correct answers for all prior questions in $S_{:i}$ with skill ID equal to K_i ($= D$), and the total number of wrong answers for all prior questions in $S_{:i}$ with skill ID equal to K_i ($= E$). We thus formed our *prompt* at each step i accordingly as:

"*Current skill ID: K_i*
Total correct for prior questions with skill ID K_i: D
Total wrong for prior questions with skill ID K_i: E
Total correct until now: B
Total wrong until now: C
Current question ID: A
Student response: "

We split the digits of all of the numbers used in our prompts by adding a space between each two consecutive digits (for example, "342" was changed to "3 4 2"), as this has been shown to improve the performance when making predictions based on numerical data [17].

We experimented using three different LLMs with different model sizes to observe the effect of larger models on their performance in the task of knowledge tracing:

2.3.1 Fine-tuning BERT

The Bidirectional Encoder Representations from Transformers (BERT) architecture [12] is a Transformer-based model with around 110 million parameters trained using the masked language modeling approach. Models from the BERT family are commonly used for text classification [15, 42], especially in educational applications to provide automated feedback to teachers or students [20, 49, 55, 27]. We fine-tuned the `bert-base-cased` model provided by HuggingFace on our data for 3 epochs with a batch size of 32 and Adam optimizer having an initial learning rate of `3e-5`. We provided the *prompts* as inputs to the model, and the binarized version of the *completions* (1 for "CORRECT" and 0 for "WRONG") as labels per each input sentence.

2.3.2 Fine-tuning GPT-2

The Generative Pre-trained Transformer 2 (GPT-2) model [34], consisting of around 1.5 billion parameters, is capable of generating text completions, given partial text as input, and has been also used in studies on educational technologies [40, 38, 47, 45, 21, 29]. We used GPT-2 as provided by HuggingFace and fine-tuned it on our data for 2 epochs with a batch size of 4. As input data, we provided concatenated *prompts* and *completions*, and separated *prompt-completion* pairs belonging to each training data point using an `<|endoftext|>` special token.

2.3.3 Fine-tuning GPT-3

Introduced by Brown et al. [4], this model, consisting of 175 billion parameters, significantly improves upon GPT-2 by exhibiting pattern matching and few-show learning capabilities [4]. We fine-tuned `babbage-002`, a GPT-3 base model developed by OpenAI, on our training data using the interface provided by OpenAI with default settings. We used the `logprobs` functionality in OpenAI's API to calculate the probabilities for output tokens. We computed the probabilities of each of the words CORRECT and WRONG individually by summing probabilities of their starting tokens (e.g., "C" and "COR" for the word CORRECT) extracted from the `logprobs` data. We then normalized the values by dividing each of the two probabilities over their sum.

2.4 Zero-shot Prediction with GPT-3.5

GPT-3.5 is a recent model developed by OpenAI, trained using the Reinforcement Learning with Human Feedback (RLHF) approach, and incorporated in ChatGPT to support users by providing answers in the form of an intelligent conversational agent [54]. Researchers have used GPT-3.5 in *zero-shot* settings, in which they do not fine-tune the model, but rather ask the model to give an answer based on a single self-describing input prompt with no other contextual input [52, 48].

We used the same minimal prompt[2], as used in our fine-tuning process, for the *user message* of the prompt at each step i. Furthermore, to define the task for GPT-3.5 and

[2]Due to resource limitations, we did not conduct our zero-shot experiment on extended prompts, and leave it for future work.

instruct it to reply correctly given no training data, we included the following as the initial *system message* in each prompt:

"You are an instructor and want to predict whether a student will get a question CORRECT or WRONG. The only information you have is the student's previous answers to a series of related questions. You know how many questions they got CORRECT and how many they got WRONG. Based on this information, you should make a prediction by outputting a single word: CORRECT if you think the student will answer the next question correctly, and WRONG if you think the student will answer the next question wrong. Output no other word at all, this is very important. Try to estimate the knowledge of the student before making your prediction."

A diagram showing our prompting approach, along with an example, can be seen in Figure 1 in the appendix.

3. METHODOLOGY

3.1 Datasets

We evaluated the performance of the different models on three datasets with various sizes from the literature. For all datasets, we used the same train-test split as provided by [14]; the data of 80% of the users was used for training and the rest (20%) for testing. For the zero-shot prediction, due to the absence of the training phase, only the testing sets were used.

