Issues in Information Systems
Volume 24, Issue 2, pp. 203-212, 2023

DOI: https://doi.org/10.48009/2_iis_2023_117

Assessment of ChatGPT-generated programming code based on exercises in an introductory programming course

Azad Ali, *Indiana University of Pennsylvania, azad.ali@iup.edu*
Kustim Wibowo, *Indiana University of Pennsylvania, kwibowo@iup.edu*

Abstract

This study aims to assess the quality of the programming code generated by ChatGPT and compare it to the programming code used in an introductory programming course. ChatGPT is a software that produces programming code based on descriptions entered by users. Our study compares this generated code with a code we use in our programming course for the same programming exercises. ChatGPT is not a single software, and it is built from scratch. Instead, ChatGPT is built on a multitude of software that all contribute to the development of ChatGPT. This multitude of technologies falls under a general term called "OpenAI". To better understand the generated code and the assessment we intend to provide in this study, knowing these technologies that preceded ChatGPT and contributed to its formation will be helpful. Thus, we provide a literature review of the technologies that started before ChatGPT and then compare the code generated by ChatGPT with the code that we use in our course for the same assignments or problems.

Keywords: ChatGPT, programming, generated programming code, programming code

Introduction

Open AI is a term that is repeated and often exchanged these days due to the power that it generates and the commanding applications it produces. It is often noted in terms of practical uses. Still, it can also be reported in disruptive applications like imitating human voices for malicious use or when it makes cheating easier (Lo, 2023). Among the applications that OpenAI is noted is about generating programming code based on descriptions written in human language (Aydın & Karaarslan, 2022). This software that generates programming code based on human-like reports is ChatGPT (Chat Generative Pre-training Transformer). It generates a lot of buzz regarding the power with which it works, the accuracy of the code it generates, and the clarity of description it provides (Biswas, 2023). However, we did not find sources that compare and assess the code that it generates with code generated in the introductory programming course.

The purpose of this paper is to assess the code that is generated from ChaptGPT with code that is used in introductory programming courses. The paper uses the same description, feeds it into ChaptGPT, and studies the generated code. Then, the paper compares it with the code used in the textbook for the same exercise. After the comparison, we assess the generated code from ChatGPT.

Although the steps deal with assessing the code, it will be helpful to understand the technological advancement that led to the development of this powerful tool called ChatGPT. Thus, we will start with a literature review of the technologies involved that led to the development of ChaptGPT code generation.

Literature review

This section reviews the literature about ChatGPT and the associated technologies. The technology of ChatGPT and its ability to generate programming code did not start in isolation. Instead, it was built on multiple previous technologies that led to the final stages of ChatGPT as we know it now. The multitude of technologies that led to the development of ChatGPT include Artificial intelligence, machine learning/deep learning, natural language processing (Or NLP), and programming code generation (Ali, et al., 2023). Figure 1 below shows the technologies that led to the development of ChatGPT. The remainder of this section elaborates on each of these technologies.

