Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025 ASEE
Paper ID #48335

BOARD # 97: WIP: The Effectiveness of Rubric-Based LLM Feedback for Programming Assessments

Mr. Joel Nirupam Raj
Ashwath Muppa, Thomas Jefferson High School for Science and Technology
Rhea Nirmal
Teo W. Kamath
Dr. Mihai Boicu, George Mason University

Mihai Boicu, Ph.D., is Assistant Professor of Information Technology at George Mason University. He is an expert in artificial intelligence, structured analytical methods, probabilistic reasoning, evidence-based reasoning, personalized education, active learning with technology, crowd-sourcing, and collective intelligence. He is the main software architect of the Disciple agent development platform and coordinates the software development of various analytical tools used in IC and education. He has over 120 publications, including 2 books and 3 textbooks. He has received the Innovative Application Award from the American Association for Artificial Intelligence, and several certificates of appreciation from the U.S. Army War College and the Air War College. He is a GMU Teacher of Distinction.

Achyut Dipukumar
Aarush Laddha

WIP: The Effectiveness of Rubric-Based LLM Feedback for Programming Assessments

Abstract

Automated feedback systems are becoming more important in programming education as class sizes grow, and instructor resources are limited. Recent advances in large language models (LLMs) offer a practical way for educators to provide structured feedback for students on various assignments. A pre-experiment involved four student researchers solving Project Euler problems and showed an average improvement of 17.5 points on a scoring rubric out of 100 after code revision using feedback generated from Claude 3.5 Sonnet. There were also notable gains in time complexity, efficiency, and edge case handling, with percentage increases 24.45%, 22.59%, and 22%, respectively. Building on these results, we designed a classroom-based experiment involving students across various programming courses. Students will be divided into control (human feedback) and treatment (LLM feedback) groups, with feedback graded with a 14-criteria rubric. Claude 3.7 Sonnet will be the LLM used in this study, as it is the latest model released by Anthropic. The study evaluates both quantitative score improvements and students' perceptions of feedback quality. The results of this study aim to inform the integration of LLMs into education assessment practices.

Introduction

The rise of large language models (LLMs) has opened new possibilities for automated feedback in education, particularly in programming courses where timely, detailed evaluation is needed for student development. While these models show promise for humanities and science assignments, their effectiveness for mathematical programming tasks remains uncertain in actual undergraduate classrooms. The potential for LLMs to assist with programming education comes at a critical time, as growing class sizes and increasing workload demands challenge instructors' ability to provide personalized feedback.

Current research presents conflicting evidence about LLMs' capabilities in educational settings. Some studies demonstrate their ability to match human grading accuracy for structured problems, while others reveal limitations in handling complex programming concepts or providing contextual suggestions. This disagreement shows the need for careful evaluation of LLMs' role in computing education, particularly for mathematical programming where solutions often involve multiple valid approaches with subtle tradeoffs between efficiency and readability.

The promise of LLM-assisted feedback lies in its potential to combine scalability with quality. For programming instructors, such tools could alleviate grading burdens while maintaining rigorous standards. For students, they could offer immediate, detailed feedback that supports iterative improvement, which is a key component of learning to program effectively. However, realizing these benefits requires understanding where and how LLM feedback can complement human expertise in authentic educational contexts.

This study examines the role of LLM-generated feedback in undergraduate computing education through the lens of mathematical programming tasks. By focusing on real classroom applications

rather than laboratory conditions, the research aims to provide practical insights for educators considering these tools. The findings will contribute to broader discussions about technology-enhanced learning and the evolving relationship between artificial intelligence and human instruction in technical disciplines.

Literature Research

Recent advances in LLMs have shown their potential to transform educational settings, particularly in programming courses where timely, detailed feedback is important. Fagbohun et al. [1] states that LLMs can automate grading with personalized feedback but that they still require careful handling of biases combined with human supervision to ensure that LLMs are fair and efficient and to reduce the occurrence of ethical risks like depersonalization. These are all important factors to consider when using Claude 3.7 Sonnet in our experiment.