Statics This dataset, introduced by Steif and Bier [39], is extracted from the data of an engineering statics course. It consists of 1223 unique problems and 282 unique users. Over all entries, 76.5% (144883) are answered correctly and the rest (44414) are answered incorrectly.

ASSISTments 2009 This dataset is compiled from the data collected using the ASSISTments system [13]. It contains 17708 unique problems and 3114 unique users. Over all entries, 65.9% (183303) are answered correctly and the rest (95033) are answered incorrectly.

ASSISTments 2017 This dataset is also compiled from the data collected using the ASSISTments system [13]. It contains 3162 unique problems and 1708 unique users. Over all entries, 37.4% (349661) are answered correctly and the rest (584977) are answered incorrectly.

We pre-processed the data by discarding the entries of any student who had answered all questions correctly or all questions incorrectly. In this way, we only took into account the students who actively participated in a *learning* process, rather than those who already knew the topic or did not learn it over time. The removed students included 1.8% of all students in Statics, 5.0% of all students in ASSISTments 2009, and 0.0% of all students in ASSISTments 2017.

3.2 Evaluation and Metrics

We extracted the predicted responses of each of the models, along with their prediction probability, to compare the models we used in our downstream task. We used the following metrics, as defined and implemented in `scikit-learn` and previously used for evaluating models for the task of knowledge tracing [36], to evaluate our models: Accuracy (Acc), Balanced Accuracy (Bal Acc), Precision, Recall, F1, Area Under Curve (AUC), and Root Mean Square Error (RMSE).

4. RESULTS

We provide an overview of our results on each of the datasets in each of the sections below. First, we compare the two different LLM approaches, zero-shot prompting on GPT-3.5 and fine-tuning on GPT-3. Then, we compare the LLM approaches to results from naive baselines. Finally, we compare results between a set of widely used knowledge tracing models and the LLM approaches, with the goal of benchmarking the performance of the LLM approaches relative to the other approaches. Tables containing results from all of our evaluation metrics across the three datasets can be found in the appendix (Tables 1, 2, and 3). Plots comparing the AUC and RMSE scores across different models for each dataset can also be found in the appendix (Figures 2a, 2b, and 2c).

4.1 LLM Methods vs. Naive Baselines

We observed that zero-shot approaches behaved worse than or equal to our naive baselines across all our datasets with regard to AUC, with fine-tuned LLMs on each dataset achieving AUC scores of 0.18 to 0.21 more than the zero-shot approach. Additionally, we found that fine-tuned GPT-3 models beat all three naive baselines in the AUC score consistently across the three datasets. The differences in AUC score between the best-performing fine-tuned models and our range of naive baselines ranged from 0.03 to 0.21, consistently in favor of the fine-tuned GPT-3 models. In nearly all cases, LLMs that were fine-tuned and evaluated on extended prompts outperformed LLMs which were fine-tuned on the minimal set of features.[3]

4.2 Fine-Tuned LLMs vs. Other KT Models

In comparing the GPT-3 models fine-tuned on the set of extended prompts to previously-used knowledge tracing approaches, we found three general trends, persistent across all of the three datasets. First, we observed that the fine-tuned LLMs on the extended prompts achieved higher (for Statics and ASSISTments 2017) or similar (for ASSISTments 2009) AUC scores to the standard BKT model. Second, we observed that other knowledge tracing models (DKT, Best-LR, and SAKT) consistently outperformed our fine-tuned LLMs in terms of AUC score (ranging from 0.04 to 0.13 above the fine-tuned LLMs). Third, we found that the AUC scores of the fine-tuned LLMs generally improved relative to the other knowledge tracing models as the size of the training data increased. For example, the AUC score of the fine-tuned LLM on extended prompts was only 0.02 below the Best-LR and SAKT models in ASSISTments 2017, while this difference grew to 0.13 on the smaller Statics dataset.

4.3 A Note on Fine-Tuning GPT-2 and BERT

We opted not to present metrics associated with fine-tuning GPT-2 or BERT, as neither model achieved the objectives of our task effectively. GPT-2 struggled to consistently predict "CORRECT" or "WRONG" and often produced other different tokens. As a result, we could not obtain a single binary value for each of the predictions, and thus no computation of our metrics was possible. Regarding the text classification BERT model, the predictions made on the test set were

[3]Because of this trend, when reporting results from the fine-tuning approaches we will only discuss the extended approach.

all constant values, consistently favoring the majority class over all predictions. Given the constant values in its output, we deemed BERT ineffective for our task, and for brevity, did not report the associated results.

