International Journal of Information technology and Computer Engineering
ISSN: 2455-5290
Vol: 03, No. 01, Dec 2022 - Jan 2023
http://journal.hmjournals.com/index.php/IJITC
DOI: https://doi.org/10.55529/ijitc.31.17.22

Use Chat GPT to Solve Programming Bugs

Nigar M. Shafiq Surameery[1], Mohammed Y. Shakor[2*]

[1]Information Technology Department, College of Computer and Information Technology, University of Garmian, Kalar, Sulaimani, Kurdistan Region, Iraq
[2]Department of English, College of Education, University of Garmian, Kalar, Sulaimani, Kurdistan Region, Iraq*

Corresponding Email: [2]mohammed.yousif@garmian.edu.krd*

Received: 30 September 2022 **Accepted:** 25 December 2022 **Published:** 28 January 2023

Abstract: ***This research paper explores the use of Chat GPT in solving programming bugs. The paper examines the characteristics of Chat GPT and how they can be leveraged to provide debugging assistance, bug prediction, and bug explanation to help solve programming problems. The paper also explores the limitations of Chat GPT in solving programming bugs and the importance of using other debugging tools and techniques to validate its predictions and explanations. The paper concludes by highlighting the potential of Chat GPT as one part of a comprehensive debugging toolkit, and the benefits of combining its strengths with the strengths of other debugging tools to identify and fix bugs more effectively.***

Keywords: Chat GPT, Debugging Assistance, Bug Prediction, Bug Explanation, Programming Bugs.

1. INTRODUCTION

Bugs in computer code can cause significant problems, ranging from minor software crashes to security vulnerabilities and loss of data. Debugging code, the process of finding and fixing these bugs, is a critical aspect of software development and can be a time-consuming and complex task. Despite the efforts of software engineers to minimize bugs and improve code quality, bugs are an inevitable part of the software development process [1].

Recent advancements in artificial intelligence (AI) have opened up new opportunities for automating various tasks in software development, including debugging. This paper proposes the use of an AI language model, such as ChatGPT, to find and fix bugs in computer code. The goal of this research is to explore the potential of using AI to improve the efficiency and accuracy of the debugging process.

International Journal of Information technology and Computer Engineering
ISSN: 2455-5290
Vol: 03, No. 01, Dec 2022 - Jan 2023
http://journal.hmjournals.com/index.php/IJITC
DOI: https://doi.org/10.55529/ijitc.31.17.22

The contribution of this research is to demonstrate the feasibility and effectiveness of using ChatGPT to find and fix bugs in computer code. The results of this study will provide insight into the capabilities of AI in software development and will inform the development of future AI-based tools for debugging code. The characteristics of ChatGPT that can be useful in finding and fixing bugs in computer code are as follows [2][3]:

- Natural Language Processing (NLP) Capabilities: ChatGPT has advanced NLP capabilities, including the ability to understand and generate human-like text. This can be useful in analyzing code, as it allows the model to understand the intent behind the code and to identify potential bugs based on the language used.
- Knowledge Representation: ChatGPT has been trained on a vast amount of text data, including information about software development and programming languages. This gives it a rich representation of the knowledge and concepts related to software development, which can be leveraged to find and fix bugs in code.
- Pattern Recognition: ChatGPT has the ability to recognize patterns in text data, which can be useful in identifying bugs in code. For example, it can identify patterns in code that are commonly associated with specific types of bugs.
- Error Correction: ChatGPT has been trained on large amounts of text data and can use this training to suggest corrections to code, including fixing bugs. This can help to automate the debugging process and reduce the time and effort required to find and fix bugs.
- Generalization: ChatGPT has the ability to generalize from its training data to new, unseen examples. This can be useful in debugging code, as it allows the model to identify bugs in new code based on its prior knowledge.

These are some of the key characteristics of ChatGPT that can be useful in finding and fixing bugs in computer code. However, it's important to note that the effectiveness of using ChatGPT for debugging will depend on the specific task, the quality of the training data, and the design of the system.

Related Works

There is still little research on ChatGPT due to its novelty. The authors in [4] conclude that the ChatGPT is a strong Natural Language Processing tool that can produce conversations that appear to be written by a human. Utilizing this technology has advantages such as enhanced efficiency, higher precision, and reduced costs. Nonetheless, it faces some hurdles such as security issues and limitations in capabilities. Despite these difficulties, ChatGPT is a noteworthy Artificial Intelligence tool that can be utilized for automating discussions and generating more precise replies. The researchers in [5] also showed that future of ChatGPT and other large language models is filled with thrilling prospects and the possibility of having a significant effect on how we interact with technology. The integration with other AI systems, the opportunity for further customization and personalization, and the ongoing improvement of language model performance are some of the many exhilarating chances for this technology to make a positive impact on our lives. The authors in [6] evaluate the effectiveness of using

