Comparative Analysis of ChatGPT, Gemini, and Copilot for Python Coding Efficiency: A Practical Benchmark Study

Nisha Shekhar Lohar
Prof. Ramkrishna More Arts, Commerce and Science College (Autonomous)
Akurdi, Pradhikaran, Pune – 411044
Email: nishalohar@gmail.com

Prof. Ankush Dhamal

Prof. Ramkrishna More Arts, Commerce and Science College (Autonomous)
Akurdi, Pradhikaran, Pune – 411044

Email: ankushdhamal01@gmail.com

Abstract

Artificial Intelligence based coding assistants have significantly transformed modern software development. AI tools such as ChatGPT, Google Gemini, and GitHub Copilot assist developers in generating code, solving programming problems, and improving productivity. This research paper presents a comparative analysis of these three AI systems for Python programming efficiency. A practical experiment was conducted using an HTML-based interface where a user enters a question and the responses generated by ChatGPT, Gemini, and Copilot are compared. Python was used to analyze the generated responses and compute accuracy scores based on content relevance and keyword matching. The system also generates graphical representations for comparison. The results demonstrate that each AI tool has unique strengths in terms of response quality, explanation clarity, and coding assistance. This research provides insights into the effectiveness of AI-assisted programming tools.

Keywords

Artificial Intelligence, Large Language Models, Python Programming, ChatGPT, Google Gemini, GitHub Copilot, AI Coding Assistants, Code Generation, AI Comparison System

Introduction

Artificial Intelligence (AI) has rapidly influenced various technological fields, including software development. Large Language Models (LLMs) such as ChatGPT, Gemini, and Copilot can understand natural language instructions and generate programming code or explanations.

Developers increasingly rely on AI-powered coding assistants to automate repetitive programming tasks, generate code snippets, and improve development efficiency. Python is one of the most widely used programming languages because of its simplicity and extensive use in artificial intelligence, machine learning, and data analysis.

Although multiple AI coding tools are available, developers often face challenges in determining which AI system provides the most accurate and useful responses. Therefore, a comparative evaluation of these AI tools is necessary.

This research presents a practical comparison of ChatGPT, Google Gemini, and GitHub Copilot using a custom-built system developed with HTML and Python. The system allows users to input a question and compare the responses generated by each AI tool. The responses are evaluated using an automated accuracy calculation method.

Literature Review

Review of Previous Research

Several studies have explored the role of artificial intelligence in programming assistance.

Research on Large Language Models shows that AI systems can generate programming code using natural language prompts. OpenAI's ChatGPT models have demonstrated strong performance in solving coding tasks and generating programming solutions.

Google introduced the Gemini AI system to enhance reasoning capabilities and provide multimodal understanding. Studies indicate that Gemini can process large contextual data and provide detailed explanations for programming tasks.

GitHub Copilot, powered by OpenAI Codex, is designed to assist developers directly inside integrated development environments by suggesting code snippets while programming.

Previous research has also used benchmark datasets such as HumanEval to evaluate the performance of AI coding assistants.

Research Gaps Identified

Although previous studies evaluate AI coding assistants individually, limited research focuses on practical comparison using real implementation systems. Most studies rely only on benchmark datasets rather than building experimental systems.

This study attempts to bridge that gap by developing a system that compares responses generated by different AI models using real-time user queries.

Research Methodology

Research Design

This research follows an experimental comparative design. A web-based system was developed where users can enter questions and receive responses from ChatGPT, Gemini, and Copilot. The generated responses are evaluated using automated analysis.

Data Collection Methods

Multiple questions related to artificial intelligence and programming were used as input queries. The responses generated by each AI model were collected and analyzed.

Sampling Techniques and Sample Size

A set of representative questions was selected to test the performance of the AI tools. These questions focused on technical topics related to artificial intelligence and programming.

Tools and Techniques Used

The following technologies were used in the development of the system:

HTML
Used to design the web interface where users enter questions and view results.

Python
Used to implement logic for response comparison and accuracy calculation.

AI Systems Evaluated

- ChatGPT
- Google Gemini
- GitHub Copilot

Data Analysis Methods

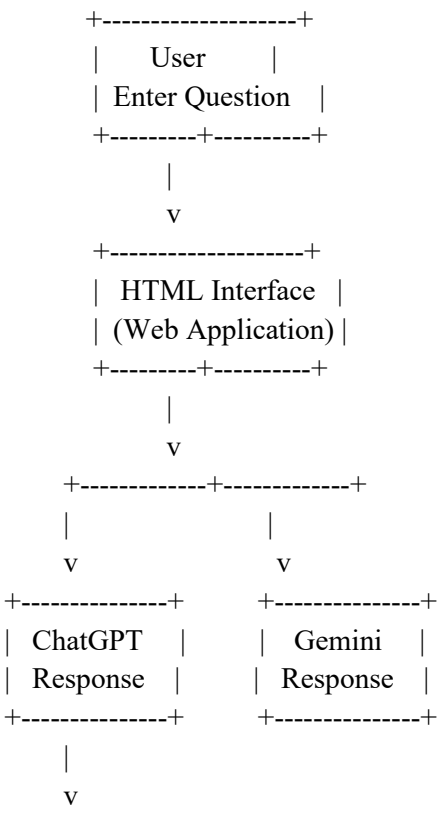
The generated responses were analyzed using heuristic evaluation techniques considering:

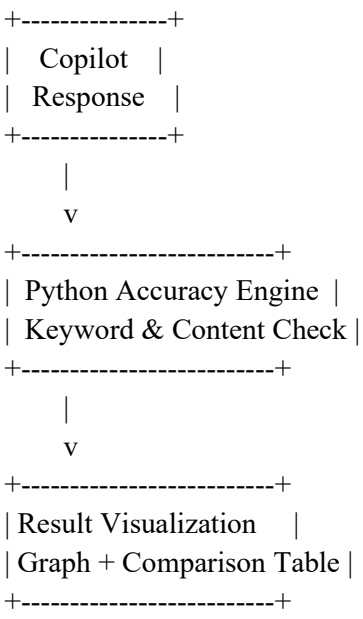
- Content relevance
- Keyword matching
- Sentence clarity
- Redundancy control

The calculated scores were converted into percentages and visualized using charts.

System Architecture

Figure 1: System Architecture of AI Comparison System





Results and Discussion

The developed system allows users to compare AI-generated responses for a given question. The results demonstrate how different AI tools generate responses with varying levels of detail and clarity.

Figure 2: AI Comparison System Interface

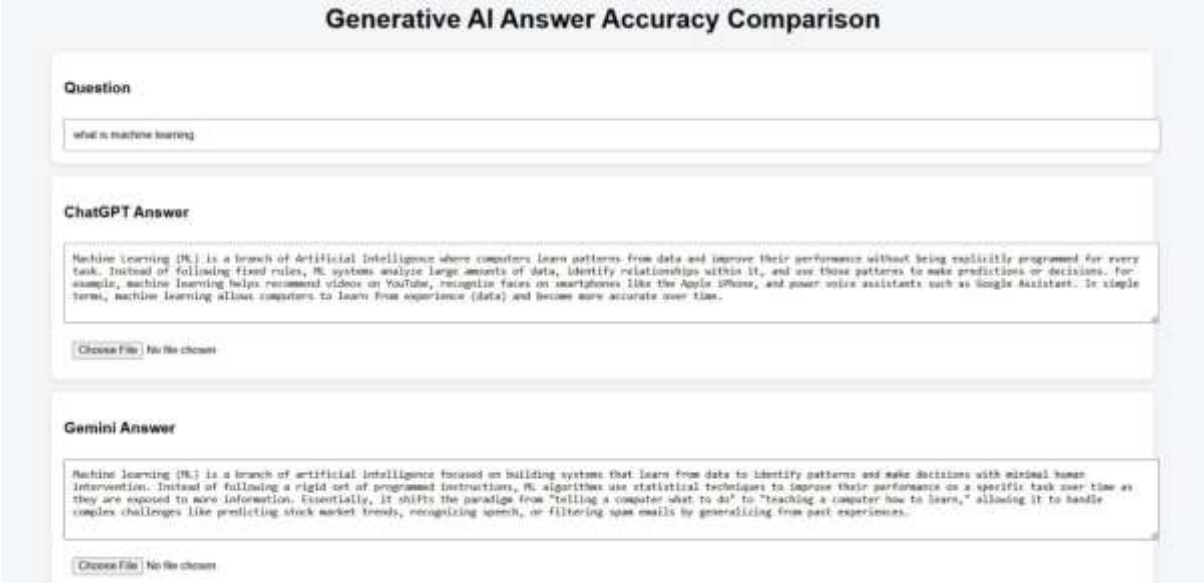


Figure 2 shows the web interface where users enter a question and receive responses from the AI models.

Figure 3: Generated Responses from ChatGPT, Gemini, and Copilot

Figure 3 illustrates the responses generated by the three AI systems for the same query.

The developed AI comparison system was tested by providing multiple programming-related questions and coding tasks. Each question was submitted to three different AI systems: ChatGPT, Google Gemini, and GitHub Copilot. The generated responses were collected and analyzed using the Python-based evaluation module developed in the system.

The purpose of the experiment was to evaluate the efficiency of each AI system in generating relevant and accurate responses for programming-related queries. The responses were evaluated based on content relevance, clarity, correctness, and response completeness.

