Generative Large Language Models for Dialog-Based Tutoring: An Early Consideration of Opportunities and Concerns

Benjamin D. Nye[1] [0000−0002−5902−9196], Dillon Mee[1], and Mark G. Core[1]

University of Southern California Institute for Creative Technologies, Playa Vista, CA 90094, USA nye@ict.usc.edu

Abstract. After many years of relatively limited capabilities for generative language models, recent large language models (LLM's) have demonstrated qualitatively better capabilities for understanding, synthesis, and inference on text. Due to the prominence of ChatGPT's chat system, both the media and many educational developers have suggested using generative AI to directly tutor students. However, despite surface-level similarity between ChatGPT interactions and tutoring dialogs, generative AI has other strengths which may be substantially more relevant for intelligent tutoring (e.g., detecting misconceptions, improved language translation, content generation) and weaknesses that make it problematic for on-the-fly tutoring (e.g., hallucinations, lack of pedagogical training data). In this paper, we discuss how we are approaching generative LLM's for tutoring dialogs, for problems such as multi-concept short answer grading and semi-supervised interactive content generation. This work shows interesting opportunities for prompt engineering approaches for short-answer classification, despite sometimes quirky behavior. The time savings for high-quality content generation for tutoring is not yet clear and further research is needed. The paper concludes with a consideration of longer-term equity and access in a world where essential capabilities require low-latency real-time connections to large, pay-per-use models. Risks and mitigating technologies for this kind of "AI digital divide" are discussed, including optimized / edge-computing LLM's and using generative AI models as simulated students to train specialized tutoring models.

Keywords: Intelligent Tutoring Systems · Conversational Tutoring · Generative Models · Large Language Models · Short Answer Grading · Content Generation

1 Introduction

A student studies how DNA works by asking for information from a chat system, which answers all their questions about the parts of genes they find unclear. Is this a tutoring system? Since the rise of ChatGPT [12], the lines between a classical information system (e.g., Google Search) and a dialog-based assistant have blurred. To some extent, this process has been ongoing for many years: Google Search results generate question-answer pairs and text summaries; Amazon Alexa tries to paraphrase Wikipedia facts; etc. However, ChatGPT is notable because it pioneers the combination of two capabilities: highly-effective text instruction (prompting) and meaningful multi-turn context. On the surface, using ChatGPT to study a topic can look somewhat similar to a tutoring dialog: the highly-knowledgeable "tutor" answers all the questions to help the learner and (sometimes) can suggest places to read more if it doesn't have all the answers.

However, research on human tutors shows this is mostly *not* how effective tutoring works. The described interaction more closely resembles a reference librarian or an inhumanly fast legal researcher. It is fairly straightforward to establish that ChatGPT and any relatively easy extension is not a tutor. Expert tutors typically differ in three major ways:

1. Initiative: The tutor guides the conversation stages and direction. Even when answering questions or listening to extensive student explanations, they are managing the interaction rather than reacting to it.
2. Answering vs. Questioning: When tutors interact with a student, they are often posing questions (e.g., prompting the student to think) rather than just responding to questions. Even when providing explanations, the tutor typically uses these to set up their next tutoring moves.
3. Pedagogical Domain Knowledge vs. Domain Knowledge: Tutors and instructors are uniquely experienced with a wide range of student understandings and misconceptions, as well as which strategies seem to work better with different students. Current LLM data sets lack many examples of pedagogical domain knowledge.

Despite this, the utility of the "Instruct GPT" technology behind ChatGPT remains compelling, and can at a minimum produce simple interactions that are known to be helpful. For example, research shows that self-testing [18] or self-reflection dialogs [21] improve learning versus merely reading passive materials like a textbook. This raises significant questions. What roles can generative AI play between a free-form dialog and a close-ended knowledge check? Why might a more flexible dialog agent be a worse tutor? When could generative AI be ready to tutor and what would the implications be?

In this work, we present our current perspective on these issues from the standpoint of exploring these emerging models. Due to the rapid convergence of generative AI capabilities, the views and assumptions in this paper will be inherently more speculative than the typical paper. As such, the goal is to contribute to the present-day discussion and to offer a reference point to re-evaluate 2-3 years in the future, when research studies have established empirical evidence on the areas where generative AI models have successfully improved versus failed to advance learning outcomes.

