Could ChatGPT Be Used for Reviewing Learnersourced Exercises?

Nea Pirttinen
University of Helsinki
Helsinki, Finland
nea.pirttinen@helsinki.fi

Juho Leinonen
Aalto University
Espoo, Finland
juho.2.leinonen@aalto.fi

ABSTRACT

Large language models and tools based on large language models such as ChatGPT have received intense attention in the past year in computing education. In this work, we explore whether ChatGPT could be used to review learnersourced exercises. One of the major downsides of learnersourcing is the dubious quality of the created content, leading to many systems using peer review for curating the content. Our results suggest that ChatGPT is not yet ready for this task.

CCS CONCEPTS

• **Social and professional topics** → *Computing education*; • **Information systems** → *Crowdsourcing*.

KEYWORDS

crowdsourcing, learnersourcing, reviews, ChatGPT, large language models, LLMs, generative AI

ACM Reference Format:
Nea Pirttinen and Juho Leinonen. 2023. Could ChatGPT Be Used for Reviewing Learnersourced Exercises?. In *23rd Koli Calling International Conference on Computing Education Research (Koli Calling '23), November 13–18, 2023, Koli, Finland.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3631802.3631845

1 INTRODUCTION AND METHOD

In the past year, the release of ChatGPT has popularized large language models (LLMs), with computing education being no exception [1, 3]. In computing education, LLMs have been found to solve introductory programming problems better than the average student [4, 22], to generate code explanations [21] that students find useful [13] and that are better than explanations created by their peers [10], and there are preliminary results suggesting they can also give feedback on student programs [7, 8, 11, 15]. In this work, we explore how well ChatGPT could be used for reviewing learnersourced programming exercises. In learnersourcing, students contribute their expertise in the creation of course content, for example, multiple-choice questions [2], programming exercises [18], or SQL exercises [12, 17]. However, one downside of learnersourcing is that not all of the created content is of high quality [5, 19, 20]. Thus, many learnersourcing systems use peer review to filter out low quality content [6, 16]. Being able to use LLMs to effectively review the created content would leave more time for students for content creation instead of reviewing content created by their peers.

We utilize a dataset of learnersourced exercises collected from an introductory programming course organized at the University of Helsinki [20]. In the course, students used a learnersourcing system to create programming exercises. For each exercise, they crafted a problem description, a model solution, a code template, and test cases. These exercises were then reviewed by their peers. We randomly sampled 50 of the exercises for the purposes of this study, which were reviewed by two programming instructors. The following Likert-scale (strongly disagree 1 – 5 strongly agree) rubric was used for the reviews: (1) The model solution corresponds to the exercise description; (2) The code is clean; (3) The model solution and the code template are separated correctly; (4) The exercise is creative; (5) The exercise is suitably difficult; (6) The exercise description corresponds to the instructions; (7) The exercise description is clear; (8) The test cases are reasonable; (9) The test coverage is on the expected level; (10) The test names are descriptive.

To generate ChatGPT reviews, we gave it the rubric, the instructions given to students about the creation of exercises, and the exercise created by the student. We used the free version of ChatGPT (August 3 Version) which is powered by GPT-3.5 (chat.openai.com). To explore how well ChatGPT can review the student-created exercises, we compare review scores between the instructors, the students, and ChatGPT using Mann-Whitney U test as the data is ordinal. In addition, we calculate Krippendorff's alphas between the three raters (instructors, students, ChatGPT) for whether the exercise could be included in the course using two quality thresholds: an average rating of over 3 and an average rating of over 4. For this analysis, for both the students and the instructors, we consider the average scores given by the students and the instructors (i.e., we average the scores given by the raters).

2 RESULTS AND DISCUSSION

Mann-Whitney U tests reveal no statistically significant differences between review score averages of the instructors and the students ($U = 1154.0, p = 0.25$), the instructors and ChatGPT ($U = 1222.0, p = 0.42$), or the students and ChatGPT ($U = 1073.0, p = 0.11$). This suggests that the overall review scores are similar between all three sources. We also looked into the averages and standard deviations of the scores and found that students had the highest average ($\mu = 3.84, \sigma = 0.71$), followed by ChatGPT ($\mu = 3.77, \sigma = 0.59$), with the instructors having the lowest average ($\mu = 3.64, \sigma = 0.98$). Using the evaluation guidelines provided by Krippendorff [9], the results of reliability calculations between the students and ChatGPT can be used to draw tentative conclusions ($\alpha > 0.667$) with the quality threshold of average rating of over 3, while the other results are too unreliable to draw any conclusions. For the threshold of greater than 3, $\alpha > 0.73$ between the students

