Machine Learning with Applications 16 (2024) 100557

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Using ChatGPT to annotate a dataset: A case study in intelligent tutoring systems

Aleksandar Vujinović, Nikola Luburić, Jelena Slivka, Aleksandar Kovačević *

Faculty of Technical Sciences, University of Novi Sad, Serbia

ARTICLE INFO

Keywords:
ChatGPT
Prompt engineering
Intelligent tutoring systems
Dataset annotation

ABSTRACT

Large language models like ChatGPT can learn in-context (ICL) from examples. Studies showed that, due to ICL, ChatGPT achieves impressive performance in various natural language processing tasks. However, to the best of our knowledge, this is the first study that assesses ChatGPT's effectiveness in annotating a dataset for training instructor models in intelligent tutoring systems (ITSs). The task of an ITS instructor model is to automatically provide effective tutoring instruction given a student's state, mimicking human instructors. These models are typically implemented as hardcoded rules, requiring expertise, and limiting their ability to generalize and personalize instructions. These problems could be mitigated by utilizing machine learning (ML). However, developing ML models requires a large dataset of student states annotated by corresponding tutoring instructions. Using human experts to annotate such a dataset is expensive, time-consuming, and requires pedagogical expertise. Thus, this study explores ChatGPT's potential to act as a pedagogy expert annotator. Using prompt engineering, we created a list of instructions a tutor could recommend to a student. We manually filtered this list and instructed ChatGPT to select the appropriate instruction from the list for the given student's state. We manually analyzed ChatGPT's responses that could be considered incorrectly annotated. Our results indicate that using ChatGPT as an annotator is an effective alternative to human experts. The contributions of our work are (1) a novel dataset annotation methodology for the ITS, (2) a publicly available dataset of student states annotated with tutoring instructions, and (3) a list of possible tutoring instructions.

1. Introduction

Recently published large language models (LLMs) show impressive performance on various natural language processing (NLP) tasks. LLMs' ability to learn from a few input-label pairs (demonstrations) given some context is referred to as *in-context learning* (ICL) (Dong et al., 2023; Min et al., 2022). The ICL enables LLMs to apply instructions supplied via prompts to perform tasks they have not been trained for (Wei et al., 2022). The ability of LLMs to follow instructions has opened up a wide range of possibilities, including using LLMs for labeling datasets (Wang et al., 2021). Although utilizing LLMs for labeling datasets shows promising results (slightly below human performance) in various NLP problems (Wang et al., 2021), it still needs to be explored in different research areas. In this paper, we utilize an LLM, specifically ChatGPT, as an annotator to automatically annotate a dataset for intelligent tutoring systems (ITSs) .[1]

The goal of ITSs is to automate teaching using artificial intelligence. ITSs provide personalized instructions, mimicking the one-on-one interaction between a human tutor and a student (Abdelkefi & Kallel, 2017; Alkhatlan & Kalita, 2018; Almasri et al., 2019; Crow et al., 2018; Lo, 2023). In this interaction, an ITS acts as a tutor and provides instructions to a student, recommending advice such as prompting the student to review lecture materials or providing a hint for a task the student needs to solve. We refer to the provided tutoring instruction as an *action*. An ITS bases the provided action on relevant information about the student, such as grades or prior knowledge, which we refer to as a *student state*. A system that automatically provides a corresponding action given a student state is a crucial decision-making component of the ITS, that we refer to as the *instructor model*.

Developing an instructor model that can effectively act as a human expert tutor remains an open topic of research in ITSs. Most of the instructor models in the current ITSs are heuristic-based, i.e., following

* Corresponding author at: Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovića 6, 21 000 Novi Sad, Serbia.
E-mail address: kocha78@uns.ac.rs (A. Kovačević).

[1] The dataset is available at: https://drive.google.com/file/d/1_XRML8OdC1cbH22U4Lqt82PAfyq5jtnU/view?usp=share_link and it will be made publicly available upon publication of this paper.

https://doi.org/10.1016/j.mlwa.2024.100557
Received 27 May 2023; Received in revised form 23 April 2024; Accepted 3 May 2024
Available online 9 May 2024

hard-coded rules to determine which action to provide for a given student state (Mousavinasab et al., 2021). However, the heuristic-based approach has drawbacks such as requiring human expertise, limited generalization, and consequently inability to provide a personalized action. These limitations can be mitigated by utilizing supervised machine learning (ML) based instructor models.

Our research is aimed at developing an ML-based approach that can generalize from expert knowledge and can be fine-tuned to provide personalized action (tutoring instruction) for each student. We will incorporate expert knowledge into the model through behavior cloning, which involves training a supervised ML model to imitate and replicate expert actions based on given student states. To personalize the instructor model for each student, we will utilize reinforcement learning to adjust the behavior of this supervised ML model through trial and error.

Unlike heuristic-based systems, the application of behavior cloning and reinforcement learning reduces the reliance on experts, making it a time-efficient and cost-effective approach. Thus, in this paper, we focus on the behavior cloning aspect of the instructor model, i.e. the application of supervised ML within the context of ITS. To the best of our knowledge, ML has not been utilized in the ITS context due to a lack of available datasets for training. Annotating a dataset of student states with the corresponding action requires human experts with pedagogical and teaching skills, making the task costly and time-consuming. For those reasons, we aim to utilize an LLM as an annotator instead of human annotators, as it eliminates the need for human experts, and enables the development and benefits of ML-based instructor models, while being scalable, time-efficient, and cost-effective.

More specifically, in this paper, we utilized ChatGPT to annotate a dataset of student states with adequate tutoring instruction. To create such a dataset, we programmatically generated a dataset of student states and then used ChatGPT to label it. First, we prompted ChatGPT to create a list of actions that were used as labels. We carefully crafted the list of actions utilizing our pedagogical expertise and years of experience in university-level teaching. Then we instructed ChatGPT on how to act as a tutor by using a custom prompt comprising a list of instructions. We refer to this prompt as the *instructional prompt*. Finally, we used subsequent prompts to feed ChatGPT with the student states and collected the actions with which it responded. In this manner, we created a labeled dataset of student states and actions.

As expert tutors with extensive teaching experience at the university level, we performed a thorough manual analysis of generated actions in the annotated dataset. We found that ChatGPT can provide actions similar in quality to that of human experts. The key contributions of this paper are as follows:

- A publicly available dataset of student states annotated with tutoring advice, that can be utilized to train supervised ML models aiming to imitate human expert tutors. Additionally, these models can also be used as a starting point for fine-tuning via reinforcement learning, as was done for AlphaGo (Silver et al., 2016). To the best of our knowledge, there are no such publicly available datasets. Thus, making our dataset a much-needed resource for the development of ML-based ITS.
- A novel dataset annotation methodology within the ITS context. This approach does not rely on the need for expert annotators, making it a widely accessible and scalable solution for dataset annotation.
- Our methodology can be directly employed in the development of an ITS instructor model by harnessing ChatGPT as a tutoring expert, eliminating the need for classical rule-based systems and tutoring experts.
- A list of pedagogical actions, serving as a valuable foundation list for researchers aiming to develop an instructor model for an ITS. To the best of our knowledge, such a list is the first of its kind in the ITS field. The provided list of actions is adaptable and can be integrated into a wide number of courses, given that most of them follow the common pattern of studying lecture materials and completing practical tasks. By crafting a universal list of actions applicable to a wide range of courses that share a common pattern, we aim to provide a strong foundation for the development of ML-based ITSs.
- Our findings that demonstrate that utilizing ChatGPT for dataset annotation instead of human annotators is a cost-efficient and highly effective approach, within the context of ITS.