5. DISCUSSION

The goal of this work was to explore whether LLMs are capable of modeling learner performance, and if they might offer a new approach to the task of predicting student behavior given prior activity.

Regarding the first RQ (see Section 1), our results indicated that LLMs are, in fact, capable of predicting student performance. Even though the methods we developed were not able to achieve state-of-the-art results, our findings suggest that LLMs can serve as the basis for a new family of approaches to student performance modeling. This potential is further underscored by our investigation into the second and third research questions, which revealed that the effectiveness of LLMs in this domain is influenced by a variety of factors.

Regarding the second RQ (see Section 1), we found stark differences in performance between zero-shot and fine-tuning approaches across all three real-world datasets. In particular, the zero-shot approach failed to beat any of the naive baselines on any meaningful metric, while the fine-tuned LLMs achieved higher AUC than all the naive baselines across all three datasets. On the two smaller datasets, Statics and ASSISTments 2009, the fine-tuned LLM performed roughly equal to BKT. However, on the largest dataset, ASSISTments 2017, the fine-tuned LLM achieved higher AUC, lower RMSE, and higher balanced accuracy than BKT, and performed nearly as well as the other knowledge tracing models we used, with the exception of DKT. These findings indicate that the accuracy of fine-tuned models improves as the volume of training data grows. However, a comprehensive analysis is essential to fully understand the dynamics of this relationship, suggesting a promising avenue for further research.

The third RQ (see Section 1) focused on the relationship between model scale and predictive performance in fine-tuned LLMs. By comparing BERT and GPT-2 with GPT-3, we found evidence suggesting that the model scale has a notable impact on model performance in this domain. The difference in scale and number of the model parameters of BERT and GPT-2 compared to GPT-3 corresponded to a dramatic difference in performance, with BERT's fine-tuned performance failing to beat the naive baselines, and with fine-tuned GPT-2 producing unusable output. These findings suggest that modeling student performance may be an emergent ability [50], i.e., an ability that emerges in large language models but is not present in smaller models. However, these findings say nothing about whether predictive performance will continue to improve with increasing parameter counts, highlighting a compelling topic for future work within the EDM community.

We found that fine-tuning GPT-3 consistently beat zero-shot prompting of GPT-3.5 across all three datasets, demonstrating that our zero-shot approach, which presented only one data point at a time for prediction with no memory built over time, struggled to identify patterns of student knowledge and learning. A possible explanation for this difference has to do with the way that the knowledge tracing task is represented in the LLM regime. In this regime, a single student's interaction data is represented as a sequence of natural language tokens, and the full dataset containing each student's data forms a corpus capturing a kind of specialized sub-language. Modeling and prediction tasks thus rely on an LLM's ability to recognize or learn the underlying linguistic structure in the sub-language and extrapolate from seen to unseen data. Thus, one possible explanation for the difference in performance was that data points related to knowledge tracing and student modeling were underrepresented in the training dataset behind GPT-3.5. This may also explain why fine-tuning GPT-3 was more effective, as it made it possible for the model to leverage its existing abilities to learn the linguistic structure of the knowledge tracing sub-language. If this explanation is correct, our work shows that there exist natural language representations of the knowledge tracing task that allow an LLM to identify linguistic patterns of student behavior and to produce useful predictions about future performance.

5.1 Limitations and Future Work

We have identified several limitations and areas of concern that warrant further discussion. First, the absence of extensive publicly available details on the GPT-3 fine-tuning process limits understanding the mechanisms through which LLMs can learn to predict student performance. This issue underlines the importance of replicating these findings by fine-tuning an open-source LLM of comparable scale, such as LLaMA 2 [44]. Second, as our methods do not explicitly model the dynamics of student learning or knowledge acquisition, their usefulness in intelligent tutoring systems may be limited. One potential solution to this problem would be to use the fine-tuned model to make separate predictions about a student's future performance for each skill and to use these predictions to estimate a student's mastery level for each skill (similar to the method employed in [17]). However, this approach would still fail to explicitly model the latent constructs of interest, and the numerous predictions needed at each timestep would be resource-intensive.