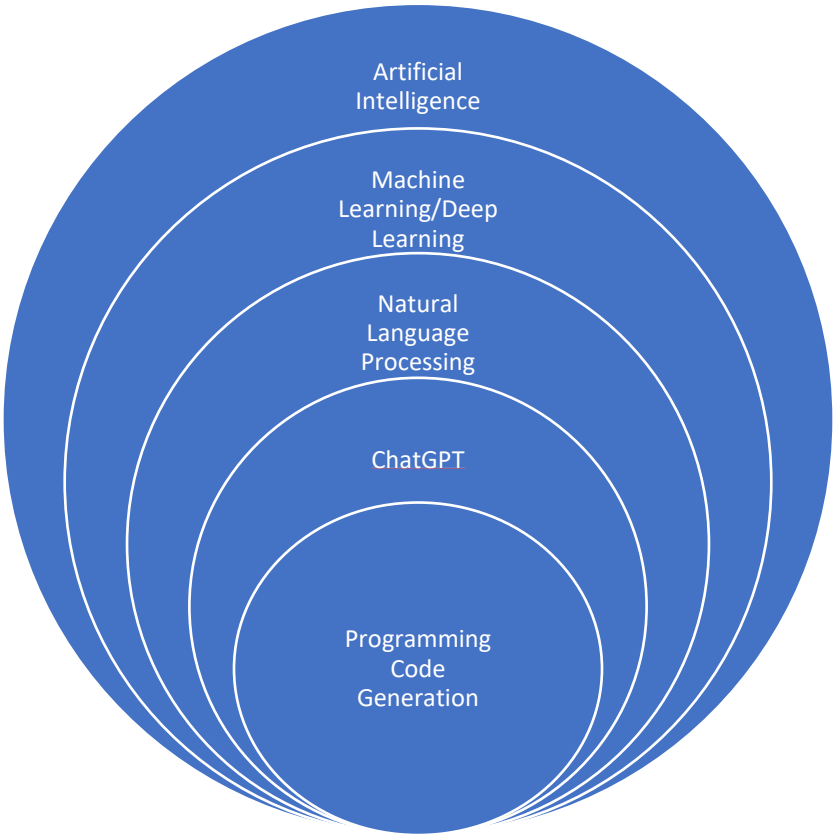

Figure 1: Technologies Included in of ChatGPT Program Generation (Aggarwal et al., 2022)

About artificial intelligence

Artificial intelligence (AI) is a field studied in computer science programs for years that was established as an academic discipline in the 1950s (Haenlein & Kaplan, 2019). The primary purpose of AI as a field of study was (and still is) to make the computer think like a human being. This field of study has grown and extended to various aspects of creating a computer program think intelligently like a human being. McCarthy (2007) explained more about artificial intelligence as:

> It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable (p. 2).

AI continued to be taught at various CS programs of study, but the content of their courses remained obscure for some time (Haenlein & Kaplan, 2019). For a long time, AI was thought to be more associated with robots that could replace human functions (Aggarwal et al., 2022). But the progress in processing large data and the development of "big data" as a field of study led to empowering the base at which AI was being developed (Haenlein & Kaplan, 2019).

AI took a base in computer education when it was used in large projects in the 1980s (Mijwel, 2015). Also, AI continued to be used in the medical field at various levels (Haleem, et al., 2019). But AI took a sharp step forward with the increasing capabilities of computers to process large volumes of data and with the development of a field named "big data analytics" (Aggarwal et al., 2022). With this increased power of computers to handle large volumes of data, it was thought that lessons could be learned from this data and the machine (the computer) could learn from this large volume of data and thus the development of Machine Learning as a field of study (Fradkov, 2020).

Machine learning

Aggarwal et al., (2022) explained that machine learning means that machines (computers) can learn based on patterns of data that are fed to them, and based on this large volume of data, they can predict future trends. This kind of trend could be collected from texts, images, videos, emails, web surfing, social media, or personal data collected on the individual. As more data is available to them, computers can see data patterns which help predict the next step.

Deep Learning

Deep learning is a step above what is learned about machine learning. It refers to understanding the behavior of humans based on the big data collected from dealing with the individual. Deep learning means that learning builds on previous learning that leads to a higher understanding of human behavior. It can also be called a higher level of learning for human behavior, enabling the machine to predict the next action to take.

Deep learning can be generated based on long interactions between humans and computers. As individuals browse the web and deal with videos, audio, images, social media, emails, and others, this generates a large volume of data that computers can analyze. After the analysis of data comes the understanding and the "deep learning "which is a deeper level of knowledge by the computer to the human interaction (Aggarwal, et al., 2022)

The development of Natural Language Processing

Natural language processing (or NLP) means that individuals can talk to computers like we are talking to each other in our spoken language. The computer then answers our questions based on the large database it accesses. In NLP, there is typically no form to fill in, no options to reply to, and no buttons to press, as in previous applications. This is a departure from what is known about computer interaction. When we need to get some output or feedback from the computer, we often have to fill out a form, select from options,

answer questions, and probably the answer will not be direct. Instead, the answer will include many links and a lot of information that is not directly related to the topic being searched.