We present our findings in detail in the remainder of the paper. The rest of the paper is organized as follows. In Section 2, we present the background of our study. We present our methodology in Section 3. Section 4 presents the generated dataset and discusses our findings. Section 5 concludes the paper.

2. Background

2.1. In-context learning with large language models

Large language models such as ChatGPT are significantly advancing the field of AI, allowing researchers to explore using LLMs for various tasks. In-context learning is an active research area where researchers examine the possible applications and limits of instructing LLMs to perform tasks in different contexts (Brown et al., 2020; Dong et al., 2023).

In contrast to few-shot learning, ICL does not require parameter updates, making it time and cost-efficient (Dong et al., 2023). Early attempts of ICL with LLMs such as GPT-3 were limited by poor performance, leading most researchers to rely on few-shot learning. However, there is a growing interest in ICL with the development of ChatGPT (Tang et al., 2023). Recent studies explored the mechanism and efficacy of ICL. Min et al. (2022) found that the format of the prompt plays a crucial role. Their results show that a well-designed instructional prompt, coupled with demonstrations, can substantially enhance the performance of ICL. However, given that ChatGPT was released very recently, ICL is still just a promising methodological direction that remains to be tested in different research areas, such as ITS.

2.2. Instructional prompt

An instructional prompt is a set of instructions that are fed into an LLM. It serves as a guide to instruct an LLM to exhibit specific behavior. A well-designed instructional prompt is crucial for enabling LLMs to solve complex tasks (Dong et al., 2023; Min et al., 2022). Defining such a prompt involves creating clear and concise instructions that precisely describe the intended task. This process of designing prompts is referred to as *prompt engineering*.

2.3. Automatic dataset annotation using large language models

Although LLMs can successfully replace human experts in many tasks, only a few studies explored how LLMs can replace human annotators. Wang et al. (2021) studied whether the GPT-3 model could annotate data for natural language processing. Their results show that GTP-3 can be used for automatic dataset annotation at the human expert level. They claim that GPT-3 can reduce labeling costs by 50 % to 96 % in various natural language understanding and generation tasks. Most recently, Tang et al. (2023) demonstrated that ChatGPT could be used successfully for synthetic data generation and annotation in clinical text mining. They achieved significant performance improvements (in terms of F-measure) in named entity recognition and relation extraction tasks, up to 63.99 % and 83.59 %, respectively.

Leveraging LLMs over human annotators brings substantial benefits, automating the annotation process, eliminating the need for experts, providing scalability, and offering a cost-efficient method. To the best of our knowledge, there are currently no LLMs utilized for automatic dataset annotation and is yet to be studied in the context of intelligent tutoring systems.

2.4. Intelligent tutoring systems

Intelligent tutoring systems are computer-based instructional systems that mimic human tutors and provide personalized instruction at scale. ITSs rely on artificial intelligence to personalize and automate teaching (Abdelkefi & Kallel, 2017; Alkhatlan & Kalita, 2018; Almasri et al., 2019; Crow et al., 2018).

A critical component of an ITS is its instructor model (Mousavinasab et al., 2021) which plays a critical role as the decision-making component, in which an underlying AI model provides an action based on the given student state. Most well-established ITSs employ a heuristics-based approach to decision-making (Mousavinasab et al., 2021). This approach involves making decisions using a predefined set of rules manually defined by experts. Even though heuristic-based systems can perform well and provide interpretable decision-making, they have several significant drawbacks. Developing and maintaining heuristics requires human expertise, making it time-consuming and expensive. Manually defined heuristics are usually complex and typically rely on a specific set of attribute values (triggers), limiting their ability to generalize. In the context of ITSs, heuristic-based systems cannot automatically adapt through interactions with individual students, thus making them a poor methodological choice when the end goal is to provide a genuinely personalized student experience.

Machine learning is a logical and promising research direction that addresses all the drawbacks of heuristic-based approaches. Most ML models have an inherent ability to generalize and can also be adapted (re-trained) through time for the needs of individual students. However, the drawback of ML models is that they typically require large amounts of training data, i.e., annotated examples. In the context of ITSs, training data comprises interactions between students and a human expert tutor. The student is described in each interaction by a set of attribute-value pairs (a state), while the tutor provides an action (annotates the state). Creating a large set of student-tutor interactions is a costly and time-consuming process that using LLMs such as ChatGPT can mitigate. To the best of our knowledge, no research has been conducted on applying LLMs for automatic data annotation in the context of ITSs.

3. Method

Fig. 1 gives an overview of our methodology. As a first step, we prompted ChatGPT to create a list of potential actions a tutor would give to a student. We then used our tutoring expertise (please see Appendix A) to filter out inadequate actions manually. The collected actions are then used as annotation labels in the annotation step. In the annotation step, we instructed ChatGPT with a custom prompt to label the student states provided to it as input. It was instructed to respond with exactly one action from the list generated in the previous step. Each of our methodological steps is presented in detail in the remaining part of this section.

3.1. A list of actions

To generate an initial list of actions, we defined an instructional prompt, telling ChatGPT to assume the role of a tutor (please see Appendix B for the complete prompt). We then posed different queries as a student seeking help and collected the responses. An initial analysis revealed that the resulting list of actions could not be used as is. It contained responses that were:

- Not appropriate actions in the context of a tutoring session with an ITS (e.g., ***Eat a healthy meal, get enough sleep, and engage in activities that make you feel good*** or ***Reflect on the cause of frustration***), and
- lexical variants of the same action (e.g., ***Break down your studying into smaller chunks of time and take breaks to avoid burnout*** and ***Take a break***).

Our expertise in pedagogy, coupled with years of experience teaching at the university level, enabled us to manually filter out the actions that could not be used as labels for our dataset. The initial list comprised 27 actions, while the filtered list included 17 unique actions (Table 1). The actions from the filtered list of actions were further utilized as labels in the dataset annotation step (Section 3.2).

3.2. Annotating the dataset

As our methodology relies on in-context learning, we defined an instructional prompt (please see Appendix B for the detailed prompt) to set the context for ChatGPT. Then, we prompted ChatGPT with student states. The resulting responses (actions) were used to annotate the dataset of student states.