Third, we found that incorporating more information into the prompts led to a modest improvement in the performance of fine-tuned models. This observation calls for a more systematic investigation into the impact of including different features (e.g., the difficulty level of each question) as prompts on model performance. Moreover, despite the fact that our attempts employing a zero-shot approach did not yield successful outcomes, we hypothesize that a chain-of-thought prompting strategy [51] might offer a viable alternative. Finally, our methodological approach involved condensing the history of student interactions into a concise prompt, drawing inspiration from logistic regression approaches. This strategy effectively circumvented the limitations imposed by the LLM's context length constraints but came at the expense of creating in-context learning opportunities for the model. We call for future works to investigate alternative strategies for incrementally feeding student interaction data into an LLM, allowing it to better engage with the temporal complexity of student interactions without breaching its context window limitations.

6. REFERENCES

[1] G. Abdelrahman and Q. Wang. Knowledge Tracing with Sequential Key-Value Memory Networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 175–184, Paris France, July 2019. ACM.

[2] G. Abdelrahman, Q. Wang, and B. Nunes. Knowledge tracing: A survey. *ACM Computing Surveys*, 55(11):1–37, 2023. Publisher: ACM New York, NY.

[3] J. R. Anderson, C. F. Boyle, and B. J. Reiser. Intelligent tutoring systems. *Science (New York, N.Y.)*, 228(4698):456–462, 1985. Publisher: American Association for the Advancement of Science.

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, and others. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] C. Cao. Scaffolding CS1 courses with a large language model-powered intelligent tutoring system. In *Companion proceedings of the 28th international conference on intelligent user interfaces*, pages 229–232, 2023.

[6] D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu. TEMPO: Prompt-based Generative Pre-trained Transformer for Time Series Forecasting, Oct. 2023. arXiv:2310.04948 [cs].

[7] H. Cen, K. Koedinger, and B. Junker. Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, M. Ikeda, K. D. Ashley, and T.-W. Chan, editors, *Intelligent Tutoring Systems*, volume 4053, pages 164–175. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. Series Title: Lecture Notes in Computer Science.

[8] C. Chang, W.-Y. Wang, W.-C. Peng, and T.-F. Chen. LLM4TS: Aligning Pre-Trained LLMs as Data-Efficient Time-Series Forecasters, Jan. 2024. arXiv:2308.08469 [cs].

[9] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating Large Language Models Trained on Code, July 2021. arXiv:2107.03374 [cs].

[10] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie. DAS3H: Modeling Student Learning and Forgetting for Optimally Scheduling Distributed Practice of Skills, May 2019. arXiv:1905.06873 [cs, stat].

[11] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4:253–278, 1994. Publisher: Springer.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] M. Feng, N. Heffernan, and K. Koedinger. Addressing the assessment challenge with an online system that tutors as it assesses. *User modeling and user-adapted interaction*, 19:243–266, 2009. Publisher: Springer.

[14] T. Gervet, K. Koedinger, J. Schneider, T. Mitchell, and others. When is deep learning the best approach to knowledge tracing? *Journal of Educational Data Mining*, 12(3):31–54, 2020.

[15] S. González-Carvajal and E. C. Garrido-Merchán. Comparing BERT against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012*, 2020.

[16] A. C. Graesser, M. W. Conley, and A. Olney. Intelligent tutoring systems. 2012. Publisher: American Psychological Association.

[17] N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson. Large Language Models Are Zero-Shot Time Series Forecasters, Oct. 2023. arXiv:2310.07820 [cs].

[18] S. Hochreiter and J. Schmidhuber. LSTM can solve hard long time lag problems. *Advances in neural information processing systems*, 9, 1996.

[19] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. Publisher: MIT press.

[20] E. Jensen, S. L. Pugh, and S. K. D'Mello. A deep transfer learning approach to modeling teacher discourse in the classroom. In *LAK21: 11th international learning analytics and knowledge conference*, pages 302–312, 2021.

[21] M. Lee, K. I. Gero, J. J. Y. Chung, S. B. Shum, V. Raheja, H. Shen, S. Venugopalan, T. Wambsganss, D. Zhou, E. A. Alghamdi, T. August, A. Bhat, M. Z. Choksi, S. Dutta, J. L. Guo, M. N. Hoque, Y. Kim, S. Knight, S. P. Neshaei, A. Sergeyuk, A. Shibani, D. Shrivastava, L. Shroff, J. Stark, S. Sterman, S. Wang, A. Bosselut, D. Buschek, J. C. Chang, S. Chen, M. Kreminski, J. Park, R. Pea, E. Rho, S. Zejiang Shen, and P. Siangliulue. A design space for intelligent and interactive writing assistants. *arXiv preprint arXiv:2403.14117*, 2024.