2 Background: Generative LLM's for Education

Compared to many AI topics, the history of generative language models for intelligent tutoring systems is relatively thin. Excluding template-based systems (e.g., generating math problems by customizing fields), little research was conducted on this topic until the last few years. In part, this has been due to the poor quality of generated text. Early successes were not very useful for most learning contexts, such as simple summaries, paraphrases, or structured narratives (e.g., baseball play-by-play). Moreover, even research as recent as this year reported that GPT-3 produced math explanations that teachers rejected about 50% of the time [16]. Similarly, research with ChatGPT indicates that it produced better-written hints (70% acceptable by experts) but that these accepted hints may have still produced lower learning gains than human-generated hints [14]. Anecdotally, at least for math, GPT-4 [13] may produce significantly better explanations than GPT-3, but as reported in [14], such content may still be less effective than teachers would make themselves.

In short, even as recently as early 2023, the quality of raw content generated by an LLM has not shown that it should be used with students if human-created content is available instead. While the quality of generated content is rising rapidly year after year (e.g., one current but informal estimate is 80% acceptance), a rejection rate of even 5% would make teachers wary of students using the content as-is. The nature of errors is also problematic, with well-documented issues of hallucination (e.g., making up fictional facts) and over-confident explanations [6]. While emerging practices in prompt engineering and improved models offer ways to reduce these issues, they are to some extent inherent and may not be resolved reliably in the next few years. As a predictive model generates a stream of coherent and mostly self-consistent text, one turn down the wrong path can lead to an extremely coherent rationale for why the wrong answer is right. From a learning standpoint, this is among the worst failures: a compelling, well-reasoned misconception.

A final issue for LLM tutoring is that there is no cut-and-dry evidence that supporting exploration and open-ended question answering provides greater learning efficiency. Conversely, research efforts have found that in some cases less exploration is appropriate: vicarious tutoring (watching a teacher agent work with a student agent) can be particularly effective for low-knowledge learners [3], expert tutors in some cases provide more prescriptive "collaborative lectures" than less-experienced tutors [11], and worked examples have been almost 50% more efficient than problem-solving for certain types of learning [8]. This is compounded by the fact that interacting productively with a relatively unconstrained dialog-based tutor has distinct issues. First, students with weak self-regulated learning (SRL) skills may struggle to ask productive questions and require training on SRL to engage effectively (e.g., metacognitive scaffolding [2]). While this may be a strength in the long term, approaching both problems simultaneously poses a challenge. Second, students may become distracted with content that neither aligns with their course nor their long-term goals (e.g., similar to reading through a series of Wikipedia links despite finding the desired information).

Ultimately, open-ended conversations may prove most effective for students with strong SRL skills, who often already learn faster.

Despite these concerns for generative models chatting in real-time educational conversations, the case for semi-supervised content remains strong. First, there is no requirement that LLM-generated explanations or hints must be used as-is. Instructors and content developers regularly improve or adapt mixed-quality content. Second, significant infrastructure is growing to enable comparison of different types of learning content, either across versions of a course template (e.g., the WISE project; [22]), through randomized micro-trials of certain content [24], or through frameworks designed to deploy large-scale A/B testing of interventions [17]. Tools to more rapidly produce, review, and revise content can have powerful synergies with frameworks designed to evaluate the effects of these changes. In the long term, such results can also be used in a feedback loop where generative models can be tuned and prompted to produce higher-quality educational materials.

3 Current Directions: Generative LLM's for Tutoring

In ongoing work, we have started exploring prompt-based LLM approaches to support dialog-based intelligent tutoring. We are targeting two problems where LLM's have already shown advantages: automated short answer grading (ASAG) [7,23] and content generation suitable for expert review [14].

This work is done in the context of OpenTutor (www.github.com/opentutor/ [10]), a rapidly-authorable dialog-based tutoring system designed to allow instructors and other non-specialized authors to create and share tutoring dialog lessons. OpenTutor follows expectation-based tutoring approaches based on AutoTutor, which has demonstrated learning gains on a variety of domains [5,9]. While instructors using OpenTutor have reported that it has high usability and instructors can create a dialog fairly easily, tuning the grading of answers remains a key technical challenge: the classifier starts with only a single ideal answer statement for each expectation concept. Typically, about 20-25 students must complete tutoring sessions and instructors grade student answers to provide sufficient training data for good-quality evaluation of new student answers (e.g., 85% accurate or higher on each expectation concept).