3.2.1. Instructional prompt

ChatGPT, while rich in knowledge, may lack specific information and knowledge in certain contexts. To address this, we utilized ICL by providing an instructional prompt, thereby enhancing ChatGPT's

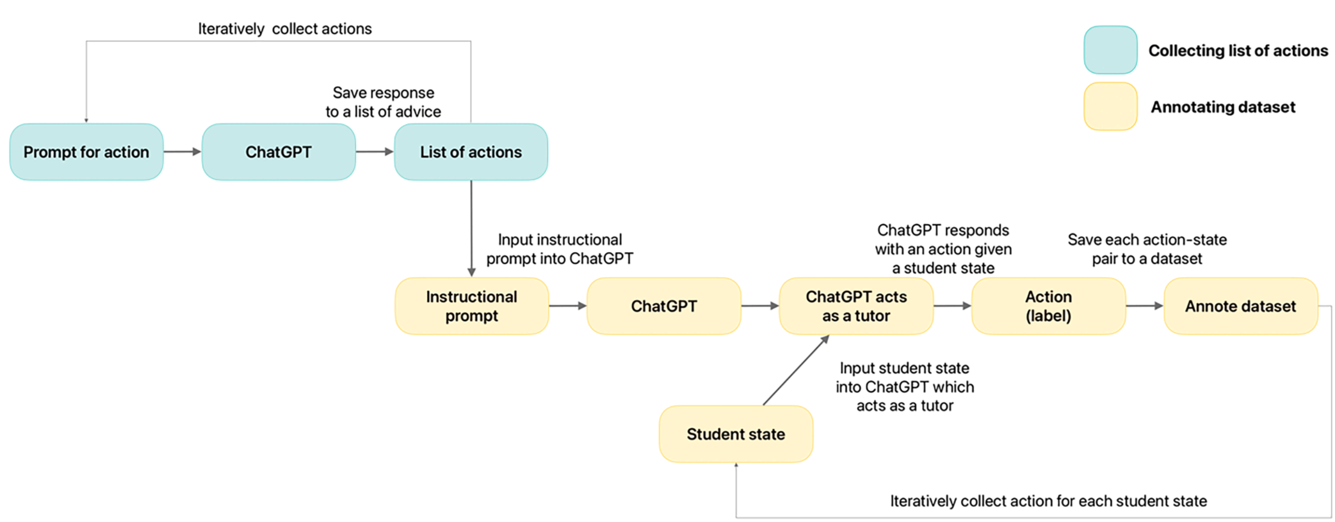

Fig. 1. An overview of the workflow of generating a list of actions and annotating the dataset.

Table 1
The original, unmodified list of actions provided by ChatGPT. Out of the 27 actions, only 17 were utilized. The utilized actions are highlighted, while the rest are not highlighted.

Actions
Apply what you've learned to new problems
Approach a task from a different angle and think creatively
Ask for a hint
Break down your studying into smaller chunks of time and take breaks to avoid burnout
Break the task down into smaller sub-tasks, and tackle them one at a time
Create a concept map or diagram to visualize
Explore related topics to expand knowledge when motivated
Find connections between the material you're learning and your own interests
Help others. Teach the material to someone else
Keep in mind that it's normal to struggle sometimes, and don't let it discourage you from continuing to learn
Look at examples or previous solutions of similar problems to get an idea
Make sure to get enough sleep and exercise
Move on to a new lecture
Practice regularly, even if you don't feel like you're making progress
Recap the lecture when finished
Recommend new lecture based on previous knowledge
Reflect on past successes and remind yourself that you have the skills and knowledge to complete the task
Reflect on the cause of frustration
Review lecture materials
Solve similar problems to solidify understanding
Summarize key concepts in your own words
Switching to a different problem or task to give your brain a change of pace
Take a break
Take practice quizzes to test your understanding of the material
Think about why it took more time to review lecture than expected
Try solving easier tasks if previous ones were too challenging
Try solving more challenging tasks if previous ones were too easy

knowledge and context. We designed a custom instructional prompt by specifying a set of instructions on how ChatGPT should act as a tutor. We instruct ChatGPT to provide an action that maximizes the student's engagement during the learning process.

In the instructional prompt, we defined a high-level description of a learning process with the tutor, thereby setting the context for the conversation (action generation). Next, we outlined the roles in the system (tutor and student) and the objective to generate the appropriate action. Then we described the conversation process in more detail by providing it with the list of steps it must follow. It's necessary to provide it with the list of actions (see Table 1) as it must know which action to generate. Additionally, to enhance contextual understanding, we incorporated three examples of student-tutor conversations. Lastly, we defined a set of explicit rules that ChatGPT must obey when generating the responses.

Through iterative experimentation with various prompts and prompt components (i.e., prompt description, set of instructions, examples provided, order of instructions), we empirically identified the one that yielded favorable results. We note that we experimentally found that excluding or altering the order of the components in the instructional prompt led ChatGPT to generate inadequate responses (actions), thus emphasizing the importance of specifying the context and clear instructions. The full prompt can be found in Appendix B.

3.2.2. Student state preprocessing

A student state comprises relevant information about a student, which an ITS uses to provide an appropriate action. It is dynamically generated and continually updated based on the student's previous interactions with the ITS, representing their progress and performance in the learning process. In our case, the state is defined by a set of attributes presented in Appendix C in Table C.2. We selected this specific set of attributes based on our previous work in the field of ITSs and our tutoring expertise (Luburić et al., 2022).

After conducting initial experimental tests, we determined that ChatGPT performs better when values of continuous attributes are discretized. For example, values of the attribute *percentage of completed tasks* were discretized from a continuous interval [0–100] to a set of values: 0, 20, 50, 80, 100. Similarly, we discretized the attributes: *typical studying time, previous lecture completion time, current lecture materials completion time,* and *current lecture assignments time*.

D'Mello and Graesser. (2012) determined that students experience four affective states during the learning process. We found that ChatGPT responded to three of these affective states during the initial experiments, so only these three were included in the student's state attribute values (Table C.2).

After preprocessing, the final state space included 38,880 unique combinations of attribute values (from Table C.2). Managing such a vast state space with the heuristic-based approach would be time-exhaustive and demand substantial effort from human experts. Notably, there are currently no ITSs that operate within such a large state space, further emphasizing the challenges associated with the conventional heuristic-based approach. For efficient action generation within such state space, it is crucial to utilize an ML-based approach, which is dependent on the much-needed annotated dataset that we provide.

Note that given the time constraints,[2] we could not label the whole state space, so we randomly sampled 260 states Thus, our dataset comprises 260 instances of student states and corresponding actions generated by ChatGPT.

3.2.3. Input prompt format

There are several ways to represent a state as an input to ChatGPT. At first, we experimented with structured representations, such as JSON. Even though ChatGPT could parse the structured formats, the initial results were not promising. Thus, we converted each student state into a purely textual representation. We do so by mapping each attribute value pair to a separate natural language sentence. For instance, we generated the following sentence from the attribute *prior knowledge* and the value *good*: "*I have **good** prior knowledge of the previous lectures.*". So, the input text to ChatGPT consisted of a few sentences in the English language. The following is an example of the student state:

[2] Each state had to be submitted as a separate query through the ChatGPT web interface available at: chat.openai.com.

Student: "I have **good** prior knowledge of the previous lectures. I completed previous lecture in **expected** time. I read lecture materials in **expected** time. I've completed **100 %** of all tasks in the current lecture and it took me more time than expected. I'm feeling **motivated**."

We intentionally omitted the use of definite and indefinite articles, aiming to decrease the token count while preserving meaning, thereby lowering the annotation cost. In the previous student state example, this approach led to a 9 % reduction in tokens. We note that the amount of reduction depends on the LLM itself.

3.2.4. Response format

In the instructional prompt, we defined a specific output format for ChatGPT that includes the action itself, followed by an explanation of why that action is the most appropriate. There are two benefits to including an explanation. Firstly, it helps with post-analysis to determine why some states were mislabeled. Secondly, an LLM can improve its accuracy by explaining the reasoning behind its output, which is referred to as the *chain of thought* (Wei et al., 2023). Simply instructing an LLM to provide its chain of thought has been shown to reduce the amount of false or incorrect information generated by the model, which is known as *hallucination* (Dong et al., 2023; Wei et al., 2023). The following is an action and explanation returned by ChatGPT for the state from the example in the previous sub-section:

Tutor: "**Explore related topics to expand knowledge when motivated.** When you are motivated, it is a great opportunity to take advantage of your enthusiasm and drive to learn even more. By exploring related topics, you will be able to expand your knowledge base, further deepen your understanding of the material, and keep your motivation levels high."

