Tracking the Evolution of Student Interactions With an LLM-Powered Tutor

Kevin Gold and Shuang Geng
Boston University
klgold@bu.edu, sgeng@bu.edu

ABSTRACT: Student usage of an LLM-powered tutor to get homework help was tracked over the course of a semester in an introductory data science class. For each homework, the GPT-4 powered tutor was given the text of the homework problems and solutions in advance, but was instructed to never reveal solutions directly, but instead guide the student to the correct answer through leading questions. Despite the free availability of ChatGPT, which could have usually produced correct answers, the majority of the class used the system. All interactions were anonymously logged, and samples of these logs were coded on sixteen dimensions of interaction. Evidence from the chatlogs and student surveys indicates that the students found the bot nearly as helpful as the human teaching assistants, and the bot was utilized much more than the staff office hours. But, some patterns of misuse, such as using the bot as a lazy way to check and debug programs, increased over the course of the semester. The coded interactions used for our analysis could be used for further machine learning that could intervene to prevent failures and misuse.

Keywords: Large language models, Chat-GPT4, API, AI-tutor, conversational tutoring, learning analytics, qualitative content analysis, higher education

1. INTRODUCTION

The widespread availability of ChatGPT and other large language models (LLMs) poses huge challenges and opportunities for how education is conducted, and nowhere are these both more apparent than in the practice of assigning homework. On the one hand, the traditional model of assigning graded work to do at home is threatened because students could simply ask an LLM to do it for them, causing the students to learn very little. On the other hand, LLMs could surely be a great tool for immediate responsiveness to student confusions and errors, creating a degree of personal attention that would have been impossible at scale using traditional teaching. This work explores a shift toward the latter model of how homework could work - as a low-stakes environment where students could receive as much help as they needed before being evaluated in an offline exam setting.

Early research on integrating LLM models in classroom has discussed its vast potential for tutoring students with greater accessibility and adaptivity (Wu, Lee, Li, Huang, & Huang, 2023; Mollick & Mollick, 2023). However, its effectiveness is yet to be determined, and early work investigating the use of ChatGPT in Education has rendered mixed results of varied performance across subject domains (Lo, 2023). Our work focuses on evaluating the performance of using a ChatGPT4 API as an AI tutor in a university-level introductory data science in Fall 2023 with 172 students. The AI tutor has three essential modifications that are intended to achieve better learning gains. One, the bot is given the text and solutions to each homework, which are passed along to GPT-4 as part of the query, in order to make student prompting more efficient and increase student usage. This design is intended to achieve a few effects (although not all have been measured): it makes the bot less prone to errors, it makes the bot less prone to unconventional solutions that aren't what the instructor intended, it increases student usage of the system because they have more confidence in its answers, and it increases student usage further by making queries such as "I don't understand 1b" intelligible to the bot. Two, the bot is given a role of roleplaying a helpful teaching assistant whose primary goal is to help the students learn; roles can help succinctly convey a variety of stylistic preferences to an LLM (CITE?), so that the bot is, for example, encouraging and succinct without our need to explicitly specify these behaviors. And three, the bot is given the instruction that it must only reply in questions designed to lead the student to the correct answer, a time-honored teaching technique known as the Socratic method to foster active learning (Jarvic, 2006). (Modifications two and three are interesting because they don't leverage any secret information the way the solutions do, but they are still changes to the context that students are unlikely to wish upon themselves.)

The following is a report on student usage of and attitudes toward the system, which was made available for all eight homework assignments of the course. Every interaction was logged, and samples of these interactions were coded for helpfulness and a variety of student misuse, bot failure modes, and question types. We believe the system could be further improved by adding machine learning systems to monitor and flag these behaviors. We aim to answer three questions regarding critical aspects of student interactions with the AI tutor: 1) How helpful can LLMs be in tutoring college students? 2) What are the main scenarios of bot/student dysfunctions, and what types of questions students tend to ask the bot? 3) What are some significant trends in student usage over time?

2. DATA & METHODOLOGY

The AI tutor was hosted on a HuggingFace space and provided a graphic user interface (GUI) for students to select an assignment via radio button and enter a query. When the query was submitted, the string was augmented with a prefix, "For this query, answer with a single question that you haven't asked before that is meant to lead someone in the right direction, without directly answering the relevant homework question - unless the problem is solved completely, in which case, quit." The query was also augmented with the system information, "You are a helpful teaching assistant in a data science course. Your primary goal is to help the students learn. This is the homework the student was talking about: [homework_text & solution]". The homework_text was the complete homework and solutions, originally a Python notebook but converted to a more basic .py file.

Deleted: GUI

Formatted: Font color: Text 1

Two data sources are evaluated. The first one is chatlogs of student-AI interactions, and were logged to a separate .csv file anonymously, with each GUI session assigned a random identifier to help identify what pieces of dialogue were from the same basic interaction. 802 sessions were recorded in total on 8 homework assignments. The class contained 127 students taking introduction to data science, which concentrated in the first half on teaching basic Python, and in the second half on topics like machine learning and visualization. According to an early poll, roughly 1/3 of the students had taken a class on Python before, 1/3 had at least dabbled in Python, and 1/3 had no programming experience. The 8 assignments were mostly programming assignments, but often introduced other concepts such as machine learning through programming. In-class offline exams designed to test programming skill and factual knowledge were assigned after assignment 6 and assignment 8, and the class culminated in a final project in which LLM use was allowed with attribution. [CUTTABLE BESIDES 1st SENT] To analyze the chatlogs regarding our research questions, we used an inductive coding approach to