Cold start performance with the fewest number of examples (0-shot and N-shot classification) is important to save instructor grading time and avoid potential student frustration for the first batch of learners. For current OpenTutor dialogs, the classifier typically starts at 65% to 80% accuracy without any labeled student answers. This is a multi-label classification task, as every student answer is graded on each concept as either true (student showed they knew the concept) or false (student did not show that they knew the concept). The confidence of the classifier must also be reported, as the OpenTutor dialog model responds differently to low-confidence classifications (e.g., neutral feedback and responses). Since it is infeasible to fine tune a language model for each new lesson, prompt-based N-shot approaches offer a way to improve the classifier when little or no data is available.

LLM prompts will also be explored as a tool to take an input document (e.g., cliff notes, video transcript) and learning objective(s), and generate an OpenTutor lesson candidate. This task is substantially more complex than previously studied content generation, because it requires generating a question prompt, expectation ideal answers, and for each expectation an average of two hints framed as leading-questions (e.g., "If you changed the polarity, what would happen?"). Moreover, the order of the expectations and the hints are pedagogically meaningful (e.g., hints should progress from more general to more specific). The first step is prompt engineering to determine the feasibility of generating each element. While existing research indicates that it should be possible to generate an OpenTutor lesson candidate, significant evaluation would need to be conducted to quantify the added-value of these dialogs. The lesson candidates must save significant time (i.e., selecting and revising a lesson must be easier than making one manually), and the dialogs must produce equivalent or better engagement and learning gains.

4 Prompt Engineering for Short-Answer Grading

Our exploratory research is investigating the feasibility of using LLM's to improve short-answer grading when limited data is available through design of a prompt which can be used to classify a student input for its coverage of multiple concepts simultaneously. The inputs to this prompt are a list of ideal answer expectations (labeled "concept_1" to "concept_N"), a JSON structure for N-shot labeled examples, and one or more student answers to classify. The output is a JSON structure containing the label for each concept (true/false), the confidence for that label, and a text justification for the label. In this initial work, ChatGPT is being used, though since only a single prompt is provided (no chaining) it should be effectively equivalent to a single InstructGPT prompt.

An example prompt is presented in Appendix A. Due to rapid iteration on the prompts at this time, it is only one of a number of candidate templates being explored. The design of these prompts was based on surveying a large variety of examples and approaches (e.g., pre-publications, Reddit communities, and blog posts; e.g., [4]), with the most influential being efforts to minimize hallucinations while retaining effective inference. We also asked the model itself what approaches should be most effective for issues such as providing data. From this guidance, we identified a number of the key prompt engineering elements that are required for a short answer grading task:

1. Instructions: Short narrative directions and context clarifications, which are minimally-sufficient to get consistent and meaningful results
2. Structured Data: Highly structured (e.g. JSON, CSV) inputs and outputs
3. Justification: Requesting that the model justify the reason for its label, rather than only generating the label

4. Emphasizing Key Elements: Insisting, repeating, or asking to double-check key instructions to give them greater weight (e.g., that the output must be valid JSON).

As shown in Appendix A, there is an initial system instruction, which contextualizes the task–in this case that of a tutor evaluating answers–and then presents unlabeled data in JSON format. In the N-shot examples, there is then a brief narrative that provides "ground truth" examples followed by labeled data in JSON format. Ground truth was the terminology used by ChatGPT when responding to a question about preferred formats. The user prompt next indicates the steps of evaluation and response, which include a brief justification and a confidence score included in a JSON formatted template for a response. Finally, there is a coda statement that asks for confirmation that all responses are properly formatted. In this statement, we also use a special term for setting the "temperature" of the model. This is associated with the level of variation in generation: a temperature of 0 will produce the same result each time, while higher values are typically used for creative tasks (e.g., fiction writing prompts). Using this formula the response is consistent and in most cases quite effective, with baseline responses often yielding an accuracy of above 80% on certain initial examples tested.