4. Results and discussion

In this section, we present the results of our study and discuss the performance of ChatGPT in the two tasks we asked it to perform. The first task was to generate a list of potential actions that a tutor would provide to a student, and the second was a more complex task to automatically annotate student states. The first task resulted in a list of 17 unique actions, while the second task resulted in a dataset of 260 student states, each annotated with its corresponding action from the list of actions. The annotated dataset consists of 11 attributes (please see Appendix C) and 17 labels (please see Table 1). The number of instances per class is presented in Section 4.2 (Label distribution). We provide more detailed dataset characteristics in Appendix D.

4.1. Analysis of the list of actions

As mentioned in the Methodology section, ChatGPT relatively successfully generated the list of potential actions. Some manual filtering was required, but expanding the list (i.e., manually adding an action ChatGPT missed) was unnecessary. Based on our analysis, the final generated list of actions comprised 17 unique actions that can be categorized into three groups, given their intended purpose:

- The first group focuses on enhancing the student's critical thinking skills, such as: ***Approach a task from a different angle and think creatively***.
- The second group contains motivational actions, such as: ***Reflect on past successes and remind yourself that you have the skills and knowledge to complete the task.***
- The third and largest group contains practical actions, such as: ***Ask for a hint***, or ***Move onto a new lecture***.

We note that certain courses might require specific actions, not included in the original list of actions or annotated dataset. This can be addressed by training the ML instructor model with the annotated dataset, followed by fine-tuning the model with specific actions tailored to the unique requirements of courses that go beyond the provided list. The pre-trained instructor model can seamlessly be fine-tuned to incorporate novel actions tailored to the unique requirements of a specific course.

4.2. Label distribution

Fig. 2 presents a horizontal bar plot of actions sorted by their frequency. We see a great disproportion in frequencies in favor of the ***Take a break*** action compared to other actions. We suspect this is due to how ChatGPT was developed, i.e., its desired behavior to avoid giving harmful output. We further confirmed this by the explanation part of the response:

Tutor: …"The advice take a break is a safe-bet advice, because in most situations, taking a break and stepping away from a task can give you a wide perspective and freshen your mind."

Although the action ***Take a break*** is not erroneous, we did not consider it useful at all times. Given that OpenAI released a superior GPT-4 model (OpenAI, 2023) at the time of our analysis, we prompted GPT-4 with the states that had the ***Take a break*** label. The GPT-4 model provided actions different from ***Take a break*** that were better suited for the given states and consistent with our expert opinion.

The subsequent most frequently provided actions within the collected dataset were ***Solve similar problems to solidify understanding*** and ***Ask for a hint***. The frequency of provided actions was expected, as it is reasonable to anticipate that human tutors would usually provide similar actions.

Some of the actions from Table C.2 were quite infrequent, such as ***Practice regularly, even if you don't feel like you're making progress*** and ***Summarize key concepts in your own words***. The action ***Find connections between the material you're learning and your own interests*** was not outputted even once. Based on our manual analysis, the infrequent actions were provided to motivate students and given when the other, more specific actions in the list, were not applicable. While this interpretation requires further investigation, it offers one possible explanation for the observed frequency of these types of actions.

The collected label distribution from the annotated dataset might not be ideal for each student and each course. While the instructor model may generate reasonable actions, they might not be optimal. Such a model's behavior can be adjusted by fine-tuning the ML-based instructor model, essentially changing the label distribution. Consider the following example where there might be a specific course that introduces novel concepts to students, requiring more detailed lecture reading than the majority is used to. Consequently, students might experience difficulties completing assignments (can be tracked with the *Percentage of successfully completed assignments for current lecture* attribute). In this scenario, the ML-based instructor model might overly suggest the action ***Take a break*** due to its training data and the difficulties that students face. Such model's behavior can be fine-tuned to reduce the probability of yielding this action and increase the probability of a more suitable action, such as *Review lecture materials*. This adaptability is a unique feature of the ML-based approach, practically not achievable with classical heuristic-based systems that require creating new rules and expert involvement.

4.3. Response analysis

Ideally, to evaluate the dataset annotation results (responses generated by ChatGPT), we would either (1) compare the results with another instructor model or (2) employ multiple experts for the manual dataset annotation and compare their responses with ChatGPT's. However, both options aren't feasible for a few reasons we discuss in the following.

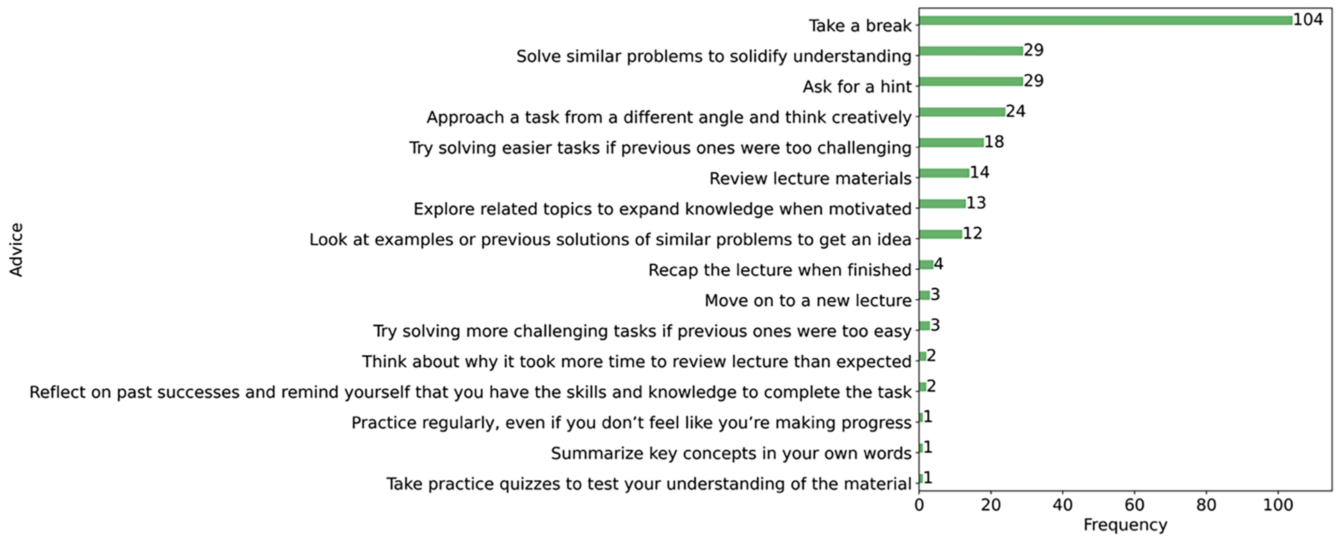

Fig. 2. A horizontal bar plot of actions (tutoring advice) sorted by frequency.

Instructor models are typically implemented as hard-coded rules. Rule-based systems have inheritably different behavior, generating actions based on the observed student state using predefined rules. Consequently, they operate within small state spaces. On the other hand, we utilized a large state space of 38,880 student states. Rule-based systems do not operate within such a large state space, which is suitable for ML-based instructor models, and to the best of our knowledge, there aren't any such ML-based or rule-based instructor models. Thus, a comparison with the existing instructor model isn't feasible.

