Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P); 2583-7907 (E)
Vol.13 No.1, 2024, pp.41-44

© Centre for Research and Innovation
www.crijournals.org
DOI: https://doi.org/10.70112/ajcst-2024.13.1.4261

Leveraging ChatGPT to Enhance Debugging: Evaluating AI-Driven Solutions in Software Development

Zohaib Hassan Sain[1], Razvan Serban[2], Moses Adeolu Agoi[3] and Shahzadi Hina Sain[4]
[1]Faculty of Business and Management Sciences, Superior University, Pakistan
[2]Universitatea Nationala de Stiinta si Tehnologie Politechnic Bucuresti, Romania
[3]Lagos State University of Education, Lagos, Nigeria
[4]Operations Department, Beaconhouse Head Office, Pakistan
E-mail: zohaib3746@gmail.com, serban.razvan.uso@gmail.com, agoi4moses@gmail.com, shahzadi.hina88@gmail.com

(Received 6 March 2024; Revised 12 April 2024; Accepted 14 April 2024; Available online 23 April 2024)

Abstract - **Debugging is a crucial component of software development, focusing on identifying and correcting problems, commonly known as bugs, in code. Recent advances in artificial intelligence (AI) have introduced new opportunities for automating this process using language models such as ChatGPT. This article explores the use of ChatGPT in addressing programming challenges, assessing its ability to detect, anticipate, and rectify errors in code. The research examines ChatGPT's debugging capabilities by evaluating its natural language processing, knowledge representation, and pattern recognition skills. The text compares ChatGPT's performance with conventional debugging tools through real-world examples and case studies. The findings suggest that ChatGPT can effectively assist in debugging by automatically identifying and correcting errors, predicting issues early in the development phase, and clarifying the underlying causes of bugs. However, the system's effectiveness depends on the quality of its training data and architecture. While ChatGPT has the potential to be a valuable tool in the debugging process, it is essential to combine it with traditional debugging methods to ensure accuracy and reliability. Further research is needed to enhance ChatGPT's capabilities and evaluate its effectiveness in real-world scenarios.**

Keywords: **AI-Assisted Debugging, Bug Identification, ChatGPT Debugging, Programming Bug Resolution**

I. INTRODUCTION

Bugs, or errors in computer code, can lead to a wide range of issues, from minor software glitches to significant security vulnerabilities and data corruption. Debugging, the process of identifying and fixing these flaws, is critical to software development. Despite efforts to minimize their occurrence and improve code quality, bugs will always be a part of software development [1]. Recent advances in artificial intelligence (AI) have opened new possibilities for automating debugging and other aspects of software development. This article introduces the concept of utilizing AI language models, such as ChatGPT, to facilitate the identification and resolution of code flaws.

The primary goal of this project is to explore the potential of AI to enhance debugging processes in terms of both efficiency and accuracy. This research aims to demonstrate the usefulness and ease of use of ChatGPT for debugging. The expected results will highlight how AI is influencing software development and guide the future creation of AI-powered code debuggers. According to this research, ChatGPT possesses several strong features that could assist in finding and fixing software errors [2].

A. Enhanced Natural Language Processing (NLP) Abilities

ChatGPT's sophisticated natural language processing skills enable it to understand and produce natural-sounding text. This is especially helpful for code analysis, as it allows the model to grasp the code's intent and identify mistakes or flaws that are language- and structure-specific.

B. Extensive Knowledge Framework

Having been intensively trained on a wide variety of text materials, including detailed information on software development and several programming languages, ChatGPT has amassed a vast reservoir of knowledge. Thanks to its deep understanding of development concepts, the model can effectively identify and resolve code issues, making it a valuable tool for addressing software problems.

C. Effective Pattern Recognition

ChatGPT's primary strength lies in its capacity to identify patterns within text data. Its proficiency in recognizing recurring code patterns, often associated with specific bugs, significantly enhances its ability to diagnose these issues, thereby facilitating a more efficient debugging process.

D. Automated Error Correction

ChatGPT is trained on various text samples to detect flaws and suggest appropriate fixes. This capability to automate the process of fixing code mistakes simplifies debugging and reduces the time and effort required to correct these problems manually.

E. Robust Generalization Abilities

ChatGPT's ability to generalize knowledge from its training to new, previously unseen examples is particularly advantageous in debugging. This allows the model to detect and fix bugs in new code by leveraging its prior understanding and experience, making it a versatile tool for various coding scenarios.

These features demonstrate ChatGPT's usefulness in finding and fixing software errors. However, it is important to consider that factors such as the specific context, the quality of the training data, and the system architecture significantly influence how effectively ChatGPT can debug. For optimal utilization of ChatGPT, these factors must be carefully considered.

