

USING CHATGPT TO GENERATE COMPUTER SCIENCE PROBLEM SETS

Tyler Flaagan

The Beacom College of Computer and Cyber Sciences, Dakota State University,
Madison, SD, USA

ABSTRACT

The topic of this paper is the exploration of using ChatGPT as a tool to assist educators in creating innovative computer science problem sets for programming classes. The motivation for this research stems from the challenges educators face in continually developing engaging and relevant problem sets that stimulate students' critical thinking and align with the evolving landscape of the computer science field. By harnessing the capabilities of ChatGPT, we aim to provide a solution that can alleviate the burden on educators and enable them to create dynamic, diverse, and effective learning experiences for their students.

KEYWORDS

ChatGPT, machine learning, computer science, programming languages, problem sets.

1. INTRODUCTION

ChatGPT, an advanced language model developed by OpenAI, has gained significant attention in recent months as a powerful tool for a variety of applications. It is based on the GPT-4 architecture and has demonstrated remarkable capabilities in generating human-like text [1]. In this paper, we explore the potential of ChatGPT to assist computer science instructors in creating innovative problem sets. ChatGPT very little in the way of information published due to its novelty [2].

The motivation for this research lies in addressing the pressing challenges educators face in their quest to develop engaging and relevant problem sets that stimulate students' critical thinking and align with the rapidly evolving landscape of the computer science field. As technology advances at an unprecedented pace, educators often need help to keep up with new trends and industry developments while ensuring that their teaching methods and materials remain compelling and captivating for students. Furthermore, the need to promote critical thinking and problem-solving skills in the context of real-world applications adds a layer of complexity to the task. By exploring innovative solutions such as ChatGPT, this research aims to empower educators with the tools and resources necessary to overcome these challenges, ultimately fostering a more enriching and dynamic learning environment for students in computer science courses.

ChatGPT has already begun to impact academia One such perspective discusses the potential implications of publishing scholarly content using artificial intelligence that delve into copyright, attribution, plagiarism, and authorship [2]. ChatGPT has shown an incredible ability to pass traditional higher-education assessments, causing some to call for bans while taking tests [3]. Attempting to mitigate the negative uses of advanced language model tools will be difficult, if not impossible, since similar products becoming available is inevitable. Other problems include a

lack of human interaction, limited understanding, bias in training data, lack of creativity, dependency on data, lack of contextual understanding, limited ability to personalize instruction, and privacy [4].

Advanced tools can positively impact students as well. One such example is the use of AI for intelligent student tutoring [5]. Outside of tutoring, others have also discussed its use for solving programming bugs to aid in software development [2]. The way in which it is used in an educational environment will decide whether its impact is positive or negative [6].

2. BACKGROUND

The rapid advancements in computer science have made it increasingly more work for educators to keep up with the pace and create engaging, relevant problem sets for their students. This challenge often leads to a heavy workload and a need for more diverse educational resources that cater to the ever-changing needs of programming classes. With the advent of advanced language models like ChatGPT, there lies an opportunity to revolutionize the way educators develop computer science problem sets, enabling them to adapt to the evolving landscape of the field more efficiently.

ChatGPT, a powerful language model based on the GPT-4 architecture, has demonstrated its capabilities in generating human-like text, understanding context, and providing creative solutions across various domains. By harnessing these capabilities, educators can create a wide array of problem sets that are tailored to their student's needs and consider current industry trends and real-world scenarios. This allows for a more dynamic and diverse learning experience, fostering critical thinking and problem-solving skills among students.

Integrating ChatGPT into creating computer science problem sets offers several advantages:

1. It can substantially reduce the time and effort educators spend developing new problem sets, allowing them to focus on other essential aspects of teaching, such as providing personalized feedback and facilitating student discussions.
2. ChatGPT can be fine-tuned to generate problems based on specific topics, difficulty levels, or learning objectives, ensuring that the generated problems align with the curriculum and educational goals.
3. ChatGPT can help educators consistently introduce novel and engaging problems, maintaining students' interest and motivation throughout the course.

In an ideal scenario, an instructor could reuse the same examples semester after semester as fundamental computer science knowledge units rarely change. In practice, some issues make the reuse of problems less than ideal. One such issue is if a student needs to retake the same course. The cause for retaking a course comes from various issues that do not need to be explored here, but a student retaking the same course may already have work completed, and if they are retaking, they may not have a strong understanding of the content. Another is that students may post solutions online. There are two common reasons why a student may post online. First, they are posting to homework-helping sites and forums such as homeworkhelper.com and chegg.com, thus providing easy access to information that would fall under academic dishonesty. There are positive reasons that a student may post online as well. An example is a student posting code to code repository sites like Github.com. Posting to publicly available code repository sites allows students to create a code portfolio to demonstrate proficiency to potential employers.