Further supporting the role of structured guidance, Mok et al. [2] evaluates AI grading from LLMs such as Claude 3.5 Sonnet and GPT-4o for undergraduate physics problems. It shows that while AI grading tends to output mathematical errors compared to human graders, it improves significantly when provided with a mark scheme (like the rubric we used in our study).

For programming specific applications, Yousef et al. [3] introduce the BeGrading Framework, which trains LLMs on 3,470 real student programming assignments and 374 synthetic examples generated by GPT-4 and refined via Gemini. The refinement process uses iterative validation, where GPT-4 grades code first, then Gemini acts as a "sanity check" to flag errors, prompting GPT-4 to re-evaluate.

In [4] we performed a detailed literature survey identifying the following global challenges: sometimes LLMs struggle to detect unseen human errors and are not capable of proper mathematical understanding (e.g., [5]); sometimes wrong answers may be regarded as helpful which indicate the need for supervision (e.g., [6]); the availability of training data is crucial for LLMs, as specialized programming tasks, as concurrent programming received around 50% accuracy [7] compared with traditional programming in which LLMs are performing very well. Related to feedback generation Escalante et al. [8] identified that LLM can provide transparent and detailed feedback and allow students to drill down on issues, but also students appreciated the opportunity for real-time interaction during human feedback. Cohn et al. [9] identified that breaking down the feedback in focus aspects improves accuracy. Related to evaluating and scoring of assignments, Koutcheme et al. [10] find that LLMs are useful for scoring by using prompting and rubrics but they need human supervision, and Stahl et al. [11] find LLMs useful to generate prompts to guide student essay writing and are benefiting automated essay scoring.

Pre-experiments

In [12] we performed two pre-experiments, to compare the consistency and accuracy of the following LLMs: Claude 3.5 Sonnet, Microsoft CoPilot, Meta Llama 3, Google Gemini, and Zapier. The first pre-experiment analyzed how relevant the feedback was to the prompt, and the student's code, and we identified Claude, Microsoft CoPilot, and Meta Llama 3 to provide the best answers. In the next pre-experiment, we analyzed the feedback provided by these selected LLMs

to problems of different difficulty (easy, medium and hard) from Project Euler submitted by 3 researchers and repeated five times for consistency checking. The minimum standard deviation for the scores was 2.668 obtained by Claude 3.5 Sonnet much lower than Microsoft Copilot (4.525) and Meta Llama 3 (7.127). Because of this higher consistency and also the balanced and accurate feedback, Claude was selected for the next experiment.

As described in [4], the purpose of the next phase was to determine the proficiency of Claude Sonnet 3.5 in providing feedback for mathematical programming problems and the main categories in which the improvement occurred. In the experiment, four researchers completed 5 Project Euler problems of varying difficulty (see *Appendix A. Solved Problems* for links to all the problems used [13]). The researchers participating in the experiment were 4 students with various knowledge of programming that are also authors of this paper. We first performed this in-house experiment to check our intuition, gain the knowledge to design the experiment, apply for the IRB and perform the proposed experiment with typical college students, as will be described in the final paper.

Researchers solved the problems in Python in maximum 40 minutes. Sample code initially submitted by researcher 1 is provided in *Appendix B Initial Python Solution Sample* [13]. This is the simplest problem and is provided as an example and has around 42 lines including code and comments. These programs were submitted to Claude Sonnet 3.5 to receive a grade and feedback in the following categories: "Correctness," "Efficiency," "Data Structures," "Code Readability," and "Testing." The prompt used to call Claude Sonnet 3.5 is provided in *Appendix D. Prompt* specifying the grading rubric included in *Appendix C. Rubric* (provided as a PDF document with the prompt). The prompt is very specific with respect to how the feedback must be provided, following the rubric specifications and has 72 lines. The sample feedback received by Researcher 1 is provided in *Appendix E. Sample Initial Feedback Received* with a global score of 75/100 and containing 74 lines (well above typical instructor feedback length).

