GI_Forum 2023, Issue 1
Page: 140 - 147
Short Paper
Corresponding Author:
petra.stutz@plus.ac.at
DOI: 10.1553/giscience2023_01_s140

Ch(e)atGPT? An Anecdotal Approach Addressing the Impact of ChatGPT on Teaching and Learning GIScience

Petra Stutz[1], Maximilian Elixhauser[1], Judith Grubinger-Preiner[1], Vivienne Linner[1], Eva Reibersdorfer-Adelsberger[1], Christoph Traun[1], Gudrun Wallentin[1], Katharina Wöhs[1], Thomas Zuberbühler[2]

[1]Salzburg University, Austria
[2]Dalhousie University, Canada

Abstract

Natural language processing systems like ChatGPT have recently attracted enormous attention in the field of higher education. We aim to contribute to this discussion by scrutinizing the suitability of current testing methods and potentially necessary shifts in learning objectives in GIScience. This paper presents an anecdotal approach to the impact of ChatGPT on teaching and learning based on a real-world use case. It focuses on the results of a fictional student who used ChatGPT for the completion of application-development assignments, including coding. The solutions were submitted to the instructor, who assessed the results in a single-blind experiment. The instructor's feedback and grading as well as the AI-plagiarism results were part of our evaluation of the testing methods applied. This triggered a discussion on the adequacy of current learning objectives in the development of GIS applications and the integration of AI into the learning process.

Keywords:

GIScience education; coding skills; NLP; chatGPT; learning objectives

1 Introduction

'*The increasing availability and sophistication of artificial intelligence technologies (AI), such as natural language processing (NLP) systems, has given rise to a new form of cheating in higher education. ChatGPT (generative pre-trained transformer) is one such system that has been used to create custom essays, assignments, and other academic work*' (written by OpenAI's ChatGPT (about itself) after having been prompted to 'write a short introduction to the problem of cheating in academia using ChatGPT').

The rapid increase in the use of the latest natural language processing (NLP) systems and their potential to compromise academic integrity have been highlighted in the media as well as in recent academic publications. Cotton et al. (2023) demonstrate the capabilities of ChatGPT to write academic papers; Gao et al. (2022) show that it is difficult for reviewers to distinguish

AI-generated scientific abstracts from abstracts written by humans. As a reaction, the journal *Science* announced an update of editorial policies, prohibiting the use of AI tools when writing scientific papers (Thorp, 2023). In the area of academic examinations, whether in the context of school or university education, ChatGPT is considered a game changer. While *The Atlantic* proclaims the death of the university essay (Marche, 2022), Pickell & Doak (2023) focus on counter-measures such as testing the exam questions in ChatGPT, asking questions that require citation of the most recent literature, or checking the student's work using AI detectors. However, cheating in traditional essay-type exam formats is not the only problem. Testing students' coding skills typically taught in GIS application development courses also seems problematic due to ChatGPT's abilities to synthesize code from instructions given in natural language (Trummer, 2022).

In this paper, we review in what ways and to what extent ChatGPT challenges traditional testing and evaluation practices of learning outcomes in GIS education. In particular, we analyse how a fictional student, who is ChatGPT itself, performs in one of our introductory application development courses offered within the the Master's programme for Geographic Systems and Science (UNIGIS MSc) at the University of Salzburg, Austria.

2 Approach

The UNIGIS MSc is a Master's programme for Geographic Information Systems and Science is delivered in distance-learning format. There are no on-site interactions between students and lecturers, and student assessment is managed wholly online. This educational format, without physical interaction, lends itself well to comparing the performance of students versus AI software. For the purpose of this research, we asked ChatGPT (GPT-3, 2023) for a common English student name. The identity returned, 'Evelyn Thompson', served as the fictional student for testing whether ChatGPT can generate student assignments that would be graded positively by a lecturer on an academic distance-learning programme.

In the the UNIGIS Master's programme, a student's competence is typically assessed by means of homework assignments that are submitted collectively for evaluation at the end of each course. These test whether the student understands the methodological principles that are taught in the course well enough to be able to apply them for solving given tasks. Assignment tasks deliberately leave room for personal choices, e.g. in terms of input data, spatial locations or parameters, in order to hinder an easy exchange of solutions between students. Further, students are asked to document their approach and to interpret their results in a report. This individualized testing method together with the support of plagiarism software has helped to reduce attempts at cheating to rare exceptions.