In conclusion, using ChatGPT to generate computer science problem sets holds immense potential in transforming programming education. By addressing the challenges educators face in

developing relevant and engaging problems, ChatGPT can help bridge the gap between the dynamic nature of the computer science field and students' learning experiences. Future research needs to explore the practical implications of ChatGPT-based problem generation, assess its effectiveness, and develop best practices that educators can adopt globally.

2.1. ChatGPT

ChatGPT, a cutting-edge language model, has recently gained significant attention as a powerful tool for various applications across various fields. To effectively utilize ChatGPT, users can start by posing a vague question and provide additional context in a subsequent prompt after receiving the initial response. This allows the AI to recall and build upon previous conversations with the user, ensuring a more coherent and meaningful interaction.

Users can refine the discussion during the conversation by providing clarifying comments or feedback. This ensures that the AI understands the user's intent better, enabling more accurate and informative responses. Additionally, ChatGPT can reject inappropriate requests, although this feature does not impact the subject matter of this paper.

However, it is essential to recognize the limitations of ChatGPT. For instance, when generating problem sets, the AI may occasionally produce incorrect information. Consequently, users should verify the problem sets for accuracy and identify potential edge cases that need addressing. Furthermore, ChatGPT may generate harmful instructions or biased information, a prevalent concern in artificial intelligence and machine learning.

Another limitation was ChatGPT's knowledge cut off in September 2021. As a result, AI may need more awareness of events, advancements, or innovations that have taken place after this date. Despite this constraint, most fundamental computer science concepts remain unchanged, allowing ChatGPT to continue serving as a valuable resource for generating problem sets. For example, teaching an introductory concept like the variable declaration in the C programming language has stayed the same since the end of ChatGPT's training period in 2021. Even with the limitations mentioned earlier, ChatGPT remains a valuable tool for educational purposes and various other applications, contributing significantly to artificial intelligence and machine learning. ChatGPT is a sophisticated language model that has attracted considerable attention lately as an influential instrument for numerous purposes across diverse fields.

3. RELATED WORKS

A myriad of sources exists for computer science problem sets, catering to the needs of both students and instructors. Project Euler, one such resource, offers a collection of challenging mathematical and computational problems [7]. These problems encourage creative problem-solving and programming skills, allowing students to test their understanding and grow their abilities. With many problems available, Project Euler is an excellent resource for educators to diversify their curriculum and challenge their students. LeetCode is another valuable source for coding challenges and problems. This platform helps students refine their programming skills and prepares them for technical interviews. Educators can use LeetCode to create problem sets that simulate real-world challenges and foster practical skills that students can carry into their professional lives [8]. HackerRank provides coding challenges, tutorials, and competitions across various domains, including algorithms, data structures, and artificial intelligence. The platform's versatility and range of topics make it a valuable resource for educators looking to create well-rounded problem sets. By incorporating HackerRank's coding challenges into their curriculum,

instructors can expose students to diverse concepts and assess their understanding in a competitive and engaging environment [9].

For those seeking a more structured learning experience, platforms like Coursera offer online courses covering various computer science topics. These courses often include problem sets, assignments, and quizzes as part of the curriculum. Educators can use these resources to supplement their classes, ensuring students receive a comprehensive understanding of the subject matter. The high-quality materials provided by Coursera can serve as a solid foundation for educators to build upon and adapt to their specific teaching needs [10].

Lastly, university websites often publish computer science course materials, including problem sets, online for public access. Renowned institutions like MIT and Stanford provide a wealth of resources for educators to create engaging and challenging problem sets. By leveraging these materials, instructors can benefit from the expertise of leading computer science departments and incorporate cutting-edge concepts into their classrooms. The wide array of sources available ensures that educators can continually update and enrich their curriculum, keeping students motivated and fostering a deep understanding of computer science principles.

4. CREATING COMPUTER SCIENCE PROBLEM SETS

The following sections will show results from ChatGPT when asking for specific learning objectives. A specific language was not used when prompting ChatGPT to show generic problems that can be adapted to whichever language is used in a particular class. Later, specifying languages will be discussing in improving results. While creating examples for demonstration, ChatGPT jailbreaks techniques such as the DAN jailbreak were not used to modify results.