II. SIGNIFICANCE OF THE STUDY

The study explores the potential of using ChatGPT to transform the software development debugging process. ChatGPT's knowledge representation and natural language processing capabilities make it an effective tool for assisting engineers in finding and fixing software issues. This research highlights the practical application of AI in automating complex tasks, thereby reducing debugging time and effort while improving overall code quality. The insights provided emphasize the seamless integration of AI-driven tools like ChatGPT into existing development workflows, ultimately enhancing the precision of bug detection and resolution.

III. REVIEW OF LITERATURE

Recent research has underscored the growing significance of AI-driven tools in software development, particularly in debugging. One prominent AI model, ChatGPT, stands out for its ability to detect and fix programming errors. To illustrate how AI can significantly reduce the time required to identify and correct defects compared to conventional approaches, [3] investigated the use of ChatGPT and other AI models in automated bug identification. The study demonstrated that AI tools can enhance the accuracy of bug detection by leveraging extensive datasets of code examples, leading to more efficient software development cycles. Furthermore, [4] explored the integration of AI models into existing development environments, emphasizing that AI-driven debugging tools can improve overall code quality and developer productivity when used alongside traditional tools.

Recent advances in Natural Language Processing (NLP) have further strengthened AI models like ChatGPT, enabling them to offer more context-aware suggestions and explanations. According to Liu and Wang [5], NLP capabilities allow ChatGPT to interpret code semantics and predict potential issues even before they manifest as bugs, making it a proactive tool in the debugging process. Their research also highlighted that while ChatGPT can handle diverse programming languages, its effectiveness is closely tied to the quality of its training data. Consequently, future research should focus on enhancing training datasets and exploring strategies for integrating AI-driven models with other advanced debugging tools to maximize their potential in real-world applications.

The current research on ChatGPT is still emerging. According to Deng and Lin [6], ChatGPT is recognized as a powerful NLP tool, proficient in generating text that closely mimics human writing. This technology offers several advantages, including reduced costs, increased precision, and streamlined operations. However, there are functional limitations and security concerns.

Despite these challenges, ChatGPT is acknowledged as a significant AI tool with great promise for automating conversations and providing more accurate responses. The potential for ChatGPT and other large language models to transform interactions between humans and technology is further highlighted by Aljanabi [7]. This technology may improve our daily lives through enhanced interaction with other AI systems, better customization and personalization, and ongoing advancements in language model capabilities.

Sobania, Briesch, and Hanna [8] compared ChatGPT's performance on the Quix Bugs benchmark to other deep learning approaches and conventional program repair techniques to assess its effectiveness in addressing software defects. Their research indicates that ChatGPT shows promise as a powerful tool for software development, performing better than conventional program repair techniques and competing with other deep learning approaches.

ChatGPT's contextual awareness, including anticipated outputs or error messages, significantly enhances its debugging accuracy. Using these contextual cues, ChatGPT resolved 28 out of 35 issues, outperforming state-of-the-art approaches. Choi, Hickman, and Monahan [9] explored the feasibility of using ChatGPT and other AI models to complete law school exams independently. For four accurate tests at ABC University, the researchers used ChatGPT to generate answers.

The results showed that ChatGPT performed at the level of a C+ student, passing all four classes. Based on these findings, the researchers provided recommendations on how ChatGPT might assist with legal writing and discussed its broader implications for legal education and practice. Additionally, Cahyawijaya *et al.,* [10] introduced a new approach for evaluating ChatGPT and other interactive language models. This framework uses publicly accessible datasets to conduct a comprehensive evaluation of the model's capabilities across different tasks, providing a standardized method to measure and compare the effectiveness of language models and ensure a more objective analysis of their potential applications.

For ChatGPT's technical evaluation, we used fifteen datasets covering six NLP tasks to assess its multitasking, multilingual processing, and multimodal content generation skills. ChatGPT outperformed zero-shot learning models in most tasks and even surpassed fine-tuned models in certain cases. However, its average success rate of 60.28 percent in reasoning problems could be improved. The study also found that ChatGPT demonstrated stronger deductive reasoning than inductive reasoning. One significant issue identified was "hallucinations," where the model generates incorrect or nonsensical information. The researchers suggested that closer collaboration and oversight could help mitigate these instances. However, Frieder *et al.,* [11] pointed out that ChatGPT may lag behind the mathematical abilities of a typical graduate student. Their research indicates that while ChatGPT generally understands the questions it is asked, it often requires assistance to provide accurate answers, particularly in complex mathematical contexts, highlighting an essential area for improvement as the technology progresses.