Koli Calling '23, November 13–18, 2023, Koli, Finland

ACM ISBN 979-8-4007-1653-9/23/11.
https://doi.org/10.1145/3631802.3631845

and ChatGPT, and $\alpha > 0.56$ between instructors and ChatGPT. For the threshold of greater than 4, $\alpha > 0.11$ between students and ChatGPT, and $\alpha > 0.22$ between instructors and ChatGPT.

A cursory look into the ratings given by ChatGPT gives some ideas for these differences. ChatGPT did not give the full 5 points to any of the rated exercises for the rubric item about the creativity of the exercise. Additionally, the instructors tended to give mostly ones or fives for the separation of template and model solution, reasoning that these two are either correctly separated or not, with few edge cases. ChatGPT ratings had more variance for this rubric item, notably with only one exercise receiving one point. Furthermore, ChatGPT was not given the example exercise provided in the course materials, which means that it was not able to compare students' exercises to the example. This lowered some exercises' scores in peer and instructor review, at least for the creativity of the exercise. Lastly, we found that ChatGPT was a more lenient grader for low quality exerciess compared to the instructors and the students – in many cases, where humans had rated an exercise very low (< 2), ChatGPT rated them higher, which supports earlier findings by Moore et al. [14]. This might explain the lower standard deviation observed for ChatGPT ratings. Altogether, our findings suggest that ChatGPT (specifically, GPT-3.5) is not yet ready to replace peer review of learnersourced content.

In our future work, we are interested in studying the use of ChatGPT for reviewing student work more systematically. For example, we are looking into the reliability of the reviews, i.e., whether they remain similar if the LLM is prompted multiple times. In addition, we are looking into prompt engineering strategies for creating effective reviews using LLMs. In this work, we found that any textual feedback given by ChatGPT rarely provided additional information beyond the numerical scores, but it is interesting to analyze if different prompts would lead to better textual content.

ACKNOWLEDGMENTS

This research was supported by the Research Council of Finland (Academy Research Fellow grant number 356114) and by the Jenny and Antti Wihuri Foundation.

REFERENCES

[1] Paul Denny, Brett A Becker, Juho Leinonen, and James Prather. 2023. Chat Overflow: Artificially Intelligent Models for Computing Education-renAIssance or apocAIypse?. In *Proc of the 2023 Conf. on Innovation and Technology in Computer Science Education V. 1*. 3–4.

[2] Paul Denny, John Hamer, Andrew Luxton-Reilly, and Helen Purchase. 2008. PeerWise: Students Sharing Their Multiple Choice Questions. In *Proc. of the Fourth Int. Workshop on Computing Education Research*. 51–58.

[3] Paul Denny, James Prather, Brett A Becker, James Finnie-Ansley, Arto Hellas, Juho Leinonen, Andrew Luxton-Reilly, Brent N Reeves, Eddie Antonio Santos, and Sami Sarsa. 2023. Computing Education in the Era of Generative AI. *Commun. ACM* (2023).

[4] James Finnie-Ansley, Paul Denny, Brett A Becker, Andrew Luxton-Reilly, and James Prather. 2022. The robots are coming: Exploring the implications of openai codex on introductory programming. In *Proc. of the 24th Australasian Computing Education Conf.* 10–19.

[5] John Hamer, Quintin Cutts, Jana Jackova, Andrew Luxton-Reilly, Robert McCartney, Helen Purchase, Charles Riedesel, Mara Saeli, Kate Sanders, and Judithe Sheard. 2008. Contributing student pedagogy. *ACM SIGCSE Bulletin* 40, 4 (2008), 194–212.

[6] John Hamer, Andrew Luxton-Reilly, Helen C. Purchase, and Judithe Sheard. 2011. Tools for "Contributing Student Learning". *ACM Inroads* 2, 2 (2011), 78–91.

