Motivating Learning in the Face of Generative Artificial Intelligence

Wilson, Sara E.
Mechanical Engineering, University of Kansas

Abstract

Generative artificial intelligence (AI) presents a number of challenges for engineering educators. It is particularly challenging for those teaching lower level programming courses where a number of generative AI tools are capable of creating functional code in several computer languages. To address this, assessment of homework in a first-year Mechanical Engineering course was changed from a focus on functioning code to the process of creating code. This switch in focus from outcome to process was accomplished using in person grading where students were expected to describe their code and any challenges they faced writing the code. Students were then asked questions to help them reflect on their code and on their understanding of new concepts. Examination of midterm exam grades found a slight improve in scores with the implementation of in person grading.

Keywords

Artificial Intelligence, Assessment, Programming

Introduction

Generative artificial intelligence (AI) has been the talk of the universities since the advent of ChatGPT in November 2022. The development of generative adversarial networks, transformers, and large language models in the last decade has allowed the creation of generative AI capable of writing fluent text, developing workable programming code, and creating artistic works [1]–[7].

When considering the impact of generative AI on a course, it is important to examine generative AI in the context of how learning is assessed within the classroom. The easiest option for this is to see what a few of the main AI tools (such as ChatGPT, Bard, Bing, and Claude) do with assignment prompts. An initial assessment by the author in January and June of 2023 found that ChatGPT did quite well at generating homework answers for a coding course. It could also generate reasonable text for writing samples but tended to have hallucinations around researched materials. For example, ChatGPT could generate a reference list of papers, but the papers referenced were entirely fictious. For problem solving in upper-level engineering courses, ChatGPT produced less useful results. Claude was found to produce answers that had the right mathematical form but were mathematical incorrect from one line to the next. More recently, ChatGPTplus (the paid version of ChatGPT) has introduced Code Interpreter and a plugin for Wolfram Alpha that may change this. For some courses, generative AI may not have an impact on courses yet. The technology is rapidly evolving, however, and one should continue to be vigilant in considering the impacts of AI.

Once one has determined the impact of AI on a course, there are several courses of action. These include banning generative AI from the classroom, allowing limited use of generative AI, incorporating generative AI into the classroom and engaging with it, structuring classroom work such that generative AI impacts are minimized, or changing assessment such that the desired process of learning is rewarded.

1. Ban all generative AI.

   Some faculty may consider banning the use of generative AI in the classroom or for high stakes activities such as exams. Nominally this might be a statement in the syllabus. It might also be banned for specific types of assignments within a class.

   As generative AI expands across platforms, such an approach can be difficult as generative AI appears in document editors and learning management systems. In some situations, such as in class exams and activities that are done on paper within the classroom, it may be possible to prohibit technology (such as cell phones and computers) that can access generative AI.

   When technology is not prohibited, a ban on generative AI may rely on students' academic integrity. This can be problematic. AI detection software is not foolproof. It is known to generate both false positives and false negatives [8], [9]. AI detectors can be fooled, sometimes with simple text changes, and don't work for all types of output (such as computer code). There are equity concerns as false positives have been found to occur in particular populations including non-native English writers [10]. If access to electronic devices is not limited and detection is difficult, academic misconduct may not be easy to prevent. Justice issues may arise if students who chose to violate AI prohibitions are not caught and appear to have success in a course.

2. Allow limited use of generative AI.

   Another option is to allow limited use of generative AI. In these cases, a course might require an original first draft of a writing assignment but then allow use of generative AI in the editing process. Conversely, a course might allow AI generated first drafts and focus on editing the output for correctness. Like outright banning of AI, these options require reliance on students' academic integrity. One might, however, integrated this approach with the other following approaches.

3. Build generative AI into coursework.

   Generative AI is quickly moving into all aspects of society. As such, many would argue that we need to think about educating the next generation to utilize generative AI productively in the same way that faculty in the 1980s and 1990s encouraged learning computer technologies such as spell checkers and word processors. It is still early days in imagining what this future world will look like. For now, a good question to ask may be what are one's learning outcomes for a course and how could they change in the future?