International Journal of Information technology and Computer Engineering
ISSN: 2455-5290
Vol: 03, No. 01, Dec 2022 - Jan 2023
http://journal.hmjournals.com/index.php/IJITC
DOI: https://doi.org/10.55529/ijitc.31.17.22

ChatGPT for solving software bugs by comparing its performance on the QuixBugs benchmark set with the results of other deep learning and standard program repair techniques. The authors found that ChatGPT's performance was competitive with other deep learning methods and notably better than standard program repair techniques. Furthermore, ChatGPT's advantage lies in its ability to accept additional information, such as expected outputs or error messages, which can increase its success rate. By providing these hints, ChatGPT was able to fix 31 out of 40 bugs, outperforming state-of-the-art results. [7] Aimed to determine the ability of AI models, specifically ChatGPT, in writing law school exams without human assistance. The researchers used ChatGPT to generate answers for four real exams at the University of Minnesota Law School and graded the exams as part of their regular grading processes. The results showed that ChatGPT performed on average at the level of a C+ student, passing all four courses. The researchers then discussed the implications of these results for legal education and the practice of law, and provided advice and example prompts on how ChatGPT can aid in legal writing. A novel ramework for evaluating interactive language models like ChatGPT using publicly available datasets is proposed in [8] . The researchers conducted a comprehensive technical evaluation of ChatGPT using 21 datasets across 8 different NLP tasks. The evaluation considered ChatGPT's capabilities for multitasking, handling multiple languages, and generating multimodal content. The results showed that ChatGPT outperforms zero-shot learning models in most tasks and even beats fine-tuned models in some cases. However, the accuracy of ChatGPT in reasoning categories was found to be unreliable, with an average accuracy of 64.33%. The researchers also found that ChatGPT performs better in deductive reasoning compared to inductive reasoning. Additionally, ChatGPT was found to have hallucination problems, which can be improved through human collaboration with the model. On the contrary, [9] is indicated that mathematical skills of ChatGPT fall short of what a typical mathematics graduate student can achieve. Findings indicate that while ChatGPT comprehends the question, it frequently fails to give accurate answers.

State of the Art: ChatGPT to Solve Programming Bugs

The use of ChatGPT for solving programming bugs involves training the model on a large dataset of code snippets, bug reports, and related information about software development. The goal of this training is to enable the model to understand the relationships between code and bugs, and to use this understanding to identify and fix bugs in new code snippets.

Debugging Assistance

ChatGPT can be used to provide suggestions and corrections to code based on its understanding of the relationships between code and bugs. This can help to automate the debugging process and reduce the time and effort required to find and fix bugs. ChatGPT can be used to provide debugging assistance for programming codes by using its natural language processing (NLP) capabilities, knowledge representation, and pattern recognition abilities.

Once the model has been trained, it can be used to provide suggestions and corrections to code based on its understanding of the relationships between code and bugs. For example, if a bug is present in the code, ChatGPT can suggest a correction based on its training data. The suggestions can be based on its prior knowledge of programming languages, common bug patterns, and best practices in software development.

International Journal of Information technology and Computer Engineering
ISSN: 2455-5290
Vol: 03, No. 01, Dec 2022 - Jan 2023
http://journal.hmjournals.com/index.php/IJITC
DOI: https://doi.org/10.55529/ijitc.31.17.22

The use of ChatGPT for debugging assistance is still an emerging area of research, and further work is needed to fully evaluate its potential and limitations. The effectiveness of using ChatGPT for debugging will depend on the quality of the training data, the design of the system, and the specific programming bugs being targeted.

Bug Prediction

ChatGPT can be used to predict the likelihood of bugs in new code based on its understanding of the relationships between code and bugs. This can be useful for identifying bugs early in the development process, before they become more difficult and costly to fix.

ChatGPT can be used for programming bug prediction by leveraging its ability to analyze and understand code snippets. The model can use its knowledge representation and pattern recognition capabilities to identify potential bugs in new code based on its training data.

The effectiveness of ChatGPT for programming bug prediction will depend on the quality of the training data and the design of the system. If the training data includes a comprehensive set of code snippets and bugs, the model can develop a strong understanding of the relationships between code and bugs. This can lead to accurate predictions and help to identify potential bugs early in the development process.

Bug Explanation

Chat GPT can be used to provide explanations for bugs, including the reasons why a particular piece of code is causing a bug and how it can be fixed. This can be useful for improving the understanding of bugs and for providing insights into how to prevent similar bugs in the future.

Chat GPT can provide explanations for programming bugs by using its knowledge representation and natural language generation capabilities. When Chat GPT identifies a bug in code, it can use its understanding of the code and the relationships between the code and the bug to generate an explanation. This explanation can help developers understand the root cause of the bug and how to fix it. The process takes place through four stages include input, analysis, explanation and output.