Using the feedback, the researchers revised their code in another 40 minutes session. The improved solution of Researcher 1 is presented in *Appendix F. Improved Solution Sample.* It was graded by Claude Sonnet 3.5 using the same prompt and rubric and obtained new feedback with the global score of 85/100, as shown in *Appendix G. Feedback for Improved Solution.*

The feedback provided by Claude Sonnet 3.5 included a score for each category above and an overall score. The numeric scores were reviewed and considered reasonable by the researchers. The results showed that there was a mean score improvement of 17.5 points after the code revision, with the highest percentage increases in time complexity (+25.45%), efficiency (+22.59%), and edge case handling (+22%).

Hypotheses

This study evaluates whether rubric-structured feedback from Claude 3.5 Sonnet can achieve comparable student learning outcomes to human-generated feedback in undergraduate programming courses. We hypothesize that LLM-generated feedback will produce similar improvements in code quality (as measured by rubric scores) while significantly reducing grading time, but that human feedback may remain superior for complex conceptual guidance. Student perceptions of feedback quality, granularity, and usefulness will also differ between the two

conditions, with LLM feedback favored for immediacy and human feedback preferred for nuanced problem-solving advice.

Experiment Design

The study will recruit approximately 20 undergraduate and graduate students enrolled in several computing related courses at George Mason University. Participants will be recruited through course announcements and voluntary sign-up. The sample will include students across undergraduate years (sophomore to master) to capture varying skill levels. Exclusion criteria will be limited to students under 18 years old or those unwilling to consent to data usage. Participant demographics including prior programming experience, GPA range, and course enrollment status will be collected to enable subgroup analysis.

The study will use four main components: (1) A set of validated mathematical programming problems adapted from Project Euler and adjusted for the level of the course, (2) A detailed 14-criteria grading rubric covering correctness, efficiency, data structures usage, code readability, and testing, (3) The Claude 3.7 Sonnet accessed through the website, and (4) A custom grading interface that randomizes feedback source assignment and tracks grader interactions. All student code submissions will be processed through a standardized pipeline that anonymizes identifiers before analysis. The infrastructure includes automated test cases for objective correctness verification and manual grading protocols for subjective criteria.

For the LLM condition, each submission will be processed through a carefully engineered prompt sequence. First, the raw code undergoes static analysis for syntax and structure. Then, the full prompt (including problem description, rubric, and student code) is submitted to Claude 3.7 Sonnet. Human graders, which will be student researchers, will grade using the same rubric. All feedback, whether LLM or human-generated, will undergo quality control checks by the research team before delivery to students. A feedback template ensures uniform formatting across conditions.

The experiment will run over four weeks of the academic semester. In Week 1, students complete Assignment A and receive randomized feedback (LLM or human). Week 2 involves code revision and resubmission. This pattern repeats Assignment B in Weeks 3-4, with feedback sources crossed over (students receiving human feedback first get LLM feedback second, and vice versa). Each assignment has a 7-day completion window with fixed deadlines. Students interact with the feedback system through their existing course management platform (Blackboard), maintaining typical workflow patterns. The procedure includes post-study surveys to measure feedback perception.

Analysis and Result Interpretation

The initial and revised grades as well as the time taken by the human grader will be recorded for statistical analysis. The students will also complete a questionnaire to rate the feedback they received. The data analysis segment of this research experiment will compare the grade improvements between the two groups. Also, a semantic and qualitative analysis of the feedback will be analyzed using natural language processing (NLP) methods. The overall significance of these findings will be assessed by using t-tests, ANOVA analysis, or Kruskal-Wallis tests. The

study will also visualize grade distributions, average grade improvements, and correlations between initial and final grades.

This experiment is a work in progress, and it is in final stage of completion. The results will be analyzed and presented at the conference.

Limitations and Future Work

While this study provides valuable insights into LLM-generated feedback, it has some limitations. The classroom setting introduced variables like student motivation and prior experience that are hard to control, and the study only evaluated one LLM (Claude 3.7 Sonnet). Future work should test more models, explore hybrid human-AI feedback systems, and investigate long-term impacts on learning. We also plan to develop better tools to help students interpret and apply LLM suggestions effectively.