The examples demonstrated in the following sections are based on the following topics that could be considered foundational concepts for developing applications in many programming and scripting languages. The list is not meant to be all possible foundational concepts but instead is a representative sample. The topics examined, in no particular order, are: if statements, loops, arrays, functions, and pointers.

Each topic supplied is prefaced with the phrase: " Write a challenging problem for a computer science course focused on learning..." followed by each specific topic from the list above.

4.1. If Statements

The following is the result of if statements without any modification by the author:

Title: "Temperature Alert System"
Problem Description:

You are tasked with developing a temperature alert system for an industrial facility. The system will read temperature values in Celsius from various sensors and categorize them into different alert levels based on the following conditions:

1. Low (L) - temperature less than or equal to 0°C
2. Normal (N) - temperature greater than 0°C and less than or equal to 80°C

3. High (H) - temperature greater than 80°C and less than or equal to 100°C
4. Critical (C) - temperature greater than 100°C

Your program should take an integer n (number of temperature readings) as input, followed by n temperature readings. For each temperature reading, the program should print the corresponding alert level.

Instructions:

1. Write a function called **alert_level** that takes a float (temperature) as input and returns a string representing the alert level.
2. Write the main function to read the temperature readings and call the **alert_level** function for each reading, then print the alert level for each temperature reading.

Constraints:

- The input integer n for the number of temperature readings will be in the range of $1 \leq n \leq 100$.
- The input float temperature readings will be in the range of $-100.0 \leq \text{temperature} \leq 200.0$.
- Use if statements to categorize the temperature readings; avoid using any loops in the **alert_level** function.
- The temperature values should be rounded to one decimal place when displayed with their alert level.

Example:

Input: 3 -5 25 105
Output: L -5.0 N 25.0 C 105.0

4.2. Loops

The following is the result of loops without any modification by the author:

Title: "Fibonacci's Nested Loop Adventure"

Problem Description:

In this challenge, you will write a program that generates a nested Fibonacci sequence matrix using loops. The nested Fibonacci sequence matrix is an $n \times n$ matrix, where each element is a Fibonacci number, and each row starts with the Fibonacci number of the index of the row. The index starts from 0.

For example, given $n = 5$, the output should be:

0 1 1 2 3 1 1 2 3 5 1 2 3 5 8 2 3 5 8 13 3 5 8 13 21

Instructions:

1. Write a function called **fibonacci** that takes an integer as an input and returns the corresponding Fibonacci number.
2. Write a function called **nested_fibonacci_matrix** that takes an integer n as input and returns a nested list representing the $n \times n$ Fibonacci matrix.
3. Print the generated matrix in a well-formatted manner, similar to the example provided.

Constraints:

- The input integer n for the `nested_fibonacci_matrix` function will be in the range of $1 \leq n \leq 10$.
- You must use loops to generate the matrix; recursion is not allowed for this problem.

Note: The Fibonacci sequence is a series of numbers in which each number is the sum of the two preceding ones, usually starting with 0 and 1.

4.3. Arrays

The following is the result of arrays without any modification by the author:

Title: "Array Transformation: Multiplying Elements by their Indices"

Problem Description:

In this challenge, you will write a program that reads an array of integers and performs a transformation on the array. The transformation multiplies each element in the array by its index, and the result is a new array. The program should then output the transformed array.

Instructions:

1. Write a function called **transform_array** that takes an array of integers, its length (int), and an empty array of the same length as input. The function should perform the transformation, filling the empty array with the transformed values.
2. Write the main function to read the input array, call the **transform_array** function, and print the transformed array.

Constraints:

- The number of input integers, n , will be in the range of $1 \leq n \leq 100$.
- The input integers will be in the range of $-1000 \leq \text{integer} \leq 1000$.
- The transformed array should have the same length as the input array.

Example:

Input: 5 2 5 8 3 6
Output: 0 5 16 9 24

4.4. Functions

The following is the result of functions without any modification by the author:

Title: "GPA Calculator for Different Grading Systems"
Problem Description:

In this challenge, you will write a program that calculates the Grade Point Average (GPA) of a student for two different grading systems. The program will take a series of letter grades as input and convert each grade into a numeric value based on the given grading system. It will then calculate the GPA using the converted numeric values.