Comparing Cha tGPT with Other Debugging Tools

ChatGPT and traditional debugging tools each have their own strengths and weaknesses, and that the best approach for solving a particular programming bug will depend on the specific circumstances of the bug and the developer's experience and preferences. Debugging tools such as integrated development environments (IDEs) and debuggers can provide a wide range of features and capabilities to help developers identify and fix bugs, including breakpoints, variable inspection, and trace analysis. However, these tools can also be complex to use and may require significant expertise to get the most out of them. On the other hand, ChatGPT can provide a more accessible and intuitive way to solve programming bugs. Its natural language processing and knowledge representation capabilities allow it to analyze code snippets and provide explanations for bugs in a way that's easy for developers to understand. This can make it particularly useful for identifying and fixing more complex bugs. Table 1 shows capabilities of ChatGPT compared with Other Debugging Tools.

International Journal of Information technology and Computer Engineering
ISSN: 2455-5290
Vol: 03, No. 01, Dec 2022 - Jan 2023
http://journal.hmjournals.com/index.php/IJITC
DOI: https://doi.org/10.55529/ijitc.31.17.22

Capability	Description
Cost	Traditional debugging tools and techniques, such as IDEs and debuggers, can be expensive, whereas ChatGPT is often provided as a cloud-based service with a more flexible pricing model.
Speed	ChatGPT can provide quick and efficient bug explanations and predictions, whereas traditional debugging tools can take longer to run and provide results.
Accuracy	The accuracy of ChatGPT's bug predictions and explanations can be impacted by the quality of its training data, whereas traditional debugging tools and techniques may offer a higher degree of accuracy, but require a deeper understanding of the code.
Customizability	Traditional debugging tools can be highly customizable, but ChatGPT is designed to work out-of-the-box and may not offer the same level of customization.
Ease of use	ChatGPT's natural language generation capabilities make it easy for developers to understand its results, whereas traditional debugging tools can be more complex and difficult to use.
Integration with existing tools	Traditional debugging tools can integrate with other tools and systems, whereas ChatGPT may not offer the same level of integration.
Scalability	ChatGPT can be used to debug code at scale, whereas traditional debugging tools may struggle to keep up with large and complex codebases.

The actual results for each capability will depend on the specific implementation of ChatGPT and the other debugging tools being compared.

2. CONCLUSION

In conclusion, ChatGPT can play a role in solving programming bugs by providing debugging assistance, bug prediction, and bug explanation. Its ability to analyze and understand code snippets, along with its knowledge representation and natural language generation capabilities, make it well-suited for these tasks. However, it's important to note that while ChatGPT can be a useful tool for solving programming bugs, it's not a perfect solution. The quality of its outputs will depend on the quality of the training data and the design of the system. Additionally, it's important to use other debugging tools and techniques to validate its predictions and explanations, and to ensure the code is free of bugs. Therefore, ChatGPT should be seen as one part of a comprehensive debugging toolkit, and used in conjunction with other tools and techniques to ensure the best possible results. By combining the strengths of ChatGPT with the strengths of other debugging tools, developers can gain a more complete understanding of their code, and identify and fix bugs more effectively.

The use of ChatGPT to solve programming bugs is a promising area of research, and further work is needed to fully evaluate its potential and limitations. The effectiveness of using

International Journal of Information technology and Computer Engineering
ISSN: 2455-5290
Vol: 03, No. 01, Dec 2022 - Jan 2023
http://journal.hmjournals.com/index.php/IJITC
DOI: https://doi.org/10.55529/ijitc.31.17.22

ChatGPT for solving bugs will depend on the quality of the training data, the design of the system, and the specific programming bugs being targeted.

3. REFERENCES

1. W. E. Wong, X. Li, P. A. Laplante, and M. Siok, "Review of the Roles Bugs Played in Software Failures PT US CR", J. Syst. Softw., 2017, doi: 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", Vol. 2, No. 2, 2022.
3. K. Elkins and J. Chun, "Can GPT-3 Pass a Writer's Turing Test?," Journal of Cultural Analytics, vol. 5, no. 2, Sep. 2020, doi: 10.22148/001c.17212.
4. J. Deng and Y. Lin, "The Benefits and Challenges of ChatGPT : An Overview", vol. 2, no. 2, pp. 81–83, 2022.
5. M. Aljanabi, "ChatGPT : Future Directions and Open possibilities", vol. 2023, pp. 16–17, 2023.
6. D. Sobania, M. Briesch, and C. Hanna, "An Analysis of the Automatic Bug Fixing Performance of ChatGPT", 2023.
7. J. H. Choi, K. E. Hickman, and A. B. Monahan, "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity", pp. 1–16, 2022.
8. S. Cahyawijaya, N. Lee, W. Dai, D. Su, and B. Wilie, "A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity", 2023, doi: 10.1016/j.jss.2017.06.069.
9. S. Frieder et al., "Mathematical Capabilities of ChatGPT", pp. 1–29, 