References

[1] Fagbohun O, Iduwe NP, Abdullahi M, Ifaturoti A, Nwanna OM. Beyond Traditional Assessment: Exploring the Impact of Large Language Models on Grading Practices. J Artif Intell Mach Learn & Data Sci 2024, 2(1), 1-8. DOI: doi.org/10.51219/JAIMLD/oluwole-fagbohun/19

[2] Morris, W., Holmes, L., Choi, J.S. et al. Automated Scoring of Constructed Response Items in Math Assessment Using Large Language Models. Int J Artif Intell Educ (2024). https://doi.org/10.1007/s40593-024-00418-w

[3] Mok, R.; Campanelli, M.; Datta, A.; Gupta, A.; Hickman, R.; Osthus, F.; Zwart, P. Using AI Large Language Models for Grading in Education: A Hands-On Test for Physics. arXiv preprint arXiv:2411.13685, 2024

[4] J. Raj, A. Muppa; A. Dipukumar; R. Nirmal; A. Laddha; T. Kamath, S. Hong, M. Potla and M. Boicu. "Quantitative Analysis of Rubric-based Feedback Received From Claude 3.5 Sonnet on Mathematical Programming Problems," 2024 IEEE MIT Undergraduate Research Technology Conference (URTC), Cambridge, MA, USA, 2024, pp. 1-5,
doi: 10.1109/URTC65039.2024.10937532.

[5] H. McNichols, J. Lee, S. Fancsali, S. Ritter, and A. Lan, “Can Large Language Models Replicate ITS Feedback on Open-Ended Math Questions?,” 2024, arXiv: 2405.06414

[6] K. M. Collins et al., “Evaluating Language Models for Mathematics through Interactions,” Proceedings of the National Academy of Sciences of the United States of America, vol. 121, no. 24, Jun. 2024, doi: https://doi.org/10.1073/pnas.2318124121.

[7] I. Estévez-Ayres, P. Callejo, M. A. Hombrados-Herrera, C. Alario-Hoyos, and C. D. Kloos, “Evaluation of LLM Tools for Feedback Generation in a Course on Concurrent Programming,” International journal of artificial intelligence in education, May 2024, doi: https://doi.org/10.1007/s40593-024-00406-0.

[8] J. Escalante, A. Pack, and A. Barrett, “AI-generated Feedback on writing: Insights into Efficacy and ENL Student Preference,” International Journal of Educational Technology in Higher Education, vol. 20, no. 1, Oct. 2023, doi: https://doi.org/10.1186/s41239-023-00425-2.

[9] C. Cohn, N. Hutchins, T. Le, and G. Biswas, “A Chain-of-Thought Prompting Approach with LLMs for Evaluating Students’ Formative Assessment Responses in Science,” in AAAI Conference on Artificial Intelligence, Mar. 2024.

[10] C. Koutcheme, N. Dainese, S. Sarsa, A. Hellas, J.Leinonen, P.D.L. Koutcheme “Open Source Language Models Can Provide Feedback: Evaluating LLMs’ Ability to Help Students Using GPT-4-As-A-Judge," 2024, arXiv: 2405.05253

[11] M. Stahl, L. Biermann, A. Nehring, H. Wachsmuth “Exploring LLM Prompting Strategies for Joint Essay Scoring and Feedback Generation,” 2024, arXiv: 2404.15845

[12] Joel Raj, Ashwath Muppa, Achyut Dipukumar, Rhea Nirmal, Aarush Laddha, Teo Kamath, Sophie Hong, Meghana Potla, Mihai Boicu. Quantitative analysis of feedback received from Claude 3.5 Sonnet on mathematical programming problems using a multi-dimensional rubric framework. Journal of Students Scientists’ Research, Vol 6, 2024, George Mason University, Fairfax, Virginia https://doi.org/10.13021/jssr.2024

[13] J. Raj, A. Muppa; A. Dipukumar; R. Nirmal; A. Laddha; T. Kamath, S. Hong, M. Potla and M. Boicu. Research Appendices for Claude 3.5 Feedback Experiment. George Mason University, Fairfax, Virginia, 2024.
https://drive.google.com/drive/folders/1OkQt2yAuklNEONVyYLN6F0qFOrR3I8SZ?usp=share_link