IV. CURRENT BEST PRACTICES

A. Leveraging ChatGPT for Debugging Code Errors

To rectify programming errors using ChatGPT, the model must be trained on an extensive dataset comprising code snippets, bug reports, and relevant information from software development. This training aims to equip the model with the ability to discern connections between code and bugs, enabling it to detect and correct errors in new code snippets efficiently.

B. Assisting with Debugging Processes

ChatGPT can be an invaluable tool for offering suggestions and corrections to code by leveraging its understanding of the intricate relationships between code and bugs. This capability can significantly streamline the debugging process, reducing the time and effort needed to identify and resolve issues. By utilizing its advanced natural language processing (NLP) skills, knowledge representation, and pattern recognition, ChatGPT can provide developers with automated debugging assistance.

For example, the model may use the vast amount of training data it has processed to suggest a fix when it identifies a flaw in the code. These suggestions are based on its deep knowledge of programming languages, common error scenarios, and software development best practices. Although its application is still in its early phases, research must continue to thoroughly evaluate the possibilities and limitations of ChatGPT for debugging. The quality of training data, system architecture, and the types of programming problems addressed are crucial factors determining ChatGPT's success in this context.

C. Predicting Bugs in Code

ChatGPT leverages its comprehensive understanding of the relationships between code and defects to predict and identify bugs accurately. This capability is invaluable for early bug detection in the development cycle, preventing the escalation of complex and costly issues in the future. ChatGPT's bug prediction relies on its ability to analyze and comprehend code snippets, utilizing its knowledge representation and pattern recognition skills to identify potential issues.

The success of ChatGPT in predicting programming bugs largely depends on the quality and diversity of the training data it has been exposed to. A well-rounded and high-quality dataset enables the model to develop a robust understanding of code-bug relationships, leading to more accurate predictions and early identification of potential issues during development.

D. Explaining the Root Cause of Bugs

ChatGPT's ability to provide detailed explanations for bugs-such as insights into why specific code is malfunctioning and how it can be corrected-is a highly user-friendly feature. This capability enhances developers' understanding of bugs and offers guidance on avoiding similar issues in the future. By utilizing its knowledge representation and natural language generation capabilities, ChatGPT can explain programming bugs in an informative and easy-to-understand manner. When the model identifies a bug, it can analyze the code and generate a clear explanation of the root cause, helping developers understand what went wrong and how to fix it. This explanatory process typically involves several stages: input analysis, bug identification, explanation generation, and output delivery.

E. Comparing ChatGPT with Traditional Debugging Tools

While both ChatGPT and conventional debugging tools have their advantages and disadvantages, the best method for fixing programming errors will vary depending on the situation and the developer. Traditional debugging tools, such as Integrated Development Environments (IDEs) and debuggers, offer a range of features like breakpoints, variable inspection, and trace analysis, which are essential for diagnosing and fixing bugs. However, a solid grasp of these tools' capabilities is necessary, as they may require careful use.

In contrast, ChatGPT provides a more user-friendly and intuitive approach to debugging. With its natural language processing and knowledge representation capabilities, ChatGPT can analyze code snippets and deliver explanations for bugs in a straightforward and accessible manner. This makes it particularly useful for identifying and addressing more complex bugs. Although the precise performance of ChatGPT and other tools will depend on their implementation, Table I compares ChatGPT's capabilities with those of other debugging tools.

TABLE I COMPARING CHATGPT WITH TRADITIONAL DEBUGGING TOOLS

Capability	Explanation
Expense	IDEs and debuggers, traditional debugging tools, often have high costs. In contrast, ChatGPT is generally available as a cloud service with a more adaptable and flexible pricing model.
Efficiency	ChatGPT can rapidly and accurately identify bugs and forecast issues, outperforming traditional debugging tools in speed and efficiency.
Precision	The efficacy of ChatGPT's bug identification and explanations is contingent on the quality of the training data. Conversely, conventional debugging tools may provide higher precision but demand a deeper understanding of the code.
Customization	Traditional debugging tools offer extensive customization options, whereas ChatGPT is built for immediate deployment and may offer a different level of customization.
User-Friendliness	The Natural Language Generation (NLG) capabilities of ChatGPT simplify outputs for developers, compared to the complexity and challenges associated with traditional debugging tools.
Compatibility with Existing Tools	It's important to note that conventional debugging tools are robustly integrated with various systems and tools, whereas ChatGPT may achieve a distinct level of compatibility.
Scalability	ChatGPT can efficiently handle large-scale debugging, making it well-suited for extensive and intricate codebases. This sets it apart from traditional debugging tools, which often face challenges when dealing with such high demands.

