https://doi.org/10.36315/2024v2end019
Education and New Developments 2024

CHATGPT AS TUTOR? A CASE STUDY ON COMPETITIVE PROGRAMMING

Juuso Rytilahti, & Erno Lokkila
Department of Computing, University of Turku (Finland)

Abstract

In this paper, we present a case study on how students utilize ChatGPT as a tutor for a short competitive programming course. The students were divided into two groups, one led by a teacher, and the other tutored by ChatGPT. The course was an intensive five-day course and both groups studied concurrently. Students could freely choose which group they participated in. The ChatGPT group was provided a guide on the basics of prompting, including approaches on how to generate the study material. Both groups were allowed to use any learning material, and only the teacher-led group excluded the use of generative AI tools. *Research questions:* In this study, we focus on the following questions: (1) How did students approach using ChatGPT as a tutor?; (2) Are there significant differences between students led by a teacher or by ChatGPT?; (3) How did students in both groups experience the teaching and did it correlate to the achieved learning results (exam)?
Methodology: The data consists of survey data, and final exam given to students (N=11). We also collected the discussion history of the ChatGPT group. The discussion history was divided into prompt-message pairs (N=340) and analyzed. The data was analyzed using mixed methods. The discussion history was analyzed using grounded theory. Statistical methods were used to find any correlation between initial skill level and learning as well as the tag distribution of the ChatGPT discussions.
Results: Differences were found between the two groups. Those with a higher initial skill level seemed to favor the ChatGPT group, whereas the less experienced chose the in-person teaching. Analysis of the ChatGPT discussion history showed mostly similar usage patterns across students. We present the distribution of tags used by the ChatGPT group. Additionally, we give insight on how to approach similar research settings in the future.
Impact: All around the world, students are already utilizing ChatGPT as a substitute for a teacher or a tutor. Our pilot study provides insight into how students approach utilizing ChatGPT as a tutor in a programming teaching setting. These preliminary results can be used to guide future research settings.

Keywords: *ChatGPT, programming education, case study, AI in education.*

1. Introduction

ChatGPT and other large language models (LLMs) have in a relatively short period grown in capabilities and are being used across different industries. Additionally, the good availability and performance of these tools have led students across the world to start using ChatGPT (and other LLMs) to aid them in their schoolwork. Although some of the potential use cases are nefarious, such as using ChatGPT to generate full answers to introductory programming courses, there are valid use cases as well. Students might use ChatGPT to help them understand a hard-to-grasp topic or try to replace incompetent tutors. They also might be studying independently or they might try to update outdated material with the help of these now easily available tools. This creates a need to understand how well ChatGPT can act as a tutor, tutorial, or substitute a tutor. This begs the question: can students identify potential caveats related to these approaches and what is the performance of the publicly available models related to in-person interactive learning experiences?

2. Related work

ChatGPT is a LLM published by OpenAI. LLMs models that have been trained with massive amounts of data. It can perform a variety of different tasks across multiple domains from code generation, and translation to text classification (Liu et al., 2023, Achiam et al., 2023). OpenAI has published GPT-3.5

and GPT-4-based ChatGPT models. GPT-4 performs better than GPT-3.5 in almost all tasks (Achiam et al., 2023). There are potential downsides. LLMs can and also will generate convincing-sounding false text, known as hallucinations (Maynez, Narayan, Bohnet, and McDonald 2020). It might also generate text that reinforces stereotypes. These potential risks should be kept in mind when using ChatGPT or other LLMs.

One of the challenges is also the art of prompting (prompt engineering). A prompt is the user's input to the model. Aligning the large language model to give relevant answers has always been a challenge, and prompts can greatly affect the performance of these models (Reynolds & McDonell, 2021). On the other hand, utilizing specific prompting techniques such as chain-of-thoughts (Wei et al. 2022) can improve the performance of the models. Chain-of-thoughts is a prompting technique where the model breaks down the problem into smaller separate steps, often increasing the model's performance on tasks requiring reasoning. Yilmaz and Yilmaz (2023) note that providing instructions for students on prompting is important because a prompt can greatly affect the quality of the output of the model. This can concretely be seen, for example, in some of the prompts written by the students in Qureshi (2023), where some of the provided example prompts of students lack clear instructions and do not give enough context for the model to provide a correct answer.