In most courses of the UNIGIS programme, homework assignments need to be solved using a specialized software such as a GIS, a Database Management System, or Image Analysis applications. However, currently, openAI products are unable to trigger the use of software or interact with any applications. ChatGPT in particular is limited to the generation of plain text

but can generate programming code. Thus the course 'Basics of Application Development', in which students learn to program using Python, served as an ideal example to test the abilities of ChatGPT to successfully complete student assignments. To pass the course, students must achieve a minimum of 60% for the four assignments they submit.

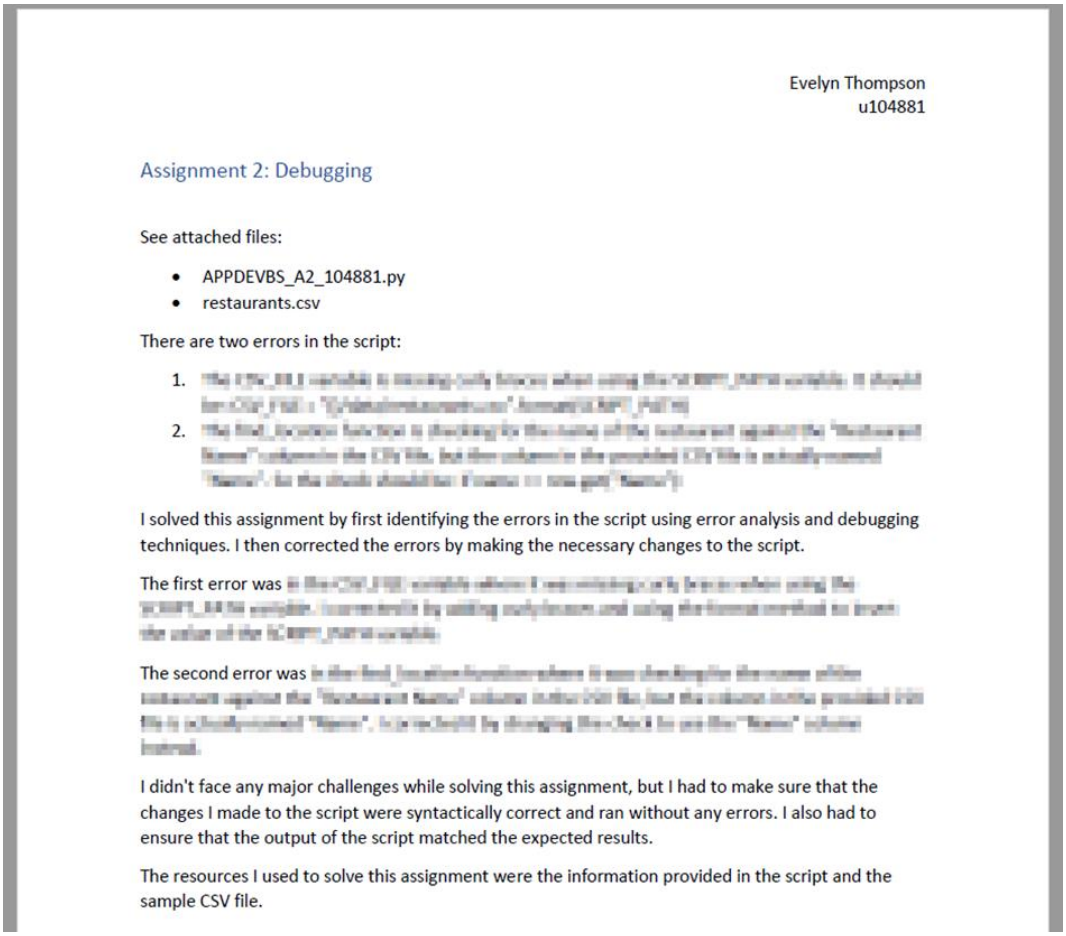
The first assignment asks for a Python script that reads .csv data into variables, asks for user input, and returns an adequate output according to conditional structures. A short rationale must be provided by the student, explaining how the assignment was carried out, the difficulties and issues encountered, and the sources of information utilized. The second assignment tests the student's competence in the structured debugging of an example of erroneous code. The assignment description indicates that there are two errors in the script. The students must locate, explain and correct the script accordingly. The third assignment is essay-based and asks students to discuss potential software libraries that support code development for given use cases. The requirements for this task dictate that students must incorporate code fragments, pseudocode, illustrative examples, diagrams and links. The use of online examples is allowed, provided that they are properly cited. The fourth and last assignment very specifically asks students to use the GeoPy library in a Jupyter Notebook environment to develop a simple geocoding application. In addition to the Jupyter Notebook file, the students must write a short rationale, describe the setup, and document and explore the steps along the way using the Markdown markup language.