The two grading systems are as follows:

1. Standard Grading System (SGS):

- A: 4.0
- B: 3.0
- C: 2.0
- D: 1.0
- F: 0.0

2. Modified Grading System (MGS):

- A+: 4.3
- A: 4.0
- A-: 3.7
- B+: 3.3
- B: 3.0
- B-: 2.7
- C+: 2.3
- C: 2.0
- C-: 1.7
- D: 1.0
- F: 0.0

Instructions:

1. Write a function called **convert_grade** that takes a string (letter grade) and a string (grading system) as input and returns a float representing the numeric value of the grade based on the given grading system.
2. Write a function called **calculate_gpa** that takes a list of letter grades, a string (grading system), and the number of courses (int) as input. The function should call the

convert_grade function for each letter grade and return the calculated GPA.

3. Write the main function to read the input letter grades and call the **calculate_gpa** function for both grading systems, then print the GPA for each system.

Constraints:

- The number of input letter grades, n , will be in the range of $1 \leq n \leq 20$.
- The input letter grades will be valid according to the grading systems provided (e.g., no invalid letter combinations).
- The input letter grades will be in uppercase.
- Round the calculated GPA to two decimal places.

Example:

Input: 5 A B A- C+ B-

Output: SGS GPA: 2.80 MGS GPA: 2.94

4.5. Pointers

The following is the result of pointers without any modification by the author:

Title: "Pointer Manipulation: Sorting an Array of Strings"

Problem Description:

In this challenge, you will write a C program that takes an array of strings as input and sorts the strings alphabetically using pointers. Your program should output the sorted array of strings.

Instructions:

1. Write a function called **compare_strings** that takes two pointers to strings (char^*) as input and returns an integer. The function should return a negative value if the first string is lexicographically less than the second string, a positive value if the first string is lexicographically greater than the second string, and 0 if both strings are equal.
2. Write a function called **swap_strings** that takes two pointers to strings (char^{**}) as input and swaps the string pointers.
3. Write a function called **sort_strings** that takes an array of pointers to strings (char^{**}) and its size (int) as input. The function should sort the strings alphabetically using the **compare_strings** and **swap_strings** functions.
4. Write the main function to read the input strings, call the **sort_strings** function, and print the sorted array of strings.

Constraints:

- The number of input strings, n , will be in the range of $1 \leq n \leq 50$.
- The length of each input string will be in the range of $1 \leq \text{length} \leq 100$.
- The input strings will only contain lowercase English letters.
- Use pointers and pointer manipulation to implement the sorting process. Avoid using built-in string manipulation functions.

Example:

Input: 5 apple orange banana kiwi grape

Output: apple banana grape kiwi orange

5. REFINEMENTS

ChatGPT's emergence onto the Internet is still recent at the time of writing, meaning its full potential may still need to be recognized. As technology advances, users and developers will continue discovering new capabilities and applications for this powerful tool. One of the exciting aspects of ChatGPT is its ability to adapt its responses based on additional information provided by the user.

For example, if a user asks ChatGPT to "answer the question like I am five years old," the AI model can tailor its response to the request, simplifying complex concepts and explanations. Additionally, users can request that the response be written in a specific style, adding another layer of customization to the generated content.

One of the critical capabilities of ChatGPT is its ability to reference previous parts of the conversation. This feature creates opportunities for refining the content while creating problem sets. ChatGPT can generate more relevant and accurate solutions by maintaining context and incorporating prior information. Using the loop example from the prior section, it is possible to refine or regenerate the response altogether. If a generated solution is too similar to another example or the educator sees issues or errors, it can easily be regenerated by re-entering the same question or clicking the regenerate button. This flexibility allows for continuous improvement of the problem sets. Furthermore, continuing to enter questions in the same chat session can impact the results. Since ChatGPT can reference previous prompts, it will adapt to the questions and improve the generated content over time. This iterative process enables users to achieve increasingly refined and targeted results. The difficulty is another aspect that can be modified. By simply asking ChatGPT to increase the difficulty, a more complex problem can be created for advanced computer science courses or for students who need to be challenged by the current content. This adaptability ensures that problem sets remain engaging and appropriate for students at various skill levels.

AI in higher education can produce positive results, but with the capabilities of advanced language models, there is also the potential for negative impacts. The power of Large Language Models (LLMs) is impressive in solving questions accurately when the user provides considerable detail. Given enough information, ChatGPT can generate code and solutions for various problems, highlighting its potential as a valuable tool in computer science education.