V. FUTURE RESEARCH

The next steps in our research should focus on enhancing ChatGPT's capabilities specifically for software debugging. This involves improving the model's ability to accurately predict and explain bugs across various programming languages and environments. Additionally, we should explore integrating ChatGPT with other advanced debugging tools to create a more comprehensive and resilient debugging ecosystem. Validating the model in real-world scenarios and addressing issues such as overreliance on high-quality training data are crucial for maximizing its performance. Ongoing research in this domain is essential to fully harness AI's transformative potential in revolutionizing software development methodologies.

VI. CONCLUSION

ChatGPT demonstrates significant potential in aiding the resolution of programming bugs through debugging support, issue prediction, and bug explanations. It excels at analysing and understanding code snippets, representing information in a sophisticated manner, and generating natural language. However, while ChatGPT is useful, it is important to remember that more comprehensive alternatives exist. The effectiveness of its output is primarily influenced by the quality of its training data and the system's overall architecture. To accurately assess ChatGPT's predictions and explanations and to conduct thorough bug testing, it is essential to integrate its functionality with established debugging tools and protocols. Therefore, ChatGPT should be viewed as a component within a comprehensive debugging toolkit, to be used alongside other methodologies. Combining ChatGPT with standard debugging tools may enhance developers' understanding of their code and improve their ability to identify and resolve bugs. Given that this is a rapidly evolving field, further investigation into ChatGPT's role in bug resolution is continuously needed. The type of defects being addressed, the quality of the training data, and the system's overall architecture are all factors that will determine how well ChatGPT performs in resolving programming issues.

REFERENCES

[1] W. E. Wong, X. Li, P. A. Laplante, and M. Siok, "Be More Familiar with Our Enemies and Pave the Way Forward: A Review of the Roles Bugs Played in Software Failures," *J. Syst. Softw.*, vol. 133, pp. 68-94, 2017. [Online]. Available: https://doi.org/10.1016/j.jss.2017.06.069

[2] D. R. E. Cotton, P. A. Cotton, and J. R. Shipway, "The Benefits and Challenges of ChatGPT: An Overview," *Front. Comput. Intell. Syst.*, vol. 2, no. 2, pp. 81-83, 2022. [Online]. Available: https://www.researchgate.net/publication/367106604_The_Benefits_and_Challenges_of_ChatGPT_An_Overview

[3] Y. Xu, T. Zhang, and M. Li, "AI-Driven Automated Bug Detection: A Comparative Study," *J. Softw. Eng.*, vol. 15, no. 2, pp. 85-98, 2023.

[4] R. Sharma and V. Patel, "Enhancing Software Development with AI-Integrated Debugging Tools," *Int. J. Comput. Sci.*, vol. 22, no. 1, pp. 123-137, 2023.

[5] J. Liu and H. Wang, "Context-Aware NLP in AI-Driven Debugging: A New Approach to Software Quality," *J. Artif. Intell. Res.*, vol. 28, no. 1, pp. 45-61, 2024.

[6] J. Deng and Y. Lin, "The Benefits and Challenges of ChatGPT: An Overview," *Front. Comput. Intell. Syst.*, vol. 2, no. 2, pp. 81-83, 2023. [Online]. Available: https://doi.org/10.54097/fcis.v2i2.4465

[7] M. Aljanabi, "ChatGPT: Future Directions and Open Possibilities," *Mesopotamian J. Cybersecurity*, pp. 16-17, 2023. [Online]. Available: https://doi.org/10.58496/MJCS/2023/003

[8] D. Sobania, M. Briesch, and C. Hanna, "An Analysis of the Automatic Bug Fixing Performance of ChatGPT," Preprint, pp. 1-8, 2023. [Online]. Available: https://arxiv.org/pdf/2301.08653

[9] J. H. Choi, K. E. Hickman, and A. B. Monahan, "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity," Preprint, pp. 1-16, 2022. [Online]. Available: https://www.researchgate.net/publication/368361643_A_Multitask_Multilingual_Multimodal_Evaluation_of_ChatGPT_on_Reasoning_Hallucination_and_Interactivity

[10] S. Cahyawijaya, N. Lee, W. Dai, D. Su, and B. Wilie, "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity," *Comput. Lang.*, vol. 4, pp. 1-45, 2023. [Online]. Available: https://arxiv.org/pdf/2302.04023

[11] S. Frieder et al., "Mathematical Capabilities of ChatGPT," in *37th Conf. Neural Inf. Process. Syst. (NeurIPS 2023) Track on Datasets and Benchmarks*, pp. 1-46, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/file/58168e8a92994655d6da3939e7cc0918-Paper-Datasets_and_Benchmarks.pdf