IBM (2023) explained that the power of NLP stems from the following points:

- Translate text from one language to another
- Respond to spoken commands
- Summarize large volumes of text rapidly even in real-time.

NLP drives the power of machine learning and deep learning models. Both models enable computers to process our spoken language in spoken or written form to understand the full meaning of the words spoken to it or the text entered into the computer.

About ChatGPT

Aljanabi (2023) called ChatGPT "Cutting-edge language" and described it as is exciting advancement of AI. The major power distinguishing ChatGPT is the ability to engage in conversations with humans, understand conversations from humans, and generate text understandable by humans to provide solutions and answers to questions asked. It is an automated computerized interaction that can be exchanged between users and the machine. The trick here is the speed at which it produces the results and the answer to the question. Aljanabi et al., further noted that ChatGPT provides a summary of papers, extracts key points, and even provides citations. It can provide text for academic documents such as papers, essays, and dissertations.

Ali et al., (2023) studied the impact of ChatGPT on learning motivation from teachers and students' voices. It found that ChatGPT motivates various parties to complete tasks. Surameery and Shakor (2023) listed some advantages of using ChaptGPT, which include efficiency, higher precision, and reduced costs.

ChatGPT and Programming Code

Among the notable capabilities of ChatGPT is the ability to produce programming code based on the description of human languages. This kind of code generation from ChatGPT has become a subject of wide recognition of programming strength Chen et al., (2023) suggested that there is some software called GPTutor that generates code explanations and is efficient about it.

Although there were some concerns that ChatGPT could be used for cheating by students (Lo, 2023), the general trend is to view this as a powerful tool for generating programming code. Tian et al., (2023) for example, suggested that ChatGPT is considered the ultimate programming assistant and questioned how far it can go.

Tian et al., (2023) noted that it is the long-standing dream of software engineering: to repair software automatically with minimal human intervention. And that LLM (Large Language Model) has a bright future. Biswas (2023) suggested that ChatGPT can be used for computer programming for various tasks, including

- Code generation and correction
- Document generation
- Chatot development

- Text-to-code generation
- Answering technical queries

Biswas (2023) noted that ChatGPT can be integrated into the programming environment to improve developer productivity and speed up the coding process. Gewortz and Windsor (2023) suggested that ChatGPT can generate code in multiple programming languages, including PHP, Python, Java, Kotlin, Swift, C#, COBOL, Fortran, Forth, LISP, ALGOL, RPG, and others.

- The purpose of this paper is to assess the quality of programming code generated by ChatGPT based on instructions given in textbooks taught in introductory programming courses. To achieve this purpose, we are going to follow these steps for completing our assessment:
- We enter the programming exercise instruction from the textbook we teach in our introductory programming course into ChatGPT to generate the programming code.
- We examine the code generated by ChatGPT to check for accuracy.
- We compare the generated code with the solution provided by the textbook.
- We make our assessment based on the comparison between the generated code from ChatGPT and the code the textbook provides.

To make the assessment more comprehensive, we will examine it based on three examples we selected from the textbook. The first example is going to be a program from the first few chapters of the textbook. We then use an intermediate-level program from the middle chapters of the textbook. Last, we give it another program based on the last few chapters where the problem gets more advanced. By following this approach, we can better assess the strength of ChatGPT and the accuracy of the code that it generates based on different levels of program complexity in our introductory programming course.

In this paper, we will not approach the issue of program correction. Different sources explained and tested that ChatGPT can correct programming code and does it efficiently (Aljanabi et al., 2023; Tian et al., 2023). While syntax errors are easier to find, and most modern editors are very helpful in locating and correcting syntax errors, ChatGPT also helps find logical errors (Surameery & Shakor, 2023).

