Explaining Code Examples in Introductory Programming Courses: LLM vs Humans

Reviewer 1

Summary of Contributions: The paper introduced an evaluation of whether ChatGPT can be used for code explanation creation. The authors included the comparisons of the generated ChatGPT explanations with the expert and student explanations and showed that ChatGPT can create explanations with decent quality but still need to be more readable.

Strengths:

- Good topic that addresses a great direction leveraging LLMs for CSEd tasks, and the authors are very clear about the educational implications of the work, as they included how code explanations can be used in intelligent tutoring systems.
- Evaluations are systematic. The authors used not only simple metrics but also metrics from different perspectives, including readability, lexical density, etc., for a more complete evaluation.

Weaknesses:

- One downside of the work, which the paper has acknowledged, is that the students' explanations seem to have a very different distribution from the ChatGPT and expert ones. The readers will benefit more from the information about how such explanations are generated by students and in what scenarios they are asked to write the explanations. Some examples can be more intuitive if the space allows.

Thank you, we have now explained this under “Dataset Collection” under “Student Explanations”

“For example, for the line of code \textit{``private int y''} a student participating in the study explained \textit{``Creates a new object class called Point''}.”

Reviewer 2

I have a few comments:

1. What criteria did you use to choose the 4 questions in your analysis? Are these diverse in the programming concepts/ difficulty level they cover?

Thank you, we have now explained this in the “Dataset Collection” section.

“PCEX example exploration system….These four programs were selected in the increasing order of difficulty, such that the simpler program involved array search and print statements, while the hardest program focused more on introductory object oriented programming principles….”

2. I believe the goal is to make LLMs generate human-like explanations. In this context, it is understandable why these systems score low on readability metrics. However, with the similarity metric you mention that the simple prompting strategy aligns more closely with experts. Do you know why this happens?

Thank you, we have added the following paragraph under “similarity metrics” under “Results” section. We provide more details on this in the Appendix

Added the text “The high semantic alignment between the expert explanation and the simple prompt can be attributed to the fact that the advanced prompt provides explanations for the significance of each line of code. In contrast, the expert explanation lacks consistent explanations regarding the importance of each code line(see Appendix B)” as an explanation for the semantic alignment between the explanations generated by expert and simple prompting strategy

3. The authors claim that extended prompt produces the "best" results. However, the similarity metric results do not agree with this claim. It would be good to explain more.

Thank you, we have updated this text to say the following under the “Extended Prompt” under “Dataset Collection”.

“To obtain the most elaborate ChatGPT explanations, we used \emph{Extended prompt}, which requested ChatGPT to further enhance the explanations generated by the Advanced prompt (Iteration \#2 in Figure…, with focus on consistency and coverage of the generated content.”

Rating: 8: Top 50% of accepted papers, clear accept
Confidence: 4: The reviewer is confident but not absolutely certain that the evaluation is correct