Yilmaz and Yilmaz (2023) noted in their conducted study that incorporating ChatGPT into programming courses seems beneficial and that incorporating it seems to enhance "students' self-confidence, learning motivation, and code-writing skills".

Qureshi (2023) conducted a quite similar study to ours. The study setting was that students were given programming tasks to solve within a strict time limit. In the study student students (N=24) were into two groups. The groups were composed of six pairs of students. The first group had no access to the internet (Group A) but they had access to the textbooks and notes of programming courses while the students in the second group (Group B) were given access to ChatGPT and were encouraged to use it. The ChatGPT group also had access to ChatGPT in the exam. Group B had a better overall score than group A. However, in group B the submitted code lowered the score due to its lack (to some degree) of accuracy and consistency.

3. Methodology

The setting in this study was a 5-day intensive course themed "Introduction to Competitive Programming" (3 ECTS). The course was held as an alternative to a voluntary "custom project" course. Students were free to choose their tutor (human or ChatGPT) freely. The ChatGPT group were asked to use GPT-3.5, but were allowed to use GPT-4 if they already had paid the required subscription fee themselves. The students in the contact person group were prohibited from using any LLMs on the course. The curriculum included four days of studying and a final exam on the fifth (final) day. The exam was supervised and neither group was allowed to use any LLM during the exam. The exam was the same for both groups and held for both groups at the same time.

Data collection: At the end of each the students answered the survey. At the end of the course, students were asked to e-mail the links to their chats with ChatGPT. These prompts were then collected and analyzed using grounded theory. After the exam, a small feedback session was held where students could discuss.

The ChatGPT group was given a small guide constructed by us that contained information about ChatGPT and prompting. The guide also contained instructions to generate study materials related to the course content. Students were instructed to generate materials following the workflow displayed in Table 1. The ChatGPT group could also use other freely available materials across the internet. The ChatGPT group was not given any other external materials besides the guide and the course description (learning objectives).

Table 1. Suggested workflow for study material generation in the guide given to students.

Workflow suggested to students	Additional notes for students
1. Generating a modular curriculum based on the course's description and learning goals.	Check that all of the learning goals are mentioned in the generated plan and that the plan is realistic.
2. Generating individual modules (study material) in a separate thread(s).	Remember the context limit.
3. Create exercises in a separate thread and include enough context (generated course material) to improve the exercise quality and align the ChatGPT with the set learning goals.	Remember to create programming exercises and essays.

4. Results

The more experienced students seemed to prefer to study independently using ChatGPT, as a one-tailed T-test showed a statistically significant difference (p=0.002) in the mean between the starting skill: ,0.3 and 1.0 respectively as measured by evaluating the coding question in the pre-course survey.

A one-tailed T-test revealed a statistical significance in the mean final test scores (maximum points 100) between the ChatGPT and non-ChatGPT groups: 65.4 and 79.1, with the ChatGPT-group performing better (p=.041). This result corroborates the fact that the more experienced students chose to study with ChatGPT.

The generated course syllabi were similar across the board. The in-person course syllabus developed by the teacher began with a day of introduction to competitive programming and revision of common data structures. The second day focused on problem-solving strategies, such as dynamic programming and divide-and-conquer. The third day was about recursion. The fourth day focused on graphs and graph problems. The syllabi created by ChatGPT for students always followed the same general structure as above, except for leaving out recursion and replacing it with coding practice.