Moreover, since we utilized a large state space, creating a rule-based system to match this complexity would demand significant expertise, coding, and time resources, and would be a great challenge by itself. Thus, developing a rule-based system from the ground up isn't a viable option.

Another possible approach would be to employ multiple experts for the manual dataset annotation and subsequently compare ChatGPT's responses with those of human annotators. However, this approach is highly inefficient and time-consuming and thus isn't feasible due to our limited resources.

Thus, for the previous reasons, we had to manually analyze all the responses generated by ChatGPT. Most of the responses were correct ~95 %, i.e., they correspond to the action that an experienced human tutor would provide. Leveraging our pedagogical expertise and years of university-level teaching experience, we conducted the manual analysis and corrected the instances that had been incorrectly annotated by ChatGPT. In the following Section (4.3.1) we offer insights into the manual annotation process by providing a concrete example of a student-tutor conversation and a corresponding evaluation process. We emphasize that this manual evaluation process is extensive and requires expertise. We note that in the future, we plan to conduct a case study developing and applying the ML-based instructor model to university-level courses.

Our manual analysis revealed that some generated responses could not be used directly as labels in the resulting dataset. These responses could be categorized into the following three categories: incorrect label (Section 4.3.1), incorrect output format (Section 4.3.2), and unexpected action (Section 4.3.3). In the following sections, we present our results regarding response analysis in detail. Finally, we compare results obtained from different versions of ChatGPT (Section 4.3.4).

4.3.1. Incorrect label

We consider that the *incorrectly labeled* states are those in which the action generated by ChatGPT does not correspond to the action one of our expert tutors (please see Appendix A) would have given. More formally, the *incorrect label* is when a tutoring expert wouldn't provide the same action as ChatGPT's, given a student state:

$$label = \begin{cases} correct, & expert's\ action(student\ state) = ChatGPT's\ action(student\ state) \\ incorrect, & expert's\ action(student\ state) \neq ChatGPT's\ action(student\ state) \end{cases}$$

There were ~5 % of mislabeled instances in the dataset. In the following, we provide an example of a student-tutor conversation that we consider *incorrectly labeled*. Furthermore, we provide a concrete example of the manual evaluation process. We show the decision-making process behind evaluating whether ChatGPT's generated action was correct. The following is a student-tutor conversation that resulted in an *incorrectly labeled* state:

> Student: "I have average prior knowledge of the previous lectures. I usually learn during the night, but now it's daytime. I completed previous lecture in expected time. I read lecture materials in more time than expected. I've completed 0 % of all tasks in the current lecture. I've already reviewed lecture materials. I've already taken a break. I can't solve the next task."
>
> Tutor: "**Solve similar problems to solidify understanding**."

In this example, the student has average prior knowledge and exhibits evident difficulties within the current lecture. The student's struggle is evident as the student takes longer than expected to read the lecture materials and has yet to complete any assigned tasks. While the action ***Solve similar problems to solidify understanding*** is generally an excellent strategy to reinforce knowledge, it is ineffective in this context. We consider this action *incorrect* since the student is struggling with the current task and cannot overcome the difficulty level for the current task. Encouraging the student to tackle tasks of similar difficulty levels can lead to boredom, frustration, and, finally, disengagement. Instead, providing the action to ***Ask for a hint*** is more beneficial in this context, as it provides gentle guidance and support, aiding in overcoming the challenge. Such action keeps the student engaged while maintaining motivation.

We conducted experiments on a small subset of states ~5 %, to test the consistency of ChatGPT in providing an action for the same student state multiple times. For some states, it showed inconsistency by returning different actions as output when prompted multiple times with the same prompt. This behavior was expected due to the absence of temperature adjustment on the web interface (although it is available in the API). To address the inconsistency, we considered only the first action generated. However, it is essential to acknowledge that alternative action variations could potentially be correct labels.

Note that we don't aim to develop the ITS for a specific education level, but rather for courses that share a common pattern of studying lecture materials and completing practical tasks. However, our expertise lies in higher-level education, and we considered that level when annotating incorrectly labeled instances (i.e., we don't consider the primary school environment).

4.3.2. Incorrect output format

Amongst the collected responses, there were cases in which ChatGPT did not follow the output format specified in the instructional prompt (exactly one action followed by an explanation afterward). We estimate that at least 5 % of responses had incorrect output format. In the following example, ChatGPT responded with multiple actions:

> Student: "I have bad prior knowledge of the previous lectures. I usually learn during the night, but now it's daytime. I completed previous lecture in less time than expected. I read lecture materials in expected time. I've completed 50 % of all tasks in the current lecture and it took me less time than expected. I'm feeling motivated. I've already reviewed lecture materials. I've already taken a break. I can't solve the next task."

> Tutor: "The best advice for you in this situation is to **ask for a hint** or **approach the task from a different angle and think creatively**. It's normal to get stuck when working on difficult tasks, but trying to solve the problem from a new perspective can often help. If you're still struggling, asking for a hint from a peer, teacher, or online resource can give you the boost you need to keep moving forward."

In the given example, ChatGPT deviated from the specified output format and provided contradictory actions (***Ask for a hint*** and ***Approach the task from a different angle and think creatively***), creating ambiguity regarding which action to follow. In such a case, we annotated a state with a first provided action. Besides ignoring the output format, ChatGPT sometimes responded only with the label, excluding the explanation, e.g.:

> Student: "I have good prior knowledge of the previous lectures. I usually learn during the night, but now it's daytime. I completed previous lecture in expected time. I read lecture materials in more time than expected. I've completed 20 % of all tasks in the current lecture and it took me more time than expected. I'm feeling motivated. I've already taken a break. I've already tried to solve similar problems. I can't solve the next task."

> Tutor: "Approach a task from a different angle and think creatively".

4.3.3. Unexpected action

In some cases (at least 5 %), ChatGPT responded with an action not specified in the instructional prompt. In such a situation we repeated the prompt until ChatGPT responded with the expected action, thus eliminating the unexpected actions in the final provided dataset. Consider the following state which yielded unexpected action:

> Student: "I have average prior knowledge of the previous lectures. I usually learn during the day, but now it's night-time. I completed the previous lecture in less time than expected. I read lecture materials in more time than expected. I've completed 50 % of all tasks in the current lecture, and it took me less time than expected. I'm feeling motivated. I've already reviewed the lecture materials. I can't solve the next task."

> Tutor: "It sounds like you're making good progress, despite some challenges. Based on your information, it appears that you are motivated to learn and have been making an effort to review the material and attempt to solve problems. However, you are currently facing difficulty with a task, which can be frustrating. One suggestion would be to **try to break down the task into smaller parts and tackle each part one at a time**. You can also consider seeking help from classmates or reaching out to your teacher for additional guidance."

In this example, ChatGPT provided the unexpected action to ***Try to break down the task into smaller parts and tackle each part one at a time***, which is not a part of the predefined list of actions and is therefore unexpected. This example illustrates the stochastic nature of LLMs, their ability to follow instructions, and their sensitivity to the instructional prompt. The field of prompt engineering is an emerging research field, and for many applications, designing effective instructional prompts remains an open research direction. We note that we repeated this prompt until ChatGPT responded with the expected action, and thus such unexpected actions aren't found in the dataset we provide.