[22] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, and T. Gutman-Solo. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

[23] C. Lin and M. Chi. Intervention-BKT: Incorporating Instructional Interventions into Bayesian Knowledge Tracing. In A. Micarelli, J. Stamper, and K. Panourgia, editors, *Intelligent Tutoring Systems*, volume 9684, pages 208–218. Springer International Publishing, Cham, 2016. Series Title: Lecture Notes in

Computer Science.

[24] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, and G. Hu. EKT: Exercise-Aware Knowledge Tracing for Student Performance Prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):100–115, Jan. 2021. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[25] X. Liu, D. McDuff, G. Kovacs, I. Galatzer-Levy, J. Sunshine, J. Zhan, M.-Z. Poh, S. Liao, P. Di Achille, and S. Patel. Large Language Models are Few-Shot Health Learners, May 2023. arXiv:2305.15525 [cs].

[26] A. Lopez-Lira and Y. Tang. Can ChatGPT forecast stock price movements? return predictability and large language models. *arXiv preprint arXiv:2304.07619*, 2023.

[27] P. Mejia-Domenzain, J. Frej, S. P. Neshaei, L. Mouchel, T. Nazaretsky, T. Wambsganss, A. Bosselut, and T. Käser. Enhancing procedural writing through personalized example retrieval: a case study on cooking recipes. *International Journal of Artificial Intelligence in Education*, pages 1–37, 2024. Publisher: Springer.

[28] S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. G. Arenas, K. Rao, D. Sadigh, and A. Zeng. Large Language Models as General Pattern Machines, Oct. 2023. arXiv:2307.04721 [cs].

[29] S. P. Neshaei, R. Rietsche, X. Su, and T. Wambsganss. Enhancing peer review with AI-powered suggestion generation assistance: Investigating the design dynamics. In *Proceedings of the 29th international conference on intelligent user interfaces*, pages 88–102, 2024.

[30] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*, 2019.

[31] Z. A. Pardos and N. T. Heffernan. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In J. A. Konstan, R. Conejo, J. L. Marzo, and N. Oliver, editors, *User Modeling, Adaption and Personalization*, volume 6787, pages 243–254. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. Series Title: Lecture Notes in Computer Science.

[32] P. I. Pavlik Jr, H. Cen, and K. R. Koedinger. Performance Factors Analysis–A New Alternative to Knowledge Tracing. *Online Submission*, 2009. Publisher: ERIC.

[33] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep Knowledge Tracing. *Advances in neural information processing systems*, 28, 2015.

[34] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, and others. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[35] A. I. P. Sanpablo. Development and evaluation of a diagnostic exam for undergraduate biomedical engineering students using GPT language model-based virtual agents. In *XLVI mexican conference on biomedical engineering: Proceedings of CNIB 2023, november 2–4, 2023, villahermosa tabasco, méxico-volume 1: Signal processing and bioinformatics*, volume 96, page 128. Springer Nature, 2023.

[36] S. Sarsa, J. Leinonen, and A. Hellas. Empirical evaluation of deep learning models for knowledge tracing: Of hyperparameters and metrics on performance and replicability. *arXiv preprint arXiv:2112.15072*, 2021.

[37] T. Schodde, K. Bergmann, and S. Kopp. Adaptive robot language tutoring based on Bayesian knowledge tracing and predictive decision-making. In *Proceedings of the 2017 ACM/IEEE international conference on human-robot interaction*, pages 128–136, 2017.

[38] S. Shi, E. Zhao, D. Tang, Y. Wang, P. Li, W. Bi, H. Jiang, G. Huang, L. Cui, X. Huang, and others. Effidit: Your AI writing assistant. *arXiv preprint arXiv:2208.01815*, 2022.

[39] P. Steif and N. Bier. Oli engineering statics-fall 2011, 2014.

[40] X. Su, T. Wambsganss, R. Rietsche, S. P. Neshaei, and T. Käser. Reviewriter: AI-generated instructions for peer review writing. In *Proceedings of the 18th workshop on innovative use of NLP for building educational applications (BEA 2023)*, pages 57–71, 2023.

[41] Y. Su, Q. Liu, Q. Liu, Z. Huang, Y. Yin, E. Chen, C. Ding, S. Wei, and G. Hu. Exercise-Enhanced Sequential Modeling for Student Performance Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. Number: 1.

[42] C. Sun, X. Qiu, Y. Xu, and X. Huang. How to fine-tune BERT for text classification? In *Chinese computational linguistics: 18th china national conference, CCL 2019, kunming, china, october 18–20, 2019, proceedings 18*, pages 194–206. Springer, 2019.