Can one's learning outcomes still be achieved with free use of generative AI? How might one teach about the current use of generative AI tools in parallel with a course's learning outcomes?

4. Structure work such that generative AI is not as useful.

   Rather than require one outcome, such as a paper, one can structure assignments such that generative AI is not as useful to a course. For example, in a literature review paper one might require an iterative process of:

   a. generating an outline or a flowchart,
   b. then generating a supporting reference list,
   c. then developing summaries of supporting literature,
   d. then generating text.

   For each step, one might have peer review and feedback. In such an exercise, generative AI could be used for some elements, but the bulk of the process would not be supported by AI resources.

5. Assess the process rather than the outcome.

   Nancy Gleason [11] has suggested "assessing the process rather than the outcome" as a response the generative AI. In this scenario, points are assigned not for the quality of a final deliverable but for steps used to get to a deliverable. As students are often focused on and driven by points, one can use the points to award desired process steps and reflections on learning. For the structured example above, points could be assigned for each step of the process that outweigh the score of the final paper. Points can also be awarded for the process of reflection on learning.

Implementing a New Approach in a Computer Programming Course

ME 208 is a 3-credit hour programming course for first-year students in Mechanical Engineering. In Fall 2022 the course had 83 students and in Spring 2023 the course had 77 students. The course has a 50-minute lecture, a two-hour laboratory, and a 50-minute discussion period. During a typical week, students complete an active learning exercise in the lecture period, an individual laboratory assignment during the laboratory period, and an individual homework assignment after the laboratory. During the discussion period, the students worked in groups of 5-7 students on three group projects. In the course, the first half of the semester is focused on programming Arduino microcontrollers in C++ and the second half of the semester is focused on programming in the Matlab programming platform.

For the weekly homework assignments, generative AI was identified as a threat to student learning. Using the homework prompts in January 2023, it was found that ChatGPT could generate near perfect code for many of the homework assignments in both C++ and Matlab. While the solutions created using generative AI were functional code, ChatGPT did not recognize the context of the class and often used methods that had not yet been covered in the course.

In prior semester and Fall 2022, students were required to submit their code and a video in which they described their code and demonstrated its functionality. For Arduino C++ code, students were also asked to submit a diagram of their wiring and to demonstrate the physical system on the video, For the Matlab code, students were asked to submit any figures generated. For later assignments (after coverage of loops and conditional statements), students were also asked to submit a flowchart for their code. Both the videos and flowcharts were used to encourage students to submit original work rather than code from outside sources. Students were also discouraged from using code from outside the course within the syllabus and during class lectures.

While video and flowcharts submissions did provide some structure to discourage using code from outside sources, the advent of ChatGPT and similar generative AI resources that can produce computer code required a change. In Spring 2023, while keeping the assignments similar, grading was changed to reward the process rather than the outcome. To accomplish this, grading was moved from being performed by teaching assistants outside of class to in person grading performed during laboratory hours and office hours. The points awarded for functionality of the code submitted were reduced from 10 points to 3 points (out of a total of 20 points, Figure 1). Instead of awarding 5 points for a submitted video, 10 points were assigned to the in person grading. For in person grading, 5 points were assigned just for attending a grading session. Successful explanation of submitted code (even if that code was not functional) was awarded 2 points. Finally, students were asked questions for 3 points. The questions were developed over the semester. The recommended questions that were developed included: 1. What did you not understand about your code or this assignment?, 2. What did you struggle with on this assignment?, and 3. Questions about their code, new concepts in that assignment, and how lines within their code functioned. The goal of these questions was for students to reflect on their work, demonstrate their understanding of the material, and have a chance to receive feedback on issues they may have had. This type of questioning can be intimidating so efforts wear made to take a friendly, coaching approach. Prompts and hints were given to help students do their best. The goal was to recognize effort rather than necessarily having all the answers and working code. In person grading also had the effect of creating personal accountability, as work was not graded by some anonymous teaching assistant but rather the person they were talking to.