Comparing the code

This section presents the exercises we submitted to ChatGPT based on the textbook we selected for our introductory programming course (Lambert, 2019). We aim to compare the code generated from the textbook versus what ChatGPT gives for the same description. We selected three programming exercises from Lamber's textbook: program 1.7, program 7.1, and Program 11.1 and the results from exercise are presented in tables 2 through XX. For each of the three programs we present the program description, the suggested solution by the textbook and by ChatGPT, the program explanation from ChatGPT, and our assessment. The following lists the specific items presented for each problem:

A. Lists exercise explanation from the textbook
B. List side-by-side two-column solutions for the exercise. First column, the solution suggested by the textbook. The second column lists the programming solution presented by ChatGPT.
C. It lists the program explanation suggested by ChatGPT
D. List our assessment after we reviewed the textbook and ChaptGPT solutions

First comparison example - Programming exercise 1.7

The first example was based on program 1.7 from Lambert's (2019) textbook. Table 2A presents the program explanation in the textbook. Table 2B presents the code generated from the textbook versus what is generated from ChatGPT. We put them in side-by-side columns to make the comparison easier. Then in Table 2C, we present the explanation generated by ChatGPT, and finally, in Table 2D we present our assessment by comparing program 1.7 in the textbook.

Table 2A: Explanation of the Problem 1.7 in Lambert's Textbook

Problem 1.7 as explained in Lambert's textbook	Write and test a program that accepts the user's name (as text) and age (as a number) as input. The program should output a sentence containing the user's name and age

Table 2B: Comparing the Solution of Lambert's Textbook With ChatGPT's Suggested Solution

The solution suggested in the textbook by Lambert	The solution suggested by chatGPT
name = input("Enter your name: ") years = int(input("Enter your age: ")) print(name, "is", years, "years old.")	name = input("Enter your name: ") age = int(input("Enter your age: ")) output = "Your name is {} and you are {} years old.".format(name, age) print(output)

Table 2C: Presenting the Explanation by ChatGPT

The explanation is given by ChatGPT	When you run this program, it will prompt you to enter your name. Once you provide your name and press Enter, it will then ask you to enter your age. After you enter your age and press Enter, the program will output a sentence that combines the entered name and age.

Table 2D: Our Assessment of the Solution Presented by ChatGPT to Lambert's Problem 1.7

Our Assessment	The solution suggested by ChatGPT sufficiently addresses the problem, it provides code that answers the question and provides a clear explanation.

Second comparison example- Programming exercise 7.1

Tables 3A thru 3D presents the assessment for the programming exercise 7. 1. Table 3A presents the program explanation in the textbook. Table 3B presents the code generated from the textbook versus what is generated from ChatGPT. We put them in side-by-side columns to make the comparison easier. Then in Table 3C, we present the explanation generated by ChatGPT, and finally, in Table 3D we present our assessment by comparing program 7.1 in the textbook.

Table 3A: Explanation of the Problem 7.1 in Lambert's Textbook

Problem 7.1 as explained in Lambert's textbook	Define a function drawCircle. This function should expect a Turtle object, the coordinates of the circle's center point, and the circle's radius as arguments. The function should draw the specified circle. The algorithm should draw the circle's circumference by turning 3 degrees and moving a given distance 120 times. Calculate the distance moved with the formula 2.0 × π × radius ÷ 120.0.

Table 3B: Comparing the Solution of Lambert's Textbook With ChatGPT's Suggested Solution

The solution suggested in the textbook by Lambert	The solution suggested by chatGPT

import math
from turtle import Turtle
def drawCircle(t, x, y, radius):
 	
  t.up()
  t.goto(x + radius, y)
  t.setheading(90)
  t.down()
  for count in range(120):
  t.left(3)
  t.forward(2.0 * math.pi * radius / 120.0)
def main():
  x = 50
  y = 75
  radius = 100
  drawCircle(Turtle(), x, y, radius)
 	
if __name__ == "__main__":
  main()
	import turtle
	import math
	
	def drawCircle(turtle_obj, center_x, center_y, radius):
	  circumference = 2.0 * math.pi * radius
	  distance = circumference / 120.0
	
	  turtle_obj.penup()
	  turtle_obj.goto(center_x, center_y - radius)
	  turtle_obj.pendown()
	
	  for _ in range(120):
	  turtle_obj.forward(distance)
	  turtle_obj.right(3)
	
	# Example usage:
	my_turtle = turtle.Turtle()
	drawCircle(my_turtle, 0, 0, 100)
	turtle.done()

Table 3C: Presenting the Explanation by ChatGPT

The explanation is given by ChatGPT	In this function, we import the turtle module and the math module for mathematical calculations. The function takes a turtle_obj (which is an instance of the Turtle class), center_x, and center_y as the coordinates of the circle's center point and radius as the circle's radius.