[43] A. J. Thirunavukarasu, D. S. J. Ting, K. Elangovan, L. Gutierrez, T. F. Tan, and D. S. W. Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023. Publisher: Nature Publishing Group US New York.

[44] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. LLaMA 2: Open Foundation and Fine-Tuned Chat Models, July 2023. arXiv:2307.09288 [cs].

[45] D. Tsai, W. Chang, and S. Yang. Short answer questions generation by fine-tuning BERT and GPT-2. In *Proceedings of the 29th international conference on computers in education conference, ICCE*, 2021.

[46] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin.

Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[47] T. Wambsganss, X. Su, V. Swamy, S. P. Neshaei, R. Rietsche, and T. Käser. Unraveling downstream gender bias from large language models: A study on AI educational writing assistance. *arXiv preprint arXiv:2311.03311*, 2023.

[48] J. Wang, Y. Liang, F. Meng, B. Zou, Z. Li, J. Qu, and J. Zhou. Zero-shot cross-lingual summarization via large language models. In *Proceedings of the 4th new frontiers in summarization workshop*, pages 12–23, 2023.

[49] F. Weber, T. Wambsganss, S. P. Neshaei, and M. Soellner. Structured persuasive writing support in legal education: A model and tool for German legal case solutions. In *Findings of the association for computational linguistics: ACL 2023*, pages 2296–2313, 2023.

[50] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*, June 2022.

[51] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, Jan. 2023. arXiv:2201.11903 [cs].

[52] X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang, and others. Zero-shot information extraction via chatting with ChatGPT. *arXiv preprint arXiv:2302.10205*, 2023.

[53] A. D. White, G. M. Hocky, H. A. Gandhi, M. Ansari, S. Cox, G. P. Wellawatte, S. Sasmal, Z. Yang, K. Liu, Y. Singh, and others. Assessment of chemistry knowledge in large language models that generate code. *Digital Discovery*, 2(2):368–376, 2023. Publisher: Royal Society of Chemistry.

[54] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang. A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica*, 10(5):1122–1136, 2023. Publisher: IEEE.

[55] P. Wulff, L. Mientus, A. Nowak, and A. Borowski. Utilizing a pretrained language model (BERT) to classify preservice physics teachers' written reflections. *International Journal of Artificial Intelligence in Education*, 33(3):439–466, 2023. Publisher: Springer.

[56] H. Xue, B. P. Voutharoja, and F. D. Salim. Leveraging language foundation models for human mobility forecasting. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '22, pages 1–9, New York, NY, USA, Nov. 2022. Association for Computing Machinery.

[57] X. Yu, Z. Chen, Y. Ling, S. Dong, Z. Liu, and Y. Lu. Temporal data meets LLM–Explainable financial time series forecasting. *arXiv preprint arXiv:2306.11025*, 2023.

[58] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized Bayesian Knowledge Tracing Models. In *Artificial intelligence in education: 16th international conference, AIED 2013, memphis, TN, USA, july 9-13, 2013. Proceedings 16*, pages 171–180. Springer, 2013.

[59] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic Key-Value Memory Networks for Knowledge Tracing. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 765–774, Republic and Canton of Geneva, CHE, Apr. 2017. International World Wide Web Conferences Steering Committee.

[60] T. Zhou, P. Niu, X. Wang, L. Sun, and R. Jin. One Fits All:Power General Time Series Analysis by Pretrained LM, Oct. 2023. arXiv:2302.11939 [cs].

[61] M. Zong and B. Krishnamachari. Solving math word problems concerning systems of equations with GPT-3. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 15972–15979, 2023. Issue: 13.

APPENDIX

Table 1: Results for Statics dataset (Min = Minimal, Ext = Extended)

Family	Model	AUC	F1	RMSE	Acc	Bal Acc	Precision	Recall
Naive Baselines	Mean	0.50	**0.87**	0.42	**0.77**	0.50	0.77	1.00
	NaP	0.61	**0.87**	**0.41**	**0.77**	0.50	0.77	1.00
	NaP Skills	**0.63**	0.83	0.44	0.73	**0.55**	0.79	0.87
GPT-3 LLM	0-Shot	0.49	0.48	0.54	0.41	0.49	0.76	0.35
	FT Min	0.63	**0.87**	0.41	**0.77**	0.52	0.78	0.98
	FT Ext	**0.70**	**0.87**	**0.40**	**0.77**	**0.53**	0.78	0.97
Markov Model	BKT	0.67	0.87	0.40	0.78	0.53	0.78	0.99
Logistic Regression	Best-LR	**0.83**	**0.89**	**0.36**	**0.81**	0.65	0.83	0.95
DL: RNN	DKT	**0.83**	0.88	**0.36**	**0.81**	**0.68**	0.85	0.93
DL: Transformer	SAKT	0.82	0.88	**0.36**	**0.81**	0.67	0.85	0.92