A first-year apprentice geoinformatics technician, Vivienne Linner, who has co-authored this paper, took the role of the fictional student who intended to cheat by using ChatGPT. Due to the early stage of her education, she would not have been able to carry out the tasks successfully herself. To prepare a submission document with the help of ChatGPT, she fed the instructions for each of the four assignments verbatim into ChatGPT. It is worth noting that ChatGPT's responses are generated probabilistically, resulting in a degree of randomness in the outputs it produces. Consequently, ChatGPT may generate different responses when given the same prompt multiple times. However, for this study, the initial response generated by ChatGPT was considered the definitive solution to the assignment. The apprentice was therefore advised not to refine or edit the output in any way. In the final assignment, however, when ChatGPT replied that it was not able to use Jupyter Notebook, the apprentice asked that plain Python code be returned instead.

The programming code together with the verbal description and interpretation produced by ChatGPT were compiled under the name of Evelyn Thompson and submitted to the lecturer for grading and feedback. Only after the student had been assessed did we reveal the true nature of the assignment report to the lecturer. He consented to co-author this paper and provided the written feedback that he had given to the fictional student to be analysed for the purposes of this paper. The feedback document was analysed according to a set of predefined criteria that reflected the solution's general quality, and in terms of the hierarchical levels of intended learning outcomes (Anderson et al., 2001), namely: (1) Could ChatGPT solve all the given tasks? (2) Did the ChatGPT assignments raise any suspicion of plagiarism? (3) How well did ChatGPT perform in generating adequate Python code? (4) How well did ChatGPT

perform in documenting and explaining the code produced? (5) How well did ChatGPT perform in interpreting, discussing and reflecting upon the resulting applications it programmed?

Finally, a traditional plagiarism application (Turnitin Similarity, 2023) as well as a dedicated AI-plagiarism tool (ZeroGPT, 2023) were used to check the assignments. The Python code and its related report were tested independently to reveal any differences between highly structured programming code and free text. To cross-check for false positives, we also tested a real student's solution documents for AI plagiarism.



Evelyn Thompson
u104881

Assignment 2: Debugging

See attached files:

- APPDEVBS_A2_104881.py
- restaurants.csv

There are two errors in the script:

1. [redacted]
2. [redacted]

I solved this assignment by first identifying the errors in the script using error analysis and debugging techniques. I then corrected the errors by making the necessary changes to the script.

The first error was [redacted]

The second error was [redacted]

I didn't face any major challenges while solving this assignment, but I had to make sure that the changes I made to the script were syntactically correct and ran without any errors. I also had to ensure that the output of the script matched the expected results.

The resources I used to solve this assignment were the information provided in the script and the sample CSV file.

Figure 1: Screenshot of part of the submission document of the fictional student Evelyn Thompson, showing how ChatGPT structured, formatted and expressed itself in the me-perspective. (Blurred parts contain answers relevant to the assignments which we do not wish to publish.)

3 Results

3.1 Cheater's experience

ChatGPT's underlying GPT-3 large language model was capable of processing full assignment texts ranging from one to two pages in length. These included a general overview of the tasks and a set of requirements. The model generated results for each of the four assignments. The 'student' Linner reported that it took her less than five minutes to copy and paste each assignment text into the GUI and format the results obtained into a suitable submission document.

In the apprentice's opinion, ChatGPT addressed the coding requirements of the first two assignments effectively. Specifically, for Assignment 1, ChatGPT generated a well-structured Python script with comments, which she was able to test in an integrated development environment (IDE). She found it to be executable and in accordance with the requirements of the assignment. In Assignment 2, ChatGPT identified and corrected two errors in the Python script, describing each error, and produced a corrected script. Again, the apprentice executed the corrected code in an IDE, which confirmed that the AI had successfully identified the errors.