Table 3D: Our Assessment of the Solution Presented by ChatGPT to Lambert's Problem 4.10

Our Assessment	ChatGPT and Cengage's solutions are identical. Both import "math" and "turtle" modules. However, both are using different sets of "turtle" functions.

Third comparison exercise – Programming exercise 11.1

Tables 4A thru 4D presents the assessment for the programming exercise 11.1 Table 4A presents the program explanation in the textbook. Table 4B presents the code generated from the textbook versus what is generated from ChatGPT. We put them in side-by-side columns to make the comparison easier. Then in Table 4C, we present the explanation generated by ChatGPT, and finally, in Table 4D we present our assessment by comparing program 11.1 in the textbook.

Table 4A: Explanation of the Problem in Lambert's Textbook

Problem 11.1 as explained in Lambert's textbook	A sequential search of a sorted list can halt when the target is less than a given element in the list. Define a modified version of this algorithm, and state the computational complexity, using big-O notation.

Table 4B : Comparing the Solution of Lambert's Textbook With ChatGPT's Suggested Solution

The solution suggested in the textbook by Lambert	The solution suggested by chatGPT

def sequentialSearch(target, lyst):
  position = 0
  while position < len(lyst):
  if target == lyst[position]:
  return position
  elif target < lyst[position]: # Target less, so it
  return -1 # can't be in sorted list
  position += 1
  return -1
 	
def main():
  """Tests with three lists."""
  print(sequentialSearch(3, [0, 1, 2, 3, 4]))
  print(sequentialSearch(3, [0, 1, 2]))
  # Will stop at second position.
  print(sequentialSearch(3, [0, 4, 5, 6]))
 	
if __name__ == "__main__":
  main()
	function early_exit_sequential_search(list, target):
	  i = 0
	  while i < length(list) and list[i] <= target:
	  if list[i] == target:
	  return i # Found the target
	  i = i + 1
	
	  return -1 # Target not found

Figure 4C - Presenting the Explanation by ChatGPT

The explanation is given by ChatGPT	A modified version of the sequential search algorithm that halts when the target is less than a given element in the sorted list is called the "Early Exit Sequential Search" algorithm. This algorithm takes advantage of the fact that the list is sorted and terminates the search early when the target value is determined to be less than the current element being examined.

Figure 4D - Our Assessment of the Solution Presented by ChatGPT to Lambert's Problem 4.10

Our Assessment	ChatGPT and Cengage's solutions are identical. However, ChatGPT provides only the search algorithm and Cengage provides a complete code (with main()).

Limitations, Conclusions, and Future Research

We started this paper by reviewing literature about artificial intelligence and its related technologies, including machine learning, deep learning, and natural language processing. Then, it extended to include

ChatGPT and our focus on programming code generation from instructions given in a typical introductory programming course.

After we reviewed the generated code from different levels of programming exercises that we give in our courses, we reached the following conclusions:

- ChatGPT generated well-designed code for the first few exercises that can be used to teach students about these programs.
- The explanations that ChatGPT generated are easy to understand and they are comparable to the explanation that is given in textbooks and similar programming articles.
- When assigning more advanced programs, we found some challenges and inconsistencies. This happens when asking questions requiring writing more than one file. We found that ChatGPT generates only one file in separate exercises that we tested and are not included in this study. However, in different cases, generating code in multiple files and then linking them together will be more efficient. So, a significant setback to the issue of generating code is that so far, we were not able to find a way in ChatGPT to generate code in more than one file. Thus, it will be harder to teach modularization with the code generated by ChatGPT.

We want to add that although our paper focused on code generation by ChatGPT, we also made some minimum testing to check the ChatGPT to fix errors. Yes, ChatGPT can effectively fix coding errors. This could be syntax errors or logical errors. ChatGPT will provide practical suggestions to fix the errors and enhance the logic of the program.