4.3.4. Comparison with the GPT-4 model

The results presented in this paper were collected by the GPT-3.5 model (in March 2023), as it was the only available version at the time of writing our initial preprint. Since then, we were able to gain access to the GPT-4 model. In order to compare the two models (GPT-3.5 and GPT-4), we prompted GPT-4 (in April 2024) with the same prompts as GPT-3.5. We compared the annotation results generated by GPT-3.5 and GPT-4, focusing on a subset of labels (returned by GPT 3.5) that we manually identified as incorrect (please see Section 4.3). In the remainder of this section, we present the results of the comparison.

We examined a subset of GPT-3.5 labels that could not be used directly as labels. Although the issue of incorrect labels (please see Section 4.3.1) is still present, the GPT-4 version mitigates it to some extent. More specifically the GPT-4 model provides more appropriate responses given complex student states as input. Consider the following example where the response from the GPT-3.5 was to ***Take a break***, but the GPT-4 model responded with ***Ask for a hint***:

> Student: "I have good prior knowledge of the previous lectures. I usually learn during the day, but now it's night-time. I completed previous lecture in more time than expected. I read lecture materials in more time than expected. I've completed 50 % of all tasks in the current lecture and it took me less time than expected. I'm feeling frustrated. I've already reviewed lecture materials. I've already taken a break. I've already tried to solve similar problems. I can't solve the next task."

In this example, the student is frustrated which is why GPT-3.5 responded with ***Take a break***. However, good prior knowledge and the review of lecture materials suggest that the student possess knowledge and that he or she is struggling with a specific problem, indicating a need for a hint. Hence, GPT-4 recommended to ***Ask for a hint***, a strategy that guides students towards solving the task and consequently mitigating the frustration. Thus, we'd consider the response from GPT-4 as the more appropriate one.

Compared to the GPT-3.5, GPT-4 fully eliminates the issue of the incorrect output format (please see Section 4.3.2). Each of the responses was in the specified output format and accompanied by a corresponding explanation. As for the unexpected action issue (please see Section

4.3.3), we still observe the same number of unexpected actions from both GPT-3.5 and GPT-4. We note that we compared only a subset of annotated states between the models, as manual comparison is extremely resource and time extensive.

4.4. Limitations of our study

A major limiting factor for our study was the lack of access to the ChatGPT API, which was unavailable when we collected the data due to a long waiting list. As a result, we could only use ChatGPT through the web interface. Therefore, we could only annotate less than 0.7 % of state space (260 different states out of 38,880) in the provided dataset. A larger sample might have yielded different results.

While annotating the dataset, we were also limited by the number of requests per hour and the maximum number of tokens per query. These constraints applied to both the API and web interface, with the web interface having a higher limit than the API.

We also faced limitations regarding the length of the conversation context, which consequently affected the instructional prompt's length. In the case of LLMs, the context length refers to the number of tokens used as input to estimate the probability distribution of the next word during response generation (Khandelwal et al., 2018). The conversation context length directly impacted the instructional prompt, as it limited the amount of information we could include. Due to this constraint, we included only three demonstrating examples within the instructional prompt for ChatGPT. Using more demonstrations would most likely have increased the quality of its responses, but we were limited by the number of tokens and context length.

It also turned out that ChatGPT struggled to understate the difference in magnitude between continuous values, requiring us to discretize values for some attributes. More advanced LLMs, such as GPT-4 or PALM 2 (Anil et al., 2023), may be capable of handling continuous values enabling us to provide the model with a more precise representation of the student state. We acknowledge the possibility of better models emerging in the future that could enhance annotation accuracy. However, we believe the label distribution shift will be marginal, given the already robust annotation results.

Finally, our expertise was utilized to determine the final list of actions and analyze the resulting responses, which can be considered a limiting factor. However, we stress that we have an extensive pedagogical experience in tutoring (please see Appendix A).

5. Conclusion

In this paper, we conducted a case study on the efficacy of using ChatGPT for annotating a dataset for training an instructor model for the ITS. Using ChatGPT, we automatically annotated the dataset of student states with the pedagogical action a human tutor would recommend for maximizing the student's engagement. To create this dataset, we first utilized prompt engineering techniques to create a list of actions a tutor could recommend to a student. We manually filtered the obtained list of actions and then instructed ChatGPT to return the appropriate action from the list for the given student's state. We manually analyzed and categorized ChatGPT's responses that could be considered incorrect labels in the resulting dataset.

Our results indicate that utilizing ChatGPT as an annotator is an effective and efficient alternative to relying on human experts, as it can significantly reduce both time and cost of developing a large dataset of adequate pedagogical actions. The main contribution of our work is a publicly available dataset of student states and corresponding tutoring actions, that can be used to train a supervised ML model which would imitate human expert tutors. Moreover, we present a novel method for dataset annotation in the context of ITS, which is relevant to researchers looking to develop ML-supported instructors that maximize student engagement. Additionally, we created a list of actions that can be used to improve the effectiveness of ITS. The action list is relevant to the researchers looking to develop effective instructor models for intelligent tutoring systems.

In the future, we aim to collect a larger dataset to enable the personalization of ITSs. We hope this study inspires further research on the use of large LLMs in providing personalized tutoring instructions and helps to enhance the development and use of ITS in education.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve readability and language. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

CRediT authorship contribution statement

Aleksandar Vujinović: Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Nikola Luburić:** Methodology, Writing – review & editing. **Jelena Slivka:** Writing – review & editing. **Aleksandar Kovačević:** Conceptualization, Methodology, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made publicly available upon publication of this paper

Acknowledgments

This research has been supported by the Ministry of Science, Technological Development and Innovation (Contract No. 451-03-65/2024-03/200156), and by the Faculty of Technical Sciences, University of Novi Sad through project "Scientific and Artistic Research Work of Researchers in Teaching and Associate Positions at the Faculty of Technical Sciences, University of Novi Sad" (No. 01-3394/1), and by the Science Fund of the Republic of Serbia, grant no 6521051, AI-Clean CaDET.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.mlwa.2024.100557.

Appendix A. Detailed description of our expert tutors

All of the authors were employed to create the final list of actions, utilizing their expertise to curate and finalize the list of actions. Aleksandar Vujinović is a Ph.D. student with two years of higher education teaching experience and is currently holding a teaching assistant position. Nikola

Luburić is an assistant professor with 9 years of higher education teaching experience. Jelena Slivka has 12 years of higher education teaching experience and is currently holding an associate professor position. Aleksandar Kovačević is a full professor with 16 years of higher education teaching experience.

Appendix B. Instructional prompts

In this section, we present our instructional prompts. In Prompt B.1, we show what prompt we used to create the initial list of actions. In this prompt we first set the context for ChatGPT, describing the process and roles in the system. When the context is set, we prompt ChatGPT with different states, as described in our methodology (Section 3). Once we created the final list of actions, we instructed ChatGPT with another custom prompt (Prompt B.2) to annotate the dataset by labeling student states (each state was submitted as a separate prompt) with a single action from the list of actions.