6. FUTURE WORKS

As we explore, further implementations of computer science problem sets, it is essential to recognize that the availability of such tools is still in its early stages. The rapid development and evolution of the field make it necessary to continually reevaluate various research areas, including creating problem sets using AI tools like ChatGPT. By staying up to date with advancements in the field, educators can ensure that they make the most of these emerging technologies.

With the maturation and increased adoption of AI-driven technologies, it becomes increasingly essential to revisit and reassess the implications and applications of these tools. Periodic assessments can help educators identify new possibilities and potential benefits that arise as the technology evolves. This ongoing evaluation process ensures that educators can make informed decisions on integrating AI tools into their curriculum best. Moreover, the responsible and effective use of innovative tools is crucial for harnessing their full potential in computer science education. As AI technologies become increasingly sophisticated, educators must balance their opportunities with the ethical considerations accompanying their use. By maintaining this balance, instructors can create enriching and engaging learning experiences while upholding the principles of responsible AI use.

7. RESULTS & CONCLUSION

This work shows that ChatGPT can generate focused problem sets for educators as needed. ChatGPT, an advanced language model based on the GPT-4 architecture, offers immense potential for revolutionizing computer science education. Its ability to generate human-like text, adapt to user input and reference previous conversations makes it a valuable tool for creating diverse and engaging problem sets for programming courses. By assisting educators in developing relevant and challenging problems that align with the dynamic landscape of computer science, ChatGPT contributes to fostering a more enriching and interactive learning environment for students.

However, it is essential to acknowledge the limitations and ethical considerations surrounding the use of AI tools in education. Ensuring the accuracy of generated content, mitigating potential biases, and maintaining student privacy are crucial aspects that educators must address when integrating AI-driven technologies into their curriculum. Moreover, staying up-to-date with the rapidly evolving field of AI and its potential applications in education is vital for leveraging these tools effectively and responsibly.

Future research should focus on assessing the practical implications of ChatGPT-based problem generation, evaluating its effectiveness, and developing best practices that educators can adopt globally. By doing so, the academic community can harness the full potential of advanced language models like ChatGPT, ultimately creating more engaging and stimulating learning experiences for computer science students.

REFERENCES

- [1] "ChatGPT," *OpenAI*. <https://chat.openai.com/chat?model=gpt-4> (accessed Mar. 01, 2023).
- [2] N. M. S. Surameery and M. Y. Shakor, "Use Chat GPT to Solve Programming Bugs," *Int. J. Inf. Technol. Comput. Eng.*, no. 31, pp. 17–22, Jan. 2023, doi: 10.55529/ijitc.31.17.22.
- [3] S. Kelly, "ChatGPT passes exams from law and business schools," Jan. 26, 2023. [Online]. Available: <https://www.cnn.com/2023/01/26/tech/chatgpt-passes-exams/index.html>

- [4] D. Baidoo-Anu and L. Owusu Ansah, "Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning," *SSRN Electron. J.*, 2023, doi: 10.2139/ssrn.4337484.
- [5] J. Rudolph, S. Tan, and S. Tan, "ChatGPT: Bullshit spewer or the end of traditional assessments in higher education?," *J. Appl. Learn. Teach.*, vol. 6, no. 1, Jan. 2023, doi: 10.37074/jalt.2023.6.1.9.
- [6] M. Shidiq, "The Use of Artificial Intelligence-Based ChatGPT and its Challenges for the World of Education; From the Viewpoint of the Development of Creative Writing Skills".
- [7] "Project Euler." <https://projecteuler.net/> (accessed Mar. 01, 2023).
- [8] "LeetCode." <https://leetcode.com/> (accessed Mar. 01, 2023).
- [9] "HackerRank." <https://www.hackerrank.com/> (accessed Mar. 01, 2023).
- [10] "Coursera." <https://www.coursera.org/> (accessed Mar. 01, 2023).

AUTHORS

Tyler Flaagan is an Assistant Professor at Dakota State University, teaching in the Cyber Operations and Network and Security Administration degrees. Tyler holds a Ph.D. and B.S. in Cyber Operations and an M.S. in Computer Science. Tyler is also the Director of the Deep Red Lab within Dakota State University MadLabs. Besides degrees, Tyler holds multiple professional certifications specializing in offensive security, such as Offensive Security Certified Professional and GIAC Exploit Researcher and Advanced Penetration Tester. Previously, Tyler worked as a red team operator for Naval Information Warfare Center Atlantic, supporting various projects within the Department of Defense.