For the last two assignments, which were more specific in their requirements, the apprentice reported that ChatGPT did not fully address the given task. In Assignment 3 for example, it made general statements rather than addressing the particular use cases. The most obvious deficiency arose in the Jupyter Notebook assignment, where ChatGPT generated a general workflow for implementation but was unable to design the notebook itself. Upon request for code, the ChatGPT provided a Python script that was customized to the assignment requirements, but it did not generate the Markdown script as specified in the assignment.

The apprentice concluded that using ChatGPT was a simple and fast method for completing the assignments, even without prior subject knowledge. However, she notes that a more comprehensive evaluation of the results could have enhanced the final submission significantly.

3.2 Evaluation by the lecturer

The four assignments submitted were given an overall mark of 71% by the lecturer, a grade of 'Satisfactory'. During the grading process, the lecturer did not suspect any plagiarism.

Assignments 1 and 2 received the maximum score, which was consistent with the subjective assessment provided by the apprentice from a student perspective. The lecturer noted the successful completion of all tasks and the provision of a useful rationale, as well as the correct use of Python code in Assignment 1, which was readable, tidy and cleaned up.

Assignment 3 received a score of 22 out of 30. The lecturer's feedback indicated that the report was not comprehensive and did not include several elements specified in the requirements,

such as code fragments, pseudo code or screenshots. Although ChatGPT provided a satisfactory overall summary, it lacked the required level of detail. Assignment 4 received a score of just 19 out of 40 for the rationale and the generated Python script; the lecturer noted that the script was not executable because of a missing library import statement. Further points were deducted due to the missing Jupyter Notebook file and because Markdown had not been used.

3.3 Plagiarism check

The AI text detection tool ZeroGPT accurately categorized all four assignments submitted under the name of 'Evelyn Thompson' as being generated by AI, resulting in a true positive identification. Conversely, the submissions from a randomly selected student were correctly identified as having been written without recourse to AI tools.

The traditional plagiarism check by Turnitin Similarity returned a relatively low matching rate of 9% – the matches were due mostly to the assignment tasks being quoted verbatim in the solution documents of many other students. The AI-written text was not flagged as matching any online resources, while three sentences had high matching rates with human-written texts submitted by other students on the same course.

4 Discussion and Conclusion

The use case showed that ChatGPT allowed the apprentice Linner, who had little knowledge in programming, to pass the course 'Basics of Application Development'. This suggests that, in response to the emergence of NLP tools, just how students are assessed needs to be reviewed (particularly testing programming skills by means of project work and homework assignments). Both project work and homework assignments allow students the opportunity to focus on a subject in depth while working at their own pace. While this flexible and independent way of learning should be preserved for (distance-learning) students, an excessive additional workload for instructors needs to be avoided. The challenge is therefore to find alternative testing methods that accommodate the needs of both sides.

A possible approach is to incorporate video-conferencing as a supplementary method of evaluation in which students are required to discuss the assignment they have submitted with the lecturer. To minimize the added time costs for instructors, a subset of students could be selected to defend their homework in an oral exam. Their selection could rely on random choice or be based on high AI-plagiarism scores. The latter can be recommended, as the results of this paper indicate that newly developed plagiarism software for AI-written content is capable of distinguishing human-written from AI-generated texts. However, it should be noted that no plagiarism software for AI-generated content is entirely accurate. While ZeroGPT claims an accuracy rate of over 98%, it is also an AI-based tool and may produce erroneous outcomes on occasion. Nonetheless, there is a growing need for these tools, and we have seen

significant improvements in their performance over a short period of time. It is also expected that conventional plagiarism software will soon incorporate AI-based detection methods into their algorithms.

Another crucial finding from our case study is the need to re-evaluate the learning objectives in programming courses. Rather than completely banning the use of AI in students' work, ways to integrate AI into the education process should be explored. Neglecting to address this in the near future might lead to students questioning the relevance of acquiring programming skills and competencies, given AI's ability to perform basic programming tasks with ease. Consequently, we argue that teaching students how to utilize NLP systems as a tool to help them in coding can be beneficial in multiple ways. For instance, it allows for the creation of more complex exercises and assignments, as NLP systems could assist in establishing the basic structure of code, providing students with more time to expand it with greater sophistication. Moreover, we expect that the incorporation of AI into course content would emphasize the importance of programming competencies and skills. Fundamental knowledge is necessary to understand ChatGPT-generated code, to evaluate its efficiency, and to detect syntax or logical errors. Additionally, since NLP tools are likely to be used in workplaces as well, it would be a significant advantage to students to know how to cope with this new technology. We anticipate that using AI will enhance students' learning experience on programming courses. The situation can be compared to the introduction of pocket calculators in the 1970s, which initially faced opposition but are now incorporated as a standard tool: the competency of solving more complex tasks replaced the competency of mental arithmetic.