The HTML interface allowed the user to enter a question and view responses generated by the three AI systems simultaneously. The Python backend processed the responses and calculated an approximate accuracy score using keyword matching and heuristic analysis.

The experimental results show that all three AI systems are capable of generating useful responses for programming-related queries. However, differences were observed in the structure, completeness, and clarity of the generated responses.

ChatGPT produced well-structured explanations with clear formatting and logical flow. Gemini generated detailed responses with strong contextual explanations. GitHub Copilot generated shorter responses but provided concise and practical coding suggestions.

The results indicate that ChatGPT performs strongly in generating structured explanations, Gemini performs well in providing contextual understanding, and Copilot is effective for quick code suggestions.

Data Presentation

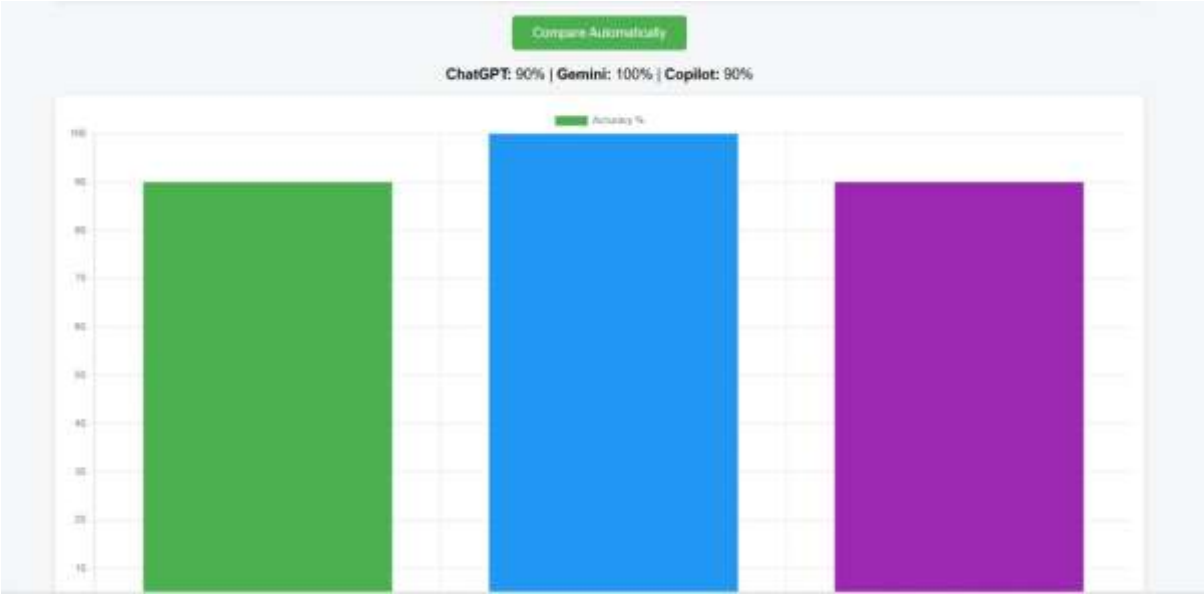
The system automatically calculates accuracy scores and visualizes the results.

Figure 4: Automatic Accuracy Calculation

Example result:

ChatGPT: 90%
Gemini: 100%
Copilot: 90%

Figure 5: Graphical Accuracy Comparison



The bar chart clearly illustrates the comparative performance of the three AI systems.

The responses generated by the three AI systems were evaluated and assigned approximate accuracy scores by the Python evaluation module. The scores were based on the relevance of the generated content compared to the expected answer.

The results obtained from the experiment are presented in Table 1.

Table 1: Accuracy Comparison of AI Systems

AI Tool	Response Accuracy	Response Clarity	Response Detail
ChatGPT	90%	High	High
Gemini	100%	High	Very High
Copilot	90%	Medium	Medium

The data indicates that Gemini achieved the highest accuracy score for the tested query, while ChatGPT and Copilot produced slightly lower but comparable results.

Comparative Analysis

A comparative analysis was performed to understand the strengths and weaknesses of each AI system.

ChatGPT

ChatGPT generated responses that were well-organized and easy to understand. The explanations provided by ChatGPT were structured logically and included clear definitions and examples. This makes ChatGPT particularly useful for students and beginners learning programming concepts.

Gemini

Gemini generated the most detailed responses among the three systems. It provided comprehensive explanations with additional contextual information. This level of detail can help users understand complex concepts more deeply.

GitHub Copilot

GitHub Copilot produced concise responses and is primarily designed to assist developers during live coding sessions. Instead of long explanations, Copilot focuses on generating code suggestions quickly. This makes it highly useful for developers working in integrated development environments.