Figure 1 shows exploratory results of ChatGPT as a short-answer multi-concept classifier for answers for a dialog on reaching out for mental health services (Reaching Out) and Figure 2 shows similar results for a dialog related to suicide prevention (Prevention). Each example shows a single run where the same test set of 20 answers is used while the number of labeled examples (N-shots) is increased from 0 to 15. Despite this, fairly clear trends show improvement of the F1 scores for both dialogs as more examples are provided. However, examining specific expectations shows that in many cases one or two expectations are classified fairly well even at baseline, while other concepts require substantially more examples. While not shown, exploratory work with other prompt structures appeared to show improvements up to about 25 or 30 examples, which is similar to the number needed for higher-quality evaluation with OpenTutor's current classifiers. Considering classifications of the two dialogs, it is also interesting that the one appears to be pickier than the other (e.g., one consistently shows higher precision while the other shows higher recall). Further prompt engineering or post-processing might be useful to enable authors to tune the classifier to be more selective versus more permissive in accepting answers.

However, some elements remain less reliable or quirky. External verification of inputs versus outputs is an obvious need. In some cases, labeled examples help induce highly insightful justifications (e.g., correctly identifying metaphors or word-play) while in other cases relevant examples appear to be ignored entirely. In some cases the output format is not produced correctly or an answer classification is missing when multiple were requested. On the converse, in one case it actually generated an *additional* answer that was classified (see Fig. 3 below). Second, the model can be highly influenced by seemingly modest changes. For example, when tests included the tutor's main question, this appeared to cause the grading to be much harsher and hurt the accuracy significantly as a result.

As a third intuition, once the LLM answers in the wrong direction for a prompt, it appears to cause additional cascading errors (e.g., it added "I'm not sure" as an answer and also said that was a correct match for an expectation, which it was most certainly not). This implies that if a validator or critic model is used to check LLM answers, catching a single problem may require re-generating the whole response. Fourth, the model has particular tendencies that are hard to override. For example, the generated "confidence" produces 0-1 scores easily but may ignore instructions to use a -1 to 1 range. Moreover, confidence ratings are connected to the "temperature" parameter: reducing the temperature to zero results in a confidence of 1 for every label. Assuming the confidence scores may

