Questions about Code Generation AI Models: General Perspective

Thank you for taking the time to consider our study.

Title of Research Study: Code Generation AI Models and User Needs

Principal Investigator, Title, and Affiliation: Professor Dr. Shafay Shamail, Lahore University of Management Sciences

Purpose, Procedure, and Duration: AI based code generation models like Copilot, ChatGPT, Gemini Code Assist, etc. are widely used by coders or software developers and actively maintained and enhanced by model engineers. In this survey we will present some questions that are aimed to better understand the why's and how's of code generation. You will be asked to indicate if the presented questions are relevant to you for **better understanding and trusting the code or the AI model's decision making**. The survey may take about 10 minutes to complete.

Eligibility: If you are either using some code generation AI models for code reuse or working on training or updating these AI models, your feedback can be very useful to understand how we can improve the models.

Benefits of Study: There is no monetary compensation provided to the participants. The participants can benefit the wider scientific community through their participation and by providing accurate evaluations.

Possible Risks of Study: There are no anticipated risks or adverse effects of this study.

Privacy and Future Use: Your responses to the survey are anonymous. That means we won't know which responses are yours. We won't collect names, internet addresses, email addresses, or any other identifiable information. We may use your responses in future research or share them with other researchers.

Contact:

If you have any questions/clarifications on this study, please contact the Co-Investigator Dr. Shamsa Abid at shamsa.abid@lums.edu.pk

If you have any questions or concerns regarding your rights as a participant in this research study and wish to contact someone unaffiliated with the research team, please contact the LUMS Institutional Review Board Convener at opsteam@lums.edu.pk. When contacting LUMS IRB, please provide the title of the Research Study and the name of the Principal Investigator, or quote the IRB approval number (IRB-0397).

Participant's Declaration

I understand that participation is voluntary. Refusal to participate will involve no penalty.

I declare that I am at least 18 years of age.

I have read and fully understood the contents of this form, and hereby give consent to the Lahore University of Management Sciences research team and its affiliates for this project to collect and/or use my data for the purpose(s) described in this form.

* Indicates required question

1. Please select "Yes" to indicate you have read this information and you accept to take the survey: *

Mark only one oval.

- ◯ Yes

Demographics data

2. Please indicate your education subject area *

Mark only one oval.

- ◯ Computer Science
- ◯ Software Engineering
- ◯ Artificial Intelligence
- ◯ Data Science
- ◯ Computer Engineering
- ◯ Electrical Engineering
- ◯ Mathematics
- ◯ Other: _____

3. Please indicate your current education level *

Mark only one oval.

- ◯ Undergraduate
- ◯ Bachelors Degree
- ◯ Enrolled in Masters
- ◯ Masters Degree
- ◯ Enrolled in PhD
- ◯ PhD Degree
- ◯ Other: _____

4. Please Indicate your current role *

Tick all that apply.

- ☐ Student
- ☐ Researcher
- ☐ Software Engineer/ Developer/ Programmer
- ☐ Software Quality Engineer/ Tester
- ☐ ML Engineer/ AI Specialist
- ☐ Assistant/ Associate/Full Professor
- ☐ Other: _____

5. Please indicate your interaction with code generation models *

Tick all that apply.

- ☐ I use AI code generation tools for writing code only
- ☐ I develop or train AI models for code generation
- ☐ I am both using and engineering AI code generation models
- ☐ I evaluate the code generation models for trust and transparency
- ☐ Other: _____

6. Please indicate the time period during which you have been working with AI tools/models for code generation. *

Mark only one oval.

- ◯ Less than 2 months
- ◯ 2 to < 6 months
- ◯ 6 to < 12 months
- ◯ 1 to < 2 years
- ◯ 2 to < 5 years
- ◯ 5 years or more

7. Which AI code generation models do you use or interact with *

Tick all that apply.

- ☐ GitHub Copilot
- ☐ Tabnine
- ☐ Codeium
- ☐ Amazon CodeWhisperer
- ☐ Replit Ghostwriter
- ☐ Cursor
- ☐ ChatGPT (GPT-4, GPT-4o)
- ☐ Claude (Claude 3.5)
- ☐ Google Gemini
- ☐ Mistral AI / Codestral
- ☐ Meta Code Llama
- ☐ StarCoder / StarCoder2
- ☐ PolyCoder
- ☐ SantaCoder
- ☐ DeepSeek Coder
- ☐ IntelliCode
- ☐ JetBrains AI Assistant
- ☐ Other: _____

Questions about Code Generation AI Models: General Perspective

Please indicate if the following questions are relevant for **better understanding and trusting the code or the model's decision making**. (1=strongly irrelevant, 2= not relevant, 3= neutral, 4= relevant , 5= strongly relevant)

8. What kinds of code artifacts (e.g., functions, scripts, projects) were used to train the model? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

9. Were the training code examples sourced from open-source repositories e.g., GitHub, Stack Overflow, textbooks, or proprietary datasets? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

10. For supervised training tasks e.g., code summarization, translation, how were ground-truth labels or outputs defined and validated? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

11. How many code files or code snippets were used to train the model? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

12. Are there any known or commonly used code datasets e.g., HumanEval, CodeSearchNet that were explicitly excluded from training? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

13. Does the training data favor specific programming languages, styles, domains e.g., web, systems, or coding conventions that may bias the model's output? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

14. What is the distribution of code samples by language, code length, project type, or complexity in the training dataset? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