Fall 2022 Grading Rubric:

- Code Functionality (10pts)
- Commenting (2pts)
- Diagrams/Figures/Flowchart (3pts)
- Video (5pts)

Spring 2023 Grading Rubric:

- Code Functionality (3pts)
- Commenting (2 pts)
- Diagrams/Figures/Flowchart (5 pts)
- In Person Grading (10 points)
  - Attended In Person Grading (5 pts)
  - Explain Code (2 pts)
  - Answered Questions (3 pts)

Figure 1 Grading rubric transition from Fall 2022 to Spring 2023 reflected a chance in emphasis from outcome to process.

Examining the grades on the first seven homework assignments in the course, it was found that average homework grades decreased with the new grading system by 1 point (out of 20 points). This difference was not found to be statistically significant. Conversely, the grades on the midterm exam increased 3.3 points (out of 100 points). This difference was also not statistically significant but demonstrated that the change in assessment had a neutral to positive effect on overall classroom performance. For this semester, students were not forbidden from using generative AI but were told they should cite any outside coding sources (including generative AI) in the comments of the code. The instructor and teaching assistants did not observe significant unauthorized use of generative AI within the course (as might be evident from the use of coding methods not yet covered in the class).

Conclusion

In a beginning programming course, generative AI is a significant threat to student learning as it can generate functioning code for entry level assignments across a range of programming languages. Focusing assessment on process rather than outcome is a means to encourage students to work on problems themselves rather than seek out easy solutions from generative AI or other resources.

References

[1] I. Goodfellow *et al.*, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2014. Accessed: Jul. 31, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

[2] M. Hawley, "The Complete Generative AI Timeline: History, Present and Future Outlook," *CMSWire.com*. https://www.cmswire.com/digital-experience/generative-ai-timeline-9-decades-of-notable-milestones/ (accessed Jul. 31, 2023).

[3] A. Vaswani *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Jul. 31, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[4] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving Language Understanding by Generative Pre-Training,” *OpenAI*, 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed Jul. 31, 2023).
[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language Models are Unsupervised Multitask Learners,” *OpenAI*, 2018. https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf (accessed Jul. 31, 2023).
[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” May 24, 2019. http://arxiv.org/abs/1810.04805 (accessed Jul. 31, 2023).
[7] M. Casey, “Large language models: their history, capabilities and limitations,” *Snorkel AI*, May 25, 2023. https://snorkel.ai/large-language-models-llms/ (accessed Jul. 31, 2023).
[8] C. Chaka, “Detecting AI content in responses generated by ChatGPT, YouChat, and Chatsonic: The case of five AI content detection tools,” *Journal of Applied Learning and Teaching*, vol. 6, no. 2, Art. no. 2, Jul. 2023, doi: 10.37074/jalt.2023.6.2.12.
[9] E. Ferrara, “Social bot detection in the age of ChatGPT: Challenges and opportunities,” *First Monday*, Jun. 2023, doi: 10.5210/fm.v28i6.13185.
[10] W. Liang, M. Yuksekgonul, Y. Mao, E. Wu, and J. Zou, “GPT detectors are biased against non-native English writers.” arXiv, Jul. 10, 2023. doi: 10.48550/arXiv.2304.02819.
[11] N. Gleason, “ChatGPT and the rise of AI writers: how should higher education respond?,” *THE Campus Learn, Share, Connect*, Dec. 09, 2022. https://www.timeshighereducation.com/campus/chatgpt-and-rise-ai-writers-how-should-higher-education-respond (accessed Jan. 04, 2023).

Sara Wilson

Sara Wilson is an associate professor of the mechanical engineering and a core faculty of the bioengineering department at the university of Kansas. Her research focus is on the neuromuscular control of human motion using engineering principles from control theory and dynamics. She is also active in teaching and development of educational tools in responsible conduct of research for graduate students in engineering.