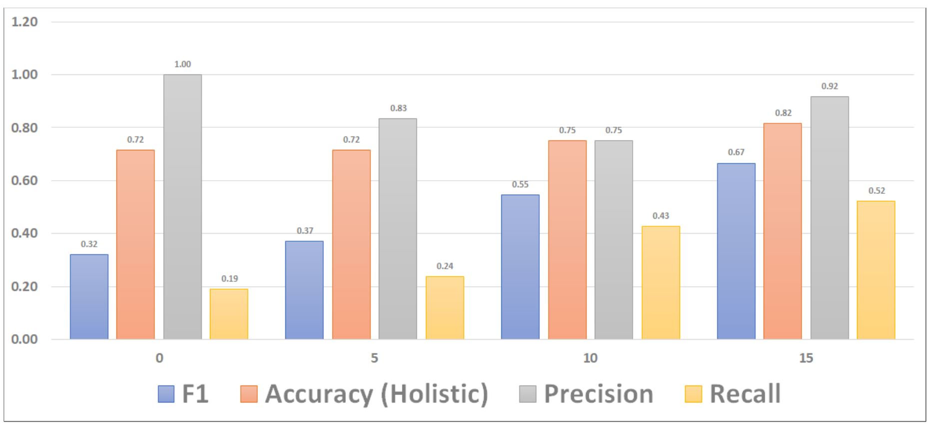

Fig. 1. "Reaching Out" Classifier Performance by N-Shots Labeled

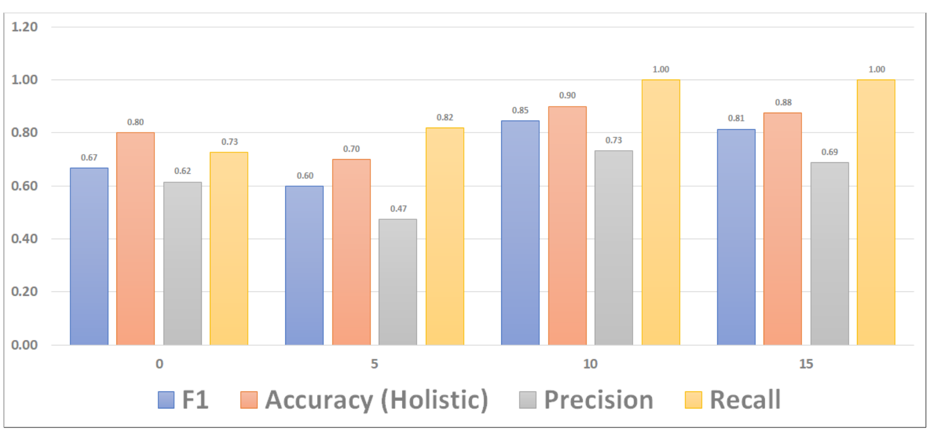

Fig. 2. "Prevention" Classifier Performance by N-Shots Labeled

```
{
  "answer text": "I'm not sure",
  "concept_1": "true",
  "concept_2": "false",
  "confidence_1": 0.8,
  "confidence_2": 0.1,
  "justification": "The answer 'I'm not sure' implies uncertainty about
       the person's suicide risk, indicating an understanding of
      Concept 1. However, it doesn't explicitly address the presence of
       ongoing suicidal thoughts, leading to uncertainty regarding
      Concept 2."
}
```


Fig. 3. Hallucination Example: Classifying a fictional student answer

be meaningful and useful (which seems to be the case), settings to use the model deterministically breaks the confidence scores.

While this work is at an early stage, one key finding is the necessity to wrap generative models with external systems for standardization and evaluation rather than using their raw results to interact with learners. This is likely relevant at multiple levels, beyond the basic elimination of hallucinations or detecting a missing classification. While minor tweaks to submissions can have significant impacts (in one case changing a single classification label was enough to yield a drop in recall score by over .3), this remains an outlier. In general, when properly structured, classification responses yield F1 scores ranging .6 to .9, where greater numbers of labeled data in the prompt improves the F1 score. The generated justifications also represent interesting candidates for tutoring dialog statements, but due to the cascading error problem these would only be trustworthy if a secondary higher-reliability classifier agrees with the label. Further areas of investigation include: the impact of multi-stage versus single-stage prompting, methods for validating confidence scores, the impact of asking for various numbers of classifications simultaneously (e.g. 2 concepts vs. 5 concepts), and optimal batch sizes (e.g., how many student answers can be graded with a single "setup prompt" without introducing noise).

5 Conclusions and Future Directions

The exploration conducted so far indicates that prompt engineering can be an effective method for short answer grading with ChatGPT which is expected to complement the existing lightweight classifier from OpenTutor. More generally, this research also integrates some specific techniques that improve classification using InstructGPT prompts (e.g., requiring a justification, structured example data, clear and emphatic context). Despite intermittent failures due to hallucination or other issues, this research indicates that it is feasible to plug a LLM into a real-time tutoring system for answer evaluation by wrapping it in services to generate the prompt and validate the answer format. Upcoming research will investigate the benefits of ensembling the LLM with the existing OpenTutor classifier, which remains an important fallback model and a model that can be used offline without an internet connection.

Research on generating interactive learning content with LLM is still at the exploratory stage. However, compared to raw LLM chat dialogs, we believe that semi-supervised content generation is more appropriate for the current state of the art. This should not be considered a negative: the ability to rapidly generate related versions of content should make it possible to systematically compare different types of content for different kinds of learners and situations (e.g., self-reflection prompt vs. multiple choice quiz vs. collaborative lecture vs. expectation-based tutoring, etc.). Moreover, for intelligent tutoring systems and other interactive learning content, generative tools which speed up producing or updating content will be highly significant. Despite well-established research that active, interactive, and constructive learning activities produce greater learning [1], many courses spend over 80% of class time on passive content (e.g., lectures, text, video) [19]. Enabling faster and easier generation and grading of interactive learning content could help shift that balance, making interactive content more prevalent. Moreover, in the cases where no vetted active learning content exists yet (e.g., emerging topics), an open-ended learning dialog about the content should still provide value much like how self-testing improves learning.

Unfortunately, as hosted and proprietary LLM's become essential for emerging learning technologies, this also raises the risk of a new digital AI-divide. Due to their memory footprint and computational requirements, current LLM models charge a non-trivial fee per use and, perhaps more problematic, require real-time responses from a reliable internet connection. Even as internet access grows worldwide, technology and business models for AI using large foundation models (LLM's, AI vision, etc.) may remain less accessible to low-resource learners. Even in more developed regions, low-income learners may be motivated to use services centered on selling user data, with risks for privacy and leaks of fine-grained information. While this is an ongoing problem, LLM systems increase both the scope of data that can be inferred and the scope of services where centralized AI models may be a critical feature.