Deleted: S

Formatted: Font color: Text 1

Deleted: for the

Formatted: Font color: Text 1

construct a set of initial interaction classifications after the second homework assignment was due. The classifications were iteratively updated as coders (the two authors and a research assistant) identified emerging new themes in later chatlogs. Our unit of coding is each *problem* rather than *session* as students can ask about multiple homework problems in one session. With each shift of conversations towards a new problem, a new coding value is assigned. The codes were mostly chosen to determine what problems might be prevalent but solvable with additional technical intervention, or to gain insight into how students were using the bot. Our final codebook includes 17 dimensions under 4 parent categories: A) Helpfulness of advice includes 4 level of helpfulness; B) bot failure includes 1-Leak approach (3 levels), 2-Clear error, 3- Provide irrelevant answer, 4-Demand extra work, 5-Fail to point back to course materials; C) student misuse includes 1-Select wrong HW in GUI, 2-Spam for hints, 3-Unclear prompt, 4-Mess with the bot; and D) type of question includes 1-Debug request, 2-Review code, 3-Improve style, 4-Clarify concept, 5-Ask for example, 6-Recommend resource, and 7-Ask generally. All dimensions are coded as Boolean variables except A and B1 are ordinal.To get a better idea as to whether student use patterns changed over time, we sampled 10 sessions from each of HW2 to HW7, with a total of 80 sessions and 112 problems rated. HW1 and HW8 are excluded from the sample because in the first assignment, the students are mostly testing around, and the last assignment is optional.

The second source is a pair of midterm and end-semester surveys on students' experience with the AI tutor. The surveys are voluntary-based. 50 copies of midterm surveys and 67 copies of end-semester surveys were collected. The survey results reflect the students' perspective, and are reviewed in comparison to coding results from the chatlogs (instructor's perspective), primarily on the helpfulness dimension.

3. RESULTS

How helpful can LLMs be in tutoring college students?

Student ratings lower than instructor ratings. However, both improved after midterm. [see specifics in 'helpfulness' sheet.

2) What are the main scenarios of bot/student dysfunctions, and what types of questions students tend to ask the bot?

Table 1: Top 2 Themes across Categories B to D, Before/After Midterm

Top Freq.	**Before Midterm** (80 problem units)			**After Midterm** (32 problem units)		
	Bot Failure	**Student Misuse**	**Question Type**	**Bot Failure**	**Student Misuse**	**Question Type**
1	Leak approach (47 occurrences)	Unclear prompt (24)	Debug request (37)	Leak approach (6)	Unclear prompt (10)	Review code (20)
2	Demand extra work (13)	Mess with bot (14)	Review code (29)	Demand extra work (6)	Mess with bot (9)	Ask generally (14)

3) What are some significant trends in student usage over time? [should I do logistic regressions over timestamp/HW number instead?]

Lastly, Fisher's exact tests were performed on all dimensions to discern significant trends in student usage before and after the midterm. Statistically significant results were observed on B1-Leak approach (p-value=8.43E-04), D1-Debug request (p=2.46E-05), and D2-Review code (p=1.96E-2). As an example of visualizing the change, Figure 1 shows boxplots of the 4 levels of B1-Bot leaks approach, across homework (0-no leak to 4-full leak). The average level changes from 1.20 (near mild leak) in HW2 to 0.35 HW7 (almost no leak), showing an improvement in bot's tutoring technique following the Socratic method. The occurrence of debug request (student coming to the bot with an error message) went down from 46% to 6%, while that of code review (pouring the code directly) went up from 36% to 63%. This is possible due to student's increased trust and familiarity with the bot's ability so that they feel comfortable interacting with it with minimal prompts [other interpretation?].

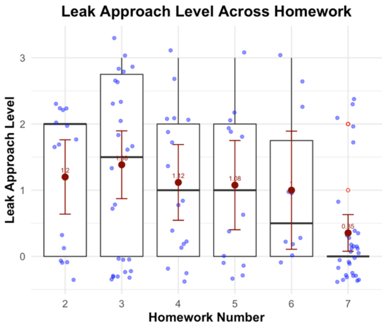

Figure 1: Boxplots of B1-Leak approach (with average points and confidence intervals in red)

4. CONCLUSIONS

The evidence generally points to the advice given by the bot being perceived as generally helpful by both students and coders evaluating the interactions. However, the evidence also suggests that misuse can rise over time as students learn to use the system in unintended ways, such as just using it as a convenient answer checker or a "debugger" that is powerful enough to write the program one debugging step at a time. Further research will look at how to intercept or flag problematic interactions as they happen. The mining of the student interactions for useful information for the instructor - for common misconceptions or omissions from lectures - is a further possible direction.

2. THIS IS A NUMBERED HEADING 1 USING JLA_HEADING 1 STYLE

This text[1] links to a footnote.

REFERENCES

Jarvis, P. (2006). The Socratic method. *The theory and practice of teaching*, 90-97.

Lo, C. K. (2023). What is the impact of ChatGPT on education? A rapid review of the literature. *Education Sciences*, 13(4), 410.

Mollick, E. R., & Mollick, L. (2023). Assigning AI: Seven Approaches for Students, with Prompts. Retrieved September 23, 2023. Available at SSRN: https://ssrn.com/abstract=4475995 or http://dx.doi.org/10.2139/ssrn.4475995

Wu, T.-T., Lee, H.-Y., Li, P.-H., Huang, C.-N., & Huang, Y.-M. (2024). Promoting Self-Regulation Progress and Knowledge Construction in Blended Learning via ChatGPT-Based Learning Aid. *Journal of Educational Computing Research*, 61(8), 3-31. https://doi.org/10.1177/07356331231191125

[1] Footnotes are reserved for URLs or commentary. Keep references for the References section.