Based on the literature we reviewed and our testing of the generating code, we want to give two thumbs up to ChatGPT generated code. We conclude that ChatGPT generates programming code for courses that teach introductory programming to a certain level. This level gets to the point when developing or breaking down programs into modules. At these points, we could not have ChatGPT break down the program into modules as suggested by many textbooks that it is the effective way to teach programming and as we learned from our teaching of this course.

We suggest starting our next study from this point – to find a way for ChatGPT to generate code in more than one file to follow modularization and enhance the program's efficiency. We also want to include longer, more complicated programs with interfaces to different applications. We include all these topics in our courses and want to assess how much ChatGPT can generate code for these applications. But so far, our assessment for this test is that ChatGPT generates efficient code and provides clear explanations. For more advanced programs, we will wait until our subsequent study to determine how efficiently ChatGPT generates code for more advanced programming requirements.

References

Aggarwal, K., Mijwil, M. M., Al-Mistarehi, A. H., Alomari, S., Gök, M., Alaabdin, A. M. Z., & Abdulrhman, S. H. (2022). Has the future started? The current growth of artificial intelligence, machine learning, and deep learning. *Iraqi Journal for Computer Science and Mathematics*, *3*(1), 115-123.1

Ali, J. K. M., Shamsan, M. A. A., Hezam, T. A., & Mohammed, A. A. (2023). Impact of ChatGPT on Learning Motivation: Teachers and Students' Voices. *Journal of English Studies in Arabia Felix*, *2*(1), 41-49.

Aljanabi, M. (2023). ChatGPT: Future directions and open possibilities. *Mesopotamian Journal of Cybersecurity*, *2023*, 16-17.

Aljanabi, M., Ghazi, M., Ali, A. H., & Abed, S. A. (2023). ChatGpt: Open Possibilities. *Iraqi Journal For Computer Science and Mathematics*, *4*(1), 62-64.

Aydın, Ö., & Karaarslan, E. (2022). OpenAI ChatGPT generated literature review: Digital twin in healthcare. *Available at SSRN 4308687*.

Biswas, S. (2023). Role of ChatGPT in Computer Programming.: ChatGPT in Computer Programming. *Mesopotamian Journal of Computer Science*, *2023*, 8-16.

Chen, E., Huang, R., Chen, H. S., Tseng, Y. H., & Li, L. Y. (2023). GPTutor: a ChatGPT-powered programming tool for code explanation. *arXiv preprint arXiv:2305.01863*.

Fradkov, A. L. (2020). Early history of machine learning. *IFAC-PapersOnLine*, *53*(2), 1385-1390.

Gewortz, D., & Windsor, A. (2023, May 5th). How to use ChatGPT to write code. ZDNet Innovation. Retrieved May 19, 2023 from https://www.zdnet.com/article/how-to-use-chatgpt-to-write-code/

Haenlein, M., & Kaplan, A. (2019). A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California management review*, *61*(4), 5-14.

Haleem, A., Javaid, M., & Khan, I. H. (2019). Current status and applications of artificial intelligence (AI) in medical field: An overview. *Current Medicine Research and Practice*, *9*(6), 231-237.

IBM (2023). What is natural language processing (NLP). Retrieved May 21, 2023 from https://www.ibm.com/topics/natural-language-processing

Lambert, K. A.(2019). *Fundamentals of Python First Programs, 2nd Edition*. Cengage Learning.

Lo, C. K. (2023). What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature. *Education Sciences*, *13*(4), 410.

McCarthy, J. (2007). From here to human-level AI. *Artificial Intelligence*, *171*(18), 1174-1182.

Mijwel, M. M. (2015). History of Artificial Intelligence Yapay Zekânın T arihi. *no. April*, *2018*.

Surameery, N. M. S., & Shakor, M. Y. (2023). Use chat gpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290*, *3*(01), 17-22.

Tian, H., Lu, W., Li, T. O., Tang, X., Cheung, S. C., Klein, J., & Bissyandé, T. F. (2023). Is ChatGPT the Ultimate Programming Assistant--How far is it?. *arXiv preprint arXiv:2304.11938*.