Future research should investigate approaches to democratize LLM access, such as by reducing their requirements and other techniques to make them highly accessible. For example, efforts are underway to build "distilled" models based on LLM, such as a specialized short-answer grading model based on ChatGPT [7]. However, due to the current licensing constraints, many of these models remain incompatible with distribution. Alternatively, LLM might be used to develop the behavior of better simulated students who test out content or policies from smaller specialized models. In this role, the LLM would be part or all of a cognitive model designed to produce better training data for other models. Exploration of these approaches has already started in domains such as teaching programming skills [15] and biology [20].

Policies and standards to ensure that equitable use of these technologies are also important, particularly for versions designed to meet the needs of developing regions (e.g., inconsistent internet, mobile-first computing devices). Smartphone platforms (e.g., iOS, Android) can play an important role for increasing equity by expanding mechanisms to download shared AI/ML libraries accessible across apps. By establishing approaches to ensure equitable access now, these models can be replicated as increasingly advanced and useful models set a new baseline for what we can accomplish with educational technologies.

Acknowledgments. This research work was sponsored through the USC ICT University Affiliated Research Center (UARC; Army W911NF-14D-0005). However, the views expressed in this publication do not necessarily reflect the official policies of the Department of Defense nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

References

1. Chi, M.T., Wylie, R.: The icap framework: Linking cognitive engagement to active learning outcomes. Educational psychologist **49**(4), 219–243 (2014)
2. Choi, I., Land, S.M., Turgeon, A.J.: Scaffolding peer-questioning strategies to facilitate metacognition during online small group discussion. Instructional science **33**(5-6), 483–511 (2005)
3. Craig, S.D., Gholson, B., Brittingham, J.K., Williams, J.L., Shubeck, K.T.: Promoting vicarious learning of physics using deep questions with explanations. Computers & Education **58**(4), 1042–1048 (2012)
4. Fan, J.: How we cut the rate of GPT hallucinations from 20%+ to less than 2% (2023), https://jfan001.medium.com/how-we-cut-the-rate-of-gpt-hallucinations-from-20-to-less-than-2-f3bfcc10e4ec
5. Graesser, A.C., D'Mello, S., Hu, X., Cai, Z., Olney, A., Morgan, B.: Autotutor. In: Applied natural language processing: Identification, investigation and resolution, pp. 169–187. IGI Global (2012)
6. Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., et al.: Chatgpt for good? on opportunities and challenges of large language models for education. Learning and Individual Differences **103**, 102274 (2023)
7. Li, J., Gui, L., Zhou, Y., West, D., Aloisi, C., He, Y.: Distilling chatgpt for explainable automated student answer assessment (2023)
8. McLaren, B.M., van Gog, T., Ganoe, C., Karabinos, M., Yaron, D.: The efficiency of worked examples compared to erroneous examples, tutored problem solving, and problem solving in computer-based learning environments. Computers in Human Behavior **55**, 87–99 (2016)
9. Nye, B.D., Graesser, A.C., Hu, X.: Autotutor and family: A review of 17 years of natural language tutoring. International Journal of Artificial Intelligence in Education **24**, 427–469 (2014)
10. Nye, B.D., Sanghrajka, R., Bodhwani, V., Acob, M., Budziwojski, D., Carr, K., Kirshner, L., Swartout, W.R.: Opentutor: Designing a rapid-authored tutor that learns as you grade. In: The International FLAIRS Conference Proceedings. vol. 34 (2021)