Prompt B.1
Instructional prompt used to create a list of actions

Description:
You are the Tutor Advice Generator, a tool for generating advice to a student given a student's current progress while learning.
The input is a student's current progress and affective state. The tutor outputs advice to a student. Given advice should maximize student's learning outcome. The goal of the tutor is to give advice (personalized feedback) to each student regarding the progress so that the student is engaged in the process of learning. A student also experiences emotions while learning. The tutor must consider student's emotions or affective states before providing advice.
While learning, a student must follow the rules:
The student learning process is a process where the student is learning about some topic.
The student reads the lecture materials and watches videos on the same topic.
Then, the student starts solving tasks given the same topic.
While solving tasks the student can go back to review lecture material and can quit learning. Quitting is bad and the goal of the tutor is to give advice so that the student doesn't quit but is engaged in a process of learning.
While solving tasks, students can be engaged in learning and that's great! On the other hand, students can feel negative emotions such as being bored, confused, and frustrated. Feeling those emotions for a long time leads to a lack of motivation which is bad.
Tutor's role: Act as the Advice Generator to maximize student's learning and engagement for learning.
Student's role: Provide information about student's learning progress within a specific topic.
Objective: Generate advice to a student (what to do next) given a student's progress.
Program Mechanics: The student inputs the progress made during learning and the tutor outputs advice which keeps the student engaged in learning.
Sequential Actions:
1. Ask user about student's state
2. Analyze student's current progress and generate advice
3. Output the prompt to the user.
Steps:
1. Ask the user questions about the student's progress
2. Wait for the student's response
3. Use the student input to generate advice which will maximize student's learning outcome
4. Output one advice
Rules:
1. Output one advice when asked.
2. Must follow all rules
Start your "act".

Prompt B.2
Instructional prompt used to annotate the dataset

Description of the Tutor:
You are the Tutor Adviser, an AI tool that provides best advice to students. The Tutor (tool) considers information about students (such as student's current progress), and then provides the appropriate advice to maximize learning outcome. The goal of the Tutor is to give the best advice to a student so that a student is motivated and engaged in a learning process. The Tutor helps student learn and feel motivated. The student must follow certain rules such as reading lecture materials, and the Tutor must also follow rules such as providing only one advice when asked.
Description of learning process with the Tutor:
In the process of learning, a student is learning about any subject (e.g., machine learning, software engineering, statistics). Each subject consists of multiple lectures. Each lecture consists of: 1. lecture materials and 2. tasks. A student reads lecture materials and then begins to solve tasks regarding the same lecture. Once a student solves each task in the lecture, that lecture is considered successfully completed. While trying to solve tasks, a student is allowed to review the lecture materials multiple times. While learning, a student can ask the Tutor for help to solve the task or understand a lecture. A student can quit learning, but that's bad and the goal of the Tutor is to give the best advice so that a student doesn't quit but is engaged in a learning process.
Tutor's role: Act as the Tutor to maximize student's learning and engagement.
Student's role: Provides information about their current state, which is specific to the current progress point in time.
Objective: Generate advice to a student what to do next given a student's progress.
Steps that the Tutor must follow:
1. Ask a student about the progress.

(*continued on next page*)

Prompt B.2 (*continued*)

2. Wait for the student's response.
3. Use a student's input to output the best advice.
4. Output the appropriate advice to the student and explain why is that the appropriate advice (if it's possible to explain).

List of advice that the Tutor gives:
- Review lecture materials,
- Ask for a hint,
- Approach a task from a different angle and think creatively,
- Take a break,
- Solve similar problems to solidify understanding,
- Look at examples or previous solutions of similar problems to get an idea,
- Move on to a new lecture,
- Recap the lecture when finished,
- Explore related topics to expand knowledge when motivated,
- Try solving more challenging tasks if previous ones were too easy,
- Try solving easier tasks if previous ones were too challenging,
- Think about why it took more time to review lecture than expected,
- Reflect on past successes and remind yourself that you have the skills and knowledge to complete the task,
- Summarize key concepts in your own words,
- Find connections between the material you're learning and your own interests,
- Practice regularly, even if you don't feel like you're making progress,
- Take practice quizzes to test your understanding of the material.

Examples of student-tutor conversation:

Example 1:
- Student: I have good prior knowledge of the previous lectures. I completed previous lecture in more time than expected. I read materials on the current lecture in less time than expected. I've completed 100 % of all tasks in the current lecture and it took me less time than expected. I'm feeling motivated.
- Tutor: Move on to a new lecture.

Example 2:
- Student: I have good prior knowledge of the previous lectures. I completed previous lecture in less time than expected. I read materials on the current lecture in less time than expected. I've completed 10 % of all tasks in the current lecture and it took me more time than expected. I can't solve the next task.
- Tutor: Review lecture materials

Example 3:
- Student: I have bad prior knowledge of the previous lectures. I completed previous lecture in more time than expected. I read materials on the current lecture in about expected time. I've completed 20 % of all tasks in the current lecture and it took me much more time than expected.
- Tutor: Try solving easier tasks if previous ones were too challenging

Rules that the Tutor must follow:
1. Output only one advice when given student state(input).
2. Output only one advice from the "List of advice that the Tutor gives". Do not output advice that isn't in this list.
3. Only consider current input when giving advice, and not previous inputs. Each received input, student's progress, is independent of previous inputs.
4. Output one advice and explain why is that best advice. When you output advice, output the advice in the first sentence, then output an explanation.

Start your ''act'' as the Tutor.

Appendix C. Student state

In the following table (Table C.2) we present attributes, carefully selected by utilizing our tutoring expertise, used to define each student. Each state is defined by 11 attributes, with each attribute corresponding to a set of predefined values, shown in the *Possible values* column. Moreover, we provide a comprehensive explanation of each attribute along with the associated value options in the *Description* column.

Table C.2
Attributes that define a student state.

Attribute	Possible values	Description
Prior knowledge	bad, average, good	Indicates the quality of a student's prior knowledge, before starting the current lecture. Attribute value is estimated by the ITS, based on their performance in previous subjects and previous lectures (for the current subject). A "bad" value indicates that the student has struggled in previous subjects, "average" indicates that they have performed adequately, while a "good" indicates that they have performed good or excellent in previous subjects and lectures.
Typical studying time	day, night, matching	Refers to the typical time (day or night) when a student interacts with the ITS, providing insight into their preferred study schedule. A "day" value indicates a preference for studying during daylight hours, while a "night" value indicates a preference for studying during nighttime hours. A " matching" value is assigned when both "Typical studying time" and "Current time" share the " matching" value, indicating that the student is currently studying within their typical study period. Notably, for action generation, the alignment of "Typical studying time" and "Current time", regardless of being day or night, is the key consideration.
Current time	day, night, matching	Refers to the current time of day or night when the student is using the ITS. The attribute can help identify whether a student is studying during their preferred time of day, or if they

(*continued on next page*)

Table C.2 (*continued*)