15. What kind of code outputs can the model generate—functions, full programs, configuration files, documentation, or test cases? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

16. How should I interpret the generated code—does it represent a complete, executable solution or just a code snippet meant to be extended or integrated? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

17. What code-related tasks can the model handle (e.g., code translation, completion, debugging, refactoring), and where are its known limitations? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

18. How can the generated code be integrated into my existing development pipeline or tools (e.g., compilers, test frameworks, CI/CD)? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

19. What are the best practices for reviewing, testing, or modifying the generated code before deploying it in production or submitting it to version control? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

20. How can this model-generated code be used efficiently within my software development workflow (e.g., prototyping, learning, accelerating repetitive tasks)? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

21. How accurate, syntactically correct, and semantically valid is the generated code? How reliable is it across different code tasks (e.g., translation, completion, test generation)? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

22. How frequently does the model produce incorrect, incomplete, insecure, or non-compiling code? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

23. Under what input conditions or code tasks (e.g., simple utility functions vs. complex algorithms) is the model more or less likely to produce correct outputs? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

24. What are the known weaknesses of the model—e.g., poor handling of rare languages, advanced design patterns, or domain-specific APIs? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

25. Does the model tend to introduce logic bugs, misuse libraries, ignore edge cases, or produce non-idiomatic code? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

26. Is the model suitable for use in production-level code, educational purposes, rapid prototyping, or legacy code migration? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

27. How does the neural network process source code or natural language input to generate corresponding code output? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

28. What syntactic or semantic code features e.g., keywords, function signatures, code structure are important in guiding predictions? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

29. Does the model consider [feature X], such as a specific variable, comment, or function call, in generating the output? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

30. What are the high-level decision patterns or learned associations the model uses to predict code completions or transformations? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

31. How much influence do different code tokens, AST nodes, or input types e.g., comments vs. code have on the model's output? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

32. Are there implicit patterns or rules the model tends to replicate, such as standard library usage or code idioms? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

33. If we modify or remove [feature X], how is the generated code affected? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

34. What are the most influential input patterns, syntax structures, or tokens that affect generation behavior? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

35. What neural architecture underlies the model e.g., Transformer, LSTM? How does this affect its capacity to understand code? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

36. What training strategies e.g., fine-tuning, pretraining and hyperparameter settings were used? What data was the model trained on? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

37. Why did the model generate this specific code snippet for the given prompt or partial code? How did it arrive at this particular output among possible alternatives? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

38. What aspects of the input (e.g., function name, comments, code context, natural language intent) influenced the model's code generation most? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

39. Why did the model generate the same or similar code for two different prompts or inputs (e.g., slightly different phrasing or variable names)? What common patterns led to this convergence? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

40. Why didn't the model generate a more efficient (or idiomatic, or expected) version of the code? Why was a particular API or construct not used in the generated code? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

41. Why did the model choose this code structure or algorithm over another possible one (e.g., iteration vs recursion, quicksort vs mergesort)? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

42. Why did similar input prompts/code snippets result in different outputs? What differences in input caused the model to generate distinct code? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

43. How should I rephrase the prompt or modify the code snippet to generate a different kind of implementation (e.g., use a different library, algorithm, or coding style)? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

44. What is the smallest modification to the input (e.g., parameter, comment, or keyword) needed to generate a significantly different version of code (e.g., with optimization, better readability, or fewer dependencies)? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

45. How should I change a specific part of the input—like the function name, data type, or comment—for the model to generate code using a different method or abstraction? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

46. What kind of prompt or code context typically leads the model to output more robust, idiomatic, or secure code compared to the one it gave now? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

47. What changes can I make to the prompt or input code (e.g., reordering comments, using synonyms, changing formatting) that still result in the same generated code or logic? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

48. How much can I vary elements like variable names, input types, or function structure before the model starts generating different code logic or structure? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

49. What specific parts of the input (e.g., algorithm name, data structure, comment cues) are critical for ensuring that the generated code performs the same operation or uses the same technique? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

50. What types of prompts or code snippets typically result in the same or similar generated code (e.g., same logic with different wording)? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

51. What kind of code would be generated if I changed the input prompt slightly—such as modifying the problem description, function name, or input constraints? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

52. How would the generated code differ if I specified a different data structure (e.g., use a list instead of a dictionary) or requested an iterative solution instead of a recursive one? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

53. What kind of code will be generated if I input a completely different coding problem or ask for a solution in a different programming language? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

54. How will the code generation model evolve over time—e.g., with updates, new training data, or fine-tuning? Will it improve in generating idiomatic code or adapting to newer libraries and language features? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

55. Can users or developers fine-tune or customize the model for a specific codebase, domain, or team coding standards? If yes, how can I do that safely and effectively? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

56. Why did the model choose a particular algorithm (e.g., BFS vs DFS) or avoid using a specific feature or library in the generated code? What data or training influenced that choice? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

57. What does it mean when the model output explanation says “attention weight,” “activation,” “temperature,” or “beam search”? How does that relate to the generated code?

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

58. What kind of experiences or typical outcomes have other developers had when using this model for code generation? Are there known issues, best practices, or communities sharing insights? *

Mark only one oval.

	1	2	3	4	5	
Not	◯	◯	◯	◯	◯	Strongly relevant

Thanks for your feedback!

We would like to know your thoughts on this survey!

59. Any thoughts ....