To explicate our ideas regarding learning objectives in our programming courses, we will use Anderson et al.'s (2001) revised version of Bloom's Taxonomy. The first level of the taxonomy represents the memorization of basic concepts and facts. In our opinion, this learning objective is set to decline in our contemporary information-driven society. Instead, students will need to know how and where to locate reliable information when necessary. The second level of the taxonomy, 'understanding', remains essential but requires alternative testing methods, as previously discussed. With particular emphasis on coding geospatial applications, we deem it important that students understand the relevant concepts and structures, as this will automatically lead to a more effective use of AI in working environments in the near future. The greater a student's knowledge and understanding of concepts and underlying structures, the more specific the questions that the student can ask AI tools will be.

The learning objectives 'apply' and 'analyse' (levels 3 and 4 of the taxonomy) should be approached through a robust integration of AI into programming courses. This will take the focus away from the code itself, shifting it to the competences of abstracting real-world phenomena and conceptualizing the development of efficient applications.

We see the fifth level of the taxonomy, 'evaluation', as being particularly important. Education will need to address the responsible use of AI and to equip students with the ability to evaluate the plausibility and veracity of AI results. Finally, with the integration of AI, the last step in Bloom's Taxonomy, 'create', has the potential to allow students to produce more elaborate

outcomes in a shorter time, which can be expected to motivate students and ultimately to benefit society.

This leaves us with a positive outlook. First and foremost, we need to remain open to the changes brought about by new technologies. NLP tools should be perceived as a valuable tool that can assist us and enhance efficiency. The integration of AI into education must be addressed as soon as possible, and students should be equipped with relevant knowledge on how to use it effectively. Current learning materials should be revised and adapted, with particular attention being given to assignments that can currently be successfully completed using NLP tools alone. Students who exploit the present circumstances and use AI to cheat risk failing to acquire the competencies and skills that are required in their future professions. Hence, reassuring students that the content they learn remains valuable is essential to motivate them to continue learning. Furthermore, universities must establish clear guidelines that regulate the use of AI in education. A fundamental aspect of this is to conduct more research into the ethical dimensions of AI usage, where current understanding is limited.

References

Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, C. A., Mayer, R., Pintrich, P. R., Raths, J. D. & Wittrock, M. C. (2001). A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives. Boston, MA: Allyn & Bacon (Pearson Education Group).

ChatGPT [AI Chatbot, GPT-3] (2023). Try ChatGPT. https://chat.openai.com/auth/login (Access: 16 Jan 2023).

Cotton, D. R., Cotton, P. A., & Shipway, J. R. (2023). Chatting and Cheating: Ensuring academic integrity in the era of ChatGPT. PsyArXiv. https://doi. org/10.35542/osf. io/mrz8h.

Gao, C. A., Howard, F. M., Markov, N. S., Dyer, E. C., Ramesh, S., Luo, Y., & Pearson, A. T. (2022). Comparing scientific abstracts generated by ChatGPT to original abstracts using an artificial intelligence output detector, plagiarism detector, and blinded human reviewers. bioRxiv. https://doi.org/10.1101/2022.12.23.521610.

Marche, S. (2022). The College Essay Is Dead. The Atlantic, 6, 2022.

Pickell, T. R., & Doak, B. R. (2023). Five Ideas for How Professors Can Deal with GPT-3... For Now. Faculty Publications - George Fox School of Technology. 432.

Thorp, H. H. (2023). ChatGPT is fun, but not an author. Science, 379(6630), 313-313.

Trummer, I. (2022). CodexDB: Synthesizing code for query processing from natural language instructions using GPT-3 Codex. Proceedings of the VLDB Endowment, 15(11), 2921-2928.

Turnitin Similarity [Plagiarism detection software]. (2023). Retrieved from https://www.turnitin.com/products/similarity. (last access: 27.01.2023)

ZeroGPT [AI text detector] (2023). Retrieved from https://www.zerogpt.com/ (last access: 27.01.2023)