Attribute	Possible values	Description
		are studying during a less optimal time. A " matching" value is assigned when both "Typical studying time" and "Current time" share the " matching" value, indicating that the student is currently studying within their typical study period. Notably, for action generation, the alignment of "Typical studying time" and "Current time", regardless of being day or night, is the key consideration.
Previous lecture completion time	less, more, expected	Indicates how much time a student took to complete the previous lecture, compared to the expected time calculated by the ITS. A "less" value indicates that the student completed the previous lecture faster than expected, a "more" value indicates that they took longer, and an "expected" value indicates that they completed within the expected time frame.
Current lecture materials completion time	less, more, expected	Indicates how much time a student took to go through the lecture materials for the current lecture, compared to the expected time calculated by the ITS. A "less" value indicates that the student completed the lecture materials faster than expected, a "more" value indicates that they took longer, and an "expected" value indicates that they completed within the expected time frame.
Current lecture assignments time	less, more, expected, not available	Indicates how much time a student spent working on the current lecture's assignments up to their current progress point, compared to the expected time calculated by the ITS. The attribute considers time spent on lecture assignments, regardless of completion status. "The attribute "Current lecture assignments time" is used in conjunction with the "Percentage of successfully completed assignments for current lecture" attribute to determine the efficiency of a student's progress in completing the lecture's assignments. It provides insight into how quickly a student is working through the assignments. A "less" value indicates that the student spent less time (solving assignments up to their current progress point) than expected, a "more" value indicates that they took longer, and an "expected" value indicates that they completed within the expected time frame. If a student hasn't yet started solving assignments up to their current point in time, we utilize a "not available" value.
Percentage of successfully completed assignments for current lecture	0, 20, 50, 80, 100	Percentage of completed assignments for the current lecture up to the student's current progress point. It indicates the proportion of tasks that they have successfully solved. A higher value indicates that the student is making more progress.
Affective state	bored, frustrated, motivated, not available	Refers to the current affective state that the student is experiencing. If a student isn't feeling bored, frustrated, or motivated, we utilize a "not available" value.
Lecture materials review status	True, False	Indicates whether the student has additionally reviewed the lecture materials up to their current progress point for the current lecture. A value of "True" indicates that the student has reviewed the lecture materials at least once, either in full or in part.
Took a break status	True, False	Indicates whether a student has taken a break during the current study session with ITS. A value of "True" indicates that the student has taken a break, while a value of "False" indicates that they have not.
Tried to solve similar problems status	True, False	Indicates whether a student has tried to solve similar problems (assignments) as the one they are currently struggling with. These similar problems are identified by the ITS, which tracks this attribute based on student's previous interactions with similar problems. A value of "True" indicates that the student has tried to solve similar problems to overcome their difficulties, while a value of "False" indicates that they have not.

Appendix D. Dataset characteristics

The annotated dataset consists of 11 attributes and 17 labels (actions). There are no outliers nor missing values in the dataset, as we programmatically generated student states. All attributes are categorical, except *Percentage of successfully completed assignments for current lecture* which has discretized numerical values (we found experimentally that ChatGPT performs better when values of continuous attributes are discretized, as mentioned in Section 3.2.2 ("Student state preprocessing"). This numerical attribute has a mean of 52, a standard deviation of 35, a median of 50, a minimum value of 0, and a maximum value of 100. Other attributes are categorical, and we present their characteristics in Table 3.

Table 3
Characteristics of the annotated dataset attributes.

Categorical attribute	Number of instances	Unique values	Most frequent value	Number of instances of the most frequent value
Prior knowledge	260	3	average	94
Typical studying time	260	3	day	95
Current time	260	3	night	95
Previous lecture completion time	260	3	more	90
Current lecture materials completion time	260	3	expected	98
Current lecture assignments time	260	4	less	79
Affective state	260	4	motivated	82
Lecture materials review status	260	2	False	139
Took a break status	260	2	False	153
Tried to solve similar problems status	260	2	False	167

References

Abdelkefi, M., & Kallel, I. (2017). Towards a fuzzy multiagent tutoring system for M-learners' emotion regulation. In *2017 16th International Conference on Information Technology Based Higher Education and Training (ITHET)* (pp. 1–6). https://doi.org/10.1109/ITHET.2017.8067821

Alkhatlan, A., & Kalita, J. (2018). *Intelligent tutoring systems: A comprehensive historical survey with recent developments* (arXiv:1812.09628). *arXiv*. https://doi.org/10.48550/arXiv.1812.09628

Almasri, A., Ahmed, A., Al-Masri, N., Sultan, Y.A., Mahmoud, A.Y., Zaqout, I., Akkila, A. N., & Abu-Naser, S.S. (2019). *Intelligent Tutoring Systems Survey for the Period 2000-2018*. 3(5).

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J. H., Shafey, L. E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., & Wu, Y. (2023). *PaLM 2 technical report* (arXiv:2305.10403). *arXiv*. http://arxiv.org/abs/2305.10403.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., & Amodei, D. (2020). *Language models are few-shot learners* (arXiv:2005.14165). *arXiv*. http://arxiv.org/abs/2005.14165.

Crow, T., Luxton-Reilly, A., & Wuensche, B. (2018). Intelligent tutoring systems for programming education: A systematic review. In *Proceedings of the 20th Australasian Computing Education Conference* (pp. 53–62). https://doi.org/10.1145/3160489.3160492

D'Mello, S., & Graesser, A (2012). Dynamics of affective states during complex learning. *Learning and Instruction, 22*(2), 145–157. https://doi.org/10.1016/j.learninstruc.2011.10.001

Dong, Q., Li, L., Dai, D., Zheng, C., Wu, Z., Chang, B., Sun, X., Xu, J., Li, L., & Sui, Z. (2023). *A survey on In-context Learning* (arXiv:2301.00234). *arXiv*. http://arxiv.org/abs/2301.00234.

Khandelwal, U., He, H., Qi, P., & Jurafsky, D. (2018). *Sharp nearby, fuzzy far away: How neural language models use context* (arXiv:1805.04623). *arXiv*. http://arxiv.org/abs/1805.04623.

Lo, C. K. (2023). What is the impact of ChatGPT on Education? A rapid review of the literature. *Education Sciences, 13*(4), 410. https://doi.org/10.3390/educsci13040410

Luburić, N., Vidaković, D., Slivka, J., Prokić, S., Grujić, K.-G., Kovačević, A., & Sladić, G. (2022). Clean code tutoring: makings of a foundation. In *Proceedings of the 14th International Conference on Computer Supported Education* (pp. 137–148). https://doi.org/10.5220/0010800900003182

Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H., & Zettlemoyer, L. (2022). *Rethinking the role of demonstrations: What makes in-context learning work?* (arXiv:2202.12837). *arXiv*. http://arxiv.org/abs/2202.12837.

Mousavinasab, E., Zarifsanaiey, N., R Niakan Kalhori, S., Rakhshan, M., Keikha, L., & Ghazi Saeedi, M. (2021). Intelligent tutoring systems: A systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments, 29*(1), 142–163. https://doi.org/10.1080/10494820.2018.1558257

OpenAI. (2023). GPT-4 technical report (arXiv:2303.08774). *arXiv*. http://arxiv.org/abs/2303.08774.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*(7587), 484–489. https://doi.org/10.1038/nature16961

Tang, R., Han, X., Jiang, X., & Hu, X. (2023). *Does synthetic data generation of LLMs help clinical text mining?* (arXiv:2303.04360). *arXiv*. http://arxiv.org/abs/2303.04360.

Wang, S., Liu, Y., Xu, Y., Zhu, C., & Zeng, M. (2021). Want to reduce labeling cost? *GPT-3 Can Help. Findings of the Association for Computational Linguistics: EMNLP, 2021*, 4195–4205. https://doi.org/10.18653/v1/2021.findings-emnlp.354

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., & Fedus, W. (2022). *Emergent abilities of large language models* (arXiv:2206.07682). *arXiv*. http://arxiv.org/abs/2206.07682.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2023). *Chain-of-thought prompting elicits reasoning in large language models* (arXiv:2201.11903). *arXiv*. http://arxiv.org/abs/2201.11903.