[7] Arto Hellas, Juho Leinonen, Sami Sarsa, Charles Koutcheme, Lilja Kujanpää, and Juha Sorva. 2023. Exploring the Responses of Large Language Models to Beginner Programmers' Help Requests. In *Proc. of the 2023 ACM Conf. on Int. Computing Education Research - Vol. 1*. ACM, 93–105.

[8] Natalie Kiesler, Dominic Lohr, and Hieke Keuning. 2023. Exploring the Potential of Large Language Models to Generate Formative Programming Feedback. *arXiv preprint arXiv:2309.00029* (2023).

[9] Klaus Krippendorff. 2004. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, Inc.

[10] Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 2023. Comparing Code Explanations Created by Students and Large Language Models. In *Proc. of the 2023 Conf. on Innovation and Technology in Computer Science Education V. 1*. ACM, 124–130.

[11] Juho Leinonen, Arto Hellas, Sami Sarsa, Brent Reeves, Paul Denny, James Prather, and Brett A Becker. 2023. Using large language models to enhance programming error messages. In *Proc. of the 54th ACM Technical Symp. on Computer Science Education V. 1*. 563–569.

[12] Juho Leinonen, Nea Pirttinen, and Arto Hellas. 2020. Crowdsourcing Content Creation for SQL Practice. In *Proc. of the 2020 ACM Conf. on Innovation and Technology in Computer Science Education*. 349–355.

[13] Stephen MacNeil, Andrew Tran, Arto Hellas, Joanne Kim, Sami Sarsa, Paul Denny, Seth Bernstein, and Juho Leinonen. 2023. Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book. In *Proc. of the 54th ACM Technical Symposium on Computer Science Education V. 1*. 931–937.

[14] Steven Moore, Huy A. Nguyen, Norman Bier, Tanvi Domadia, and John Stamper. 2022. Assessing the Quality of Student-Generated Short Answer Questions Using GPT-3. In *Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption: 17th European Conf. on Technology Enhanced Learning, EC-TEL 2022, Toulouse, France, September 12–16, 2022, Proc.* (Toulouse, France). Springer-Verlag, Berlin, Heidelberg, 243–257. https://doi.org/10.1007/978-3-031-16290-9_18

[15] Maciej Pankiewicz and Ryan S Baker. 2023. Large Language Models (GPT) for Automating Feedback on Programming Assignments. *arXiv preprint arXiv:2307.00150* (2023).

[16] Nea Pirttinen, Paul Denny, Arto Hellas, and Juho Leinonen. 2023. Lessons Learned From Four Computing Education Crowdsourcing Systems. *IEEE Access* 11 (2023), 22982–22992.

[17] Nea Pirttinen, Arto Hellas, and Juho Leinonen. 2023. Experiences from Learnersourcing SQL Exercises: Do They Cover Course Topics and Do Students Use Them?. In *Proc. of the 25th Australasian Computing Education Conf.* ACM, 123–131.

[18] Nea Pirttinen, Vilma Kangas, Irene Nikkarinen, Henrik Nygren, Juho Leinonen, and Arto Hellas. 2018. Crowdsourcing Programming Assignments with CrowdSorcerer. In *Proc. of the 23rd Annual ACM Conf. on Innovation and Technology in Computer Science Education*. 326–331.

[19] Nea Pirttinen, Vilma Kangas, Henrik Nygren, Juho Leinonen, and Arto Hellas. 2018. Analysis of Students' Peer Reviews to Crowdsourced Programming Assignments. In *Proc. of the 18th Koli Calling Int. Conf. on Computing Education Research*. 1–5.

[20] Nea Pirttinen and Juho Leinonen. 2022. Can Students Review Their Peers? Comparison of Peer and Instructor Reviews. In *Proc. of the 27th ACM Conf. on on Innovation and Technology in Computer Science Education Vol. 1*. ACM, 12–18.

[21] Sami Sarsa, Paul Denny, Arto Hellas, and Juho Leinonen. 2022. Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models. In *Proc. of the 2022 ACM Conf. on Int. Computing Education Research-Vol. 1*. 27–43.

[22] Jaromir Savelka, Arav Agarwal, Marshall An, Chris Bogart, and Majd Sakr. 2023. Thrilled by Your Progress! Large Language Models (GPT-4) No Longer Struggle to Pass Assessments in Higher Education Programming Courses. In *Proc. of the 2023 ACM Conf. on Int. Computing Education Research - Vol. 1*. ACM, 78–92.