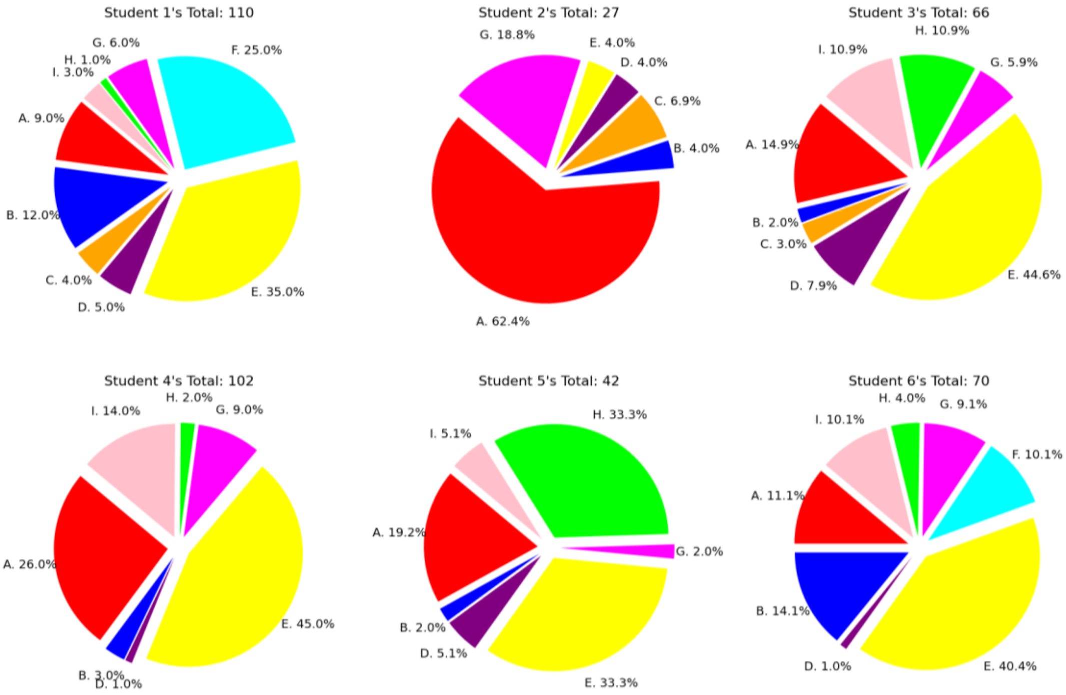
The distribution of the prompt-answer pairs can be seen in Figure 1. Due to a technical error, 39% (40 total) message pairs from student 1 were left untagged. As seen in the figure, there is quite a big difference in the number of question-answer pairs depending on the student. One prompt-answer pair could include multiple tags (although most of them had only 1 tag). Unfortunately, our number of students in the ChatGPT-led group (N=6) was too small to make solid assumptions based on this. The explanation of the tags can be seen in Table 2. The “ChatGPT explains” and” Information search” are partly aligned, and one could argue to present them together. However, we decided to separate them because the more complex questions can be answered directly by ChatGPT or other LLMs but searching using traditional search engines (e.g. Google) would not have yielded as good results.

In the feedback session after the exam, students noted that sometimes ChatGPT explained terms with wrong definitions. Also, students mentioned that exercises generated by ChatGPT were sometimes too easy. During that students confirmed that topics were mainly the same as the teacher-led group.

Table 2. Tag explanations with corresponding letter and color.

🟥	A	Exercise generation	Generation of exercises, e.g., coding questions, multiple choice questions or essays.
🟦	B	Exercise evaluation	Evaluation of the answers provided by students.
🟧	C	Exercise variation	Generating a variation, often asking for more harder exercises of already generated ones.
🟪	D	Other	Not fitting to any other category (e.g. thanking ChatGPT).
🟨	E	ChatGPT Explains	ChatGPT explains a complex topic.
🟦	F	Continue prompt	Asking ChatGPT to continue generating more material. (often used with study material generation, tag “ChatGPT explains”).
🟪	G	Study planning	Often the creating the modular curriculum.
🟩	H	Information search	A simpler question, often the answer could have been gained with a simple Google search.
⬜	I	Example answer	ChatGPT produced an example answer to an exercise.

Figure 1. The distribution of the tags. Label explanations can be seen in Table 2. The distributions of 0 have been excluded from the figure for increased readability. Total is the total tag count.



5. Discussion

Students, especially students learning programming are already learning on courses that are held fully remote and which require or even have very little input from the course personnel. However, students can complete these courses and achieve the required learning goals. So active teacher-student interaction is not necessarily an obligatory requirement, although having that interaction is almost always beneficial. In certain scenarios, ChatGPT can be, although not perfect, maybe an adequate substitute for a tutor. This of course requires that students remain vigilant and are aware of the limitations and risks of the technology.

This naturally raises the question of the quality of the generated materials and to which degree we need human intervention in the programming courses. The biggest challenge and limiting factor of LLMs is hallucination. Students need to be actively reminded that LLMs sometimes will generate convincing-sounding answers that are completely fabricated. This also sets requirements for which topics this kind of study setting can be utilized. Learning competitive programming is almost the perfect subject for this setting because it requires students to use their logical thinking and it's more about applying skills than learning blindly theoretical background. This forces students to actively challenge the answers generated by ChatGPT instead of relying on them as the ground truth.