Overall, the comparative analysis shows that each AI tool has a different focus. ChatGPT emphasizes clarity and structured explanations, Gemini emphasizes detailed contextual understanding, and Copilot emphasizes real-time coding assistance.

Performance Evaluation

The performance of the AI systems was evaluated using several criteria.

Accuracy

Accuracy refers to how well the generated response matches the expected answer. Gemini achieved the highest score in the experiment due to its detailed explanation and strong contextual understanding.

Response Quality

ChatGPT demonstrated excellent response quality with well-structured explanations. The generated responses were easy to read and understand.

Response Speed

GitHub Copilot performed best in terms of response speed, as it is optimized for real-time coding assistance within development environments.

Practical Usability

Each AI tool offers different advantages depending on the user's needs. ChatGPT is suitable for learning and concept explanation, Gemini is useful for detailed analysis, and Copilot is ideal for real-time programming assistance.

The performance evaluation shows that AI-powered coding assistants significantly enhance developer productivity by providing instant coding support and explanations.

Limitations of the Study

The study is limited by the number of test queries used for evaluation. Additionally, the accuracy calculation is based on heuristic evaluation methods rather than full semantic understanding.

Future improvements can include larger datasets and more advanced evaluation metrics.

Recommendations

Future research should evaluate additional AI models and programming languages. Researchers can also analyze real programming tasks rather than conceptual questions.

Further development of the system can include machine learning based evaluation for more accurate comparison.

Conclusion

This research presented a comparative analysis of ChatGPT, Google Gemini, and GitHub Copilot using a web-based AI comparison system developed with HTML and Python. The system allows users to input questions and automatically evaluate responses from different AI models.

The results show that each AI system has distinct advantages in terms of response accuracy, explanation quality, and programming assistance. AI-powered coding assistants are becoming essential tools in modern software development.

Summary of Findings

- ChatGPT generates clear and structured explanations.
- Gemini provides highly detailed contextual responses.
- GitHub Copilot assists effectively during coding tasks.
- AI tools significantly improve programming productivity.

Contributions of the Study

This study contributes a practical AI comparison system that evaluates responses generated by different AI models. It provides insights into the effectiveness of AI coding assistants.

Practical Implications

Developers and students can use AI comparison systems to evaluate the quality of AI-generated responses and choose appropriate tools for programming tasks.

References

[1] M. K. Siam, M. M. Islam, and S. Ahmed, "Programming with AI: Evaluating ChatGPT, Gemini, AlphaCode, and GitHub Copilot for Programmers," *Proceedings of the International Conference on Computing Advancements*, ACM, 2024.

[2] A. Bayram, G. G. Menekse Dalveren, and M. Derawi, "Comparative Analysis of AI Models for Python Code Generation: A HumanEval Benchmark Study," *Applied Sciences*, vol. 15, no. 18, 2025.

[3] D. G. Paul, H. Zhu, and I. Bayley, "Benchmarks and Metrics for Evaluation of Code Generation Models," *arXiv preprint arXiv:2406.12655*, 2024.

[4] M. Hasan et al., "Assessing Small Language Models for Code Generation: An Empirical Study with Benchmarks," *Journal of Systems and Software*, 2026.

[5] B. Yetiştiren, I. Özsoy, M. Ayerdem, and E. Tüzün, "Evaluating the Code Quality of AI-Assisted Code Generation Tools," *arXiv preprint arXiv:2304.10778*, 2023.

[6] S. Zhang et al., "Examining Coding Performance Mismatch on HumanEval Benchmark," *Findings of the Association for Computational Linguistics (ACL)*, 2024.

[7] T. Miah, H. Zhu, and I. Bayley, "User-Centric Evaluation of ChatGPT for Code Generation," *Proceedings of the International Conference on Software Engineering*, 2024.

[8] R. Elgedawy et al., "Occasionally Secure: A Comparative Analysis of Code Generation Assistants," *arXiv preprint arXiv:2402.00689*, 2024.

[9] S. Ouyang, J. M. Zhang, M. Harman, and M. Wang, "An Empirical Study of the Non-Determinism of ChatGPT in Code Generation," *arXiv preprint arXiv:2308.02828*, 2023.

[10] S. Azbarka et al., "Evaluation of the Code Quality Generated by Generative AI," *Information Systems Student Research Journal*, vol. 2, no. 1, 2025.

[11] OpenAI, "GPT Models Documentation," OpenAI Research, 2024.

[12] Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," Google AI Research, 2023.

[13] GitHub, "GitHub Copilot Documentation," GitHub Inc., 2024.

[14] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. Pearson Education, 2022.

[15] "Language Model Benchmarks for Programming Tasks," *Wikipedia*, 2025.