11. Olney, A.M., D'Mello, S., Person, N., Cade, W., Hays, P., Williams, C., Lehman, B., Graesser, A.: Guru: A computer tutor that models expert human tutors. In: Intelligent Tutoring Systems: 11th International Conference, ITS 2012, Chania, Crete, Greece, June 14-18, 2012. Proceedings 11. pp. 256–261. Springer (2012)
12. OpenAI: ChatGPT (2023), https://chat.openai.com/chat
13. OpenAI: Gpt-4 technical report (2023)
14. Pardos, Z.A., Bhandari, S.: Learning gain differences between chatgpt and human tutor generated algebra hints. arXiv preprint arXiv:2302.06871 (2023)
15. Phung, T., Cambronero, J., Gulwani, S., Kohn, T., Majumdar, R., Singla, A., Soares, G.: Generating high-precision feedback for programming syntax errors using large language models. arXiv preprint arXiv:2302.04662 (2023)
16. Prihar, E., Lee, M., Hopman, M., Kalai, A., Vempala, S., Wang, A., Wickline, G., Heffernan, N.: Comparing different approaches to generating mathematics explanations using large language models. In: AIED 2023 (2023)
17. Ritter, S., Heffernan, N., Williams, J.J., Lomas, D., Motz, B., Basu Mallick, D., Bicknell, K., McNamara, D., Kizilcec, R.F., Roschelle, J., et al.: Third annual workshop on a/b testing and platform-enabled learning research. In: Proceedings of the Ninth ACM Conference on Learning@ Scale. pp. 252–254 (2022)
18. Rodriguez, F., Kataoka, S., Janet Rivas, M., Kadandale, P., Nili, A., Warschauer, M.: Do spacing and self-testing predict learning outcomes? Active Learning in Higher Education **22**(1), 77–91 (2021)
19. Sheridan, B.J., Smith, B.: How often does active learning actually occur? perception versus reality. In: AEA Papers and Proceedings. vol. 110, pp. 304–08 (2020)
20. Sonkar, S., Liu, L., Mallick, D.B., Baraniuk, R.G.: Class meet spock: An education tutoring chatbot based on learning science principles. arXiv preprint arXiv:2305.13272 (2023)
21. Sullins, J., Craig, S.D., Hu, X.: Exploring the effectiveness of a novel feedback mechanism within an intelligent tutoring system. International Journal of Learning Technology **10**(3), 220–236 (2015)
22. Ulus, B., Oner, D.: Fostering middle school students' knowledge integration using the web-based inquiry science environment (wise). Journal of Science Education and Technology **29**, 242–256 (2020)
23. Wadhwa, M., Chen, J., Li, J.J., Durrett, G.: Using natural language explanations to rescale human judgments (2023)
24. Williams, J.J., Rafferty, A.N., Tingley, D., Ang, A., Lasecki, W.S., Kim, J.: Enhancing online problems through instructor-centered tools for randomized experiments. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 1–12 (2018)

Appendix A InstructGPT / ChatGPT Prompt Template

```
System:
The user provided a list of one or more "answers" to a tutoring question.
 The answers are provided in JSON format. You are a tutor who is
evaluating if the answer is sufficient to show that the user knows a one
or more "concepts" which will be labeled "concept_1" to "concept_N". For
each answer you evaluate, you must also express how confident you are in
your evaluation of how well the answer shows knowledge of each concept.
You will also provide a brief justification.
```

```
Concepts and Answers:
concept_1: They're much less likely to commit suicide now or in the
future, because suicidal urges typically last less than an hour.
concept_2: The person is still at risk compared to other people, because
they still have suicidal thoughts.

{
 "answer_68": {
   "Answer text": "not attempt"
  },
  "answer_106": {
   "Answer text": "find another way"
  },
  "answer_34": {
   "Answer text": "It's lessened"
  }
}
Ground Truth Examples:
Here are some examples that have already been labeled. They are presented
 in JSON format, where the answer is given, followed by each concept and
a true or false label. Consider these to be ground truth examples.
{
  {
   "Answer text": "I would say that I am unsure about people attempting
to commit suicide more than once. I don't know enough about the topic.",
   "concept_1": "false",
   "concept_2": "false"
  },
  {
   "Answer text": "they're less likely now",
   "concept_1": "true",
   "concept_2": "false"
  },
  {
   "Answer text": "yes it goes down",
   "concept_1": "true",
   "concept_2": "false"
  },
  {
   "Answer text": "the same",
   "concept_1": "false",
   "concept_2": "false"
  },
  {
   "Answer text": "I think I would be less likely give more time to
think about it. Just thinking about the effect it would have in other
peoples lifes.",
   "concept_1": "true",
   "concept_2": "false"
```

```
  },

User:
First, for each individual answer, classify whether or not it
demonstrates understanding of each key concept.
Second, respond with a JSON in the format, for each concept.
{"answer number": //Show the text of the answer being classified
{
"answer text": string // state the text of the particular answer being
classified.
"concept_N": string // true or false. If the input answer implies that
the concept is known, probability //certainty that your classification is
 correct.
"confidence_N": float // A -1 to 1 score indicating certainty that a
classification is correct. 1 is perfect confidence in true, -1 is
perfectly confident in false.
"justification": string // Why you believe the user answer is or is not
sufficient to determine if they know the concepts.
}
}
Only respond with the JSON output and no other words or symbols. The
output must be valid JSON. Check that the output is valid JSON. Return
responses with a temperature of 0.3.
```