ChatGPT's capability to act as a tutor is limited both by students too novice to validate the output as well as the model's inability to produce correct answers reliably. For more novice students the teachers should give more strict guidelines, defining the allowed and prohibited use cases clearly to students, along with the information about why restrictions were set. However, if students have enough base knowledge of the topic and the tool (LLM), students can be given more freedom. Students should be actively reminded that ChatGPT behaves more akin to a more experienced classmate than an expert in the field.

We noticed that the students who were working full- or part-time (presumably in programming-related jobs) or had the highest skill tended to select the ChatGPT group. We suspect this was due to the flexibility that independent learning can offer. It might also be that those unsure of their abilities tended to favor the possibility of interaction with real teachers. Additionally, at least in these early stages of technology, utilizing ChatGPT to its full potential requires a lot from the student, especially in terms of willingness to learn a new tool, its proficient use and restrictions.

However, the potential of the technology is still quite large. If the hallucination problem is ever solved (or students learn to double-check information), and LLM context sizes keep increasing, many possibilities could emerge. For example, teachers could focus on defining learning goals and selecting source materials, while the LLM could adapt the provided source material to best fit the learning goals the course's teacher set. However the current technology still clearly lacks the capabilities to achieve this.

We deem that giving students an alternative to ChatGPT (or other LLM) is also important. In our study, this was also emphasized in the guide given to students, in the very first sentence of the first chapter of the guide. This gives students agency and freedom to find the most efficient way to study. Although this study setting was a bit extreme, it's important to remember that students are already doing similar activities across the world.

6. Conclusion

The limitations and threats associated with ChatGPT limit the possibilities of its usage. However, we can fairly confidently say that students across the world will use ChatGPT as a tutor or teacher's substitute. The 39% (40 total) of message pairs left untagged from student 1 due to a technical error changed the distribution of tags of student 1 but did not otherwise affect our result.

Using it as an autonomous tutor should be considered carefully. The risks associated with using ChatGPT as a tutor should be recognized and carefully considered. The selected topics should, at least in this early stage of the said technology, aim to mitigate the risk of causing potential knowledge gaps in students' knowledge caused by hallucinations of the LLM. This means that students should have enough knowledge of the taught topic, as well as exclude topics where undetected hallucinations can cause harm. This also means that ChatGPT should be utilized in courses where students mainly do not try to gain new theoretical background, but instead try to apply already learned skills with possibly minor additions to their base knowledge. In a programming context, this could be e.g., course about competitive programming or amplifying the student's productivity on a final project on a programming course therefore increasing the possible scope of the project.

For future research on similar settings, we recommend dividing students with a pre-test, similar to Qureshi (2023), to avoid skewing more proficient students in the ChatGPT group. We also concur with Yilmaz and Yilmaz (2023) that students should also be given guidance on prompting and basic knowledge of LLMs before giving students access to using them in a teaching setting. In addition, we propose that separating students into three different groups could provide interesting results: a fully independent that can utilize LLMs, a fully independent without access to LLMs, and an in-person held contact group. Also, we noticed that students might be unsure about whether the content generated by ChatGPT is good enough, so confirming that the model is producing good quality content with students at regular intervals would probably be beneficial. Keeping in mind the potential and challenges related to LLMs, as mentioned in the discussion, is vital to producing good, ethical research on this topic.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., … et al. (2023). *GPT-4 Technical Report.* arXiv preprint arXiv:2303.08774

Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J.,… et al. (2023). Summary of ChatGPT-related research and perspective towards the future of large language models. *Meta-Radiology, 1*(2), 100017.

Maynez, J., Narayan, S., Bohnet, B., & McDonald, R. (2020). *On faithfulness and factuality in abstractive summarization*. arXiv preprint arXiv:2005.00661

Qureshi, B. (2023). *Exploring the use of ChatGPT as a tool for learning and assessment in undergraduate computer science curriculum: Opportunities and challenges*. arXiv preprint arXiv:2304.11214

Reynolds, L., & McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. In *CHI EA '21 Extended abstracts of the 2021 CHI Conference on human factors in computing systems, 314* (pp. 1–7).

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., … et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems, 35*, 24824-24837.

Yilmaz, R., & Yilmaz, F. G. K. (2023). Augmented intelligence in programming learning: Examining student views on the use of chatgpt for programming learning. *Computers in Human Behavior: Artificial Humans, 1*(2), 100005.