Table 2: Results for ASSISTments 2009 dataset (Min = Minimal, Ext = Extended)

Family	Model	AUC	F1	RMSE	Acc	Bal Acc	Precision	Recall
Naive Baselines	Mean	0.50	**0.80**	0.47	0.66	0.50	0.66	1.00
	NaP	0.65	0.79	**0.46**	**0.68**	0.58	0.70	0.91
	NaP Skills	**0.68**	0.72	0.49	0.65	**0.63**	0.75	0.70
GPT-3 LLM	0-Shot	0.50	0.48	0.56	0.46	0.50	0.66	0.37
	FT Min	0.67	0.79	**0.45**	0.68	**0.60**	0.71	0.87
	FT Ext	**0.71**	**0.81**	0.46	**0.70**	0.56	0.69	0.98
Markov Model	BKT	0.71	0.80	0.44	0.71	0.62	0.73	0.88
Logistic Regression	Best-LR	**0.76**	**0.81**	**0.42**	**0.73**	**0.66**	0.75	0.88
DL: RNN	DKT	0.75	**0.81**	0.43	**0.73**	**0.66**	0.76	0.87
DL: Transformer	SAKT	0.72	0.78	0.45	0.70	0.65	0.75	0.81

Table 3: Results for ASSISTments 2017 dataset (Min = Minimal, Ext = Extended)

Family	Model	AUC	F1	RMSE	Acc	Bal Acc	Precision	Recall
Naive Baselines	Mean	0.50	0.00	**0.48**	0.63	0.50	0.00	0.00
	NaP	**0.60**	0.26	**0.48**	**0.65**	0.55	0.56	0.17
	NaP Skills	0.59	**0.35**	0.51	0.63	**0.56**	0.50	0.27
GPT-3 LLM	0-Shot	0.50	0.30	0.51	0.57	0.50	0.37	0.26
	FT Min	0.66	0.38	**0.46**	0.66	0.58	0.59	0.28
	FT Ext	**0.68**	**0.39**	**0.46**	**0.67**	**0.59**	0.63	0.28
Markov Model	BKT	0.63	0.27	0.47	0.65	0.55	0.60	0.17
Logistic Regression	Best-LR	0.70	0.48	0.45	0.69	0.63	0.62	0.39
DL: RNN	DKT	**0.77**	**0.58**	**0.42**	**0.72**	**0.68**	0.64	0.54
DL: Transformer	SAKT	0.70	0.51	0.46	0.69	0.63	0.60	0.43

The majority completion label in ASSISTments 2017, as opposed to the two other datasets, is "WRONG". As a result, the Mean baseline gives a constant output of 0, and thus, the number of *true positives* is zero, which in turn makes precision, recall, and F1 all to be zero.

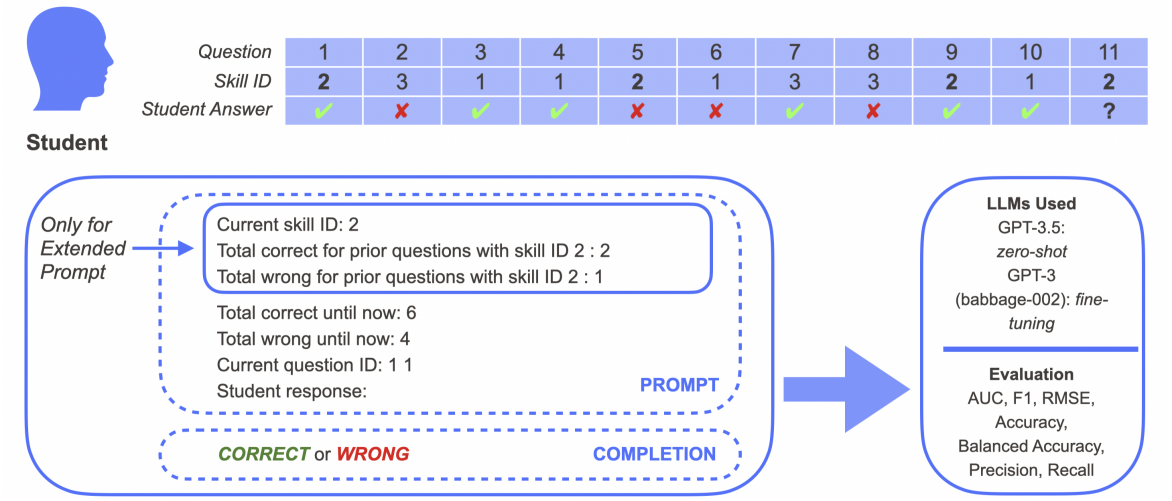
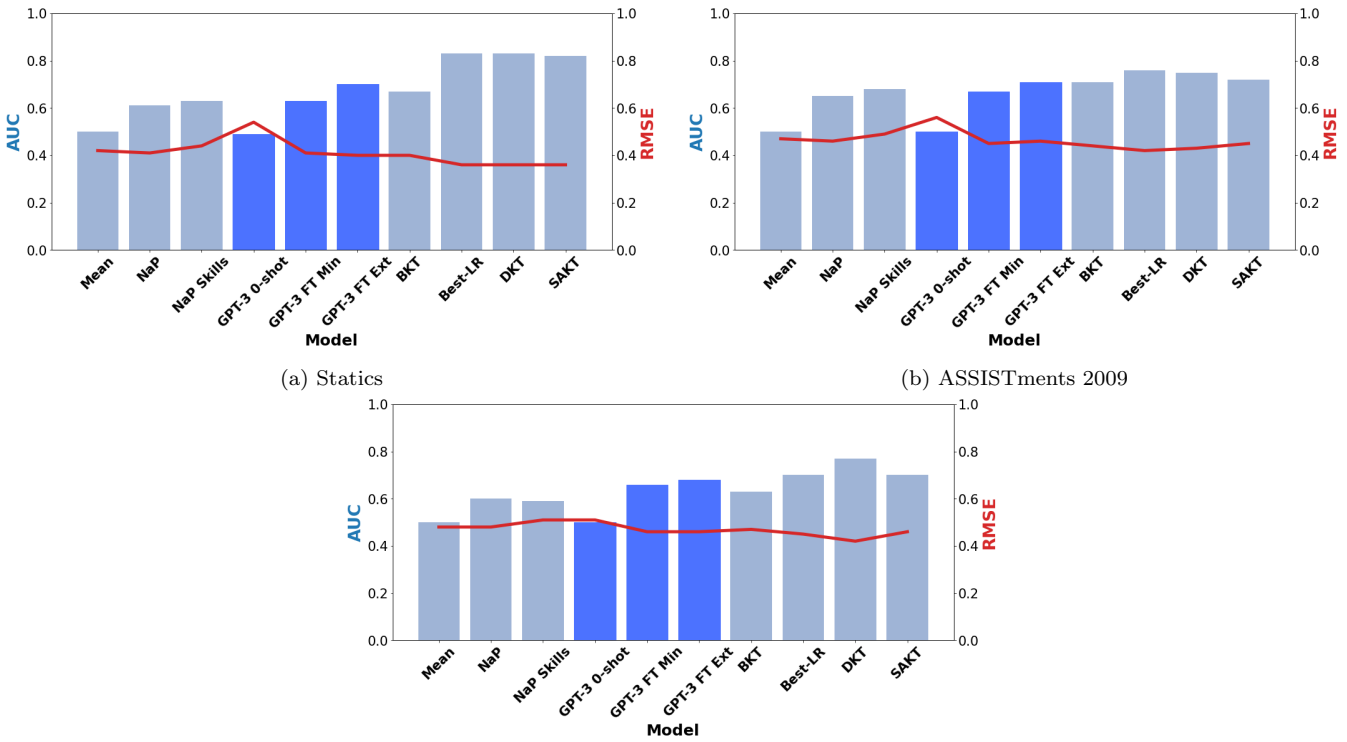


Figure 1: An overview of our prompting approach, following the feature sets used by [14], along with an example prompt based on a hypothetical sequence of student answers.



(a) Statics

(b) ASSISTments 2009

(c) ASSISTments 2017

Figure 2: Comparing the AUC and RMSE scores of different models for Statics (2a), ASSISTments 2009 (2b), and ASSISTments 2017 (2c). GPT-3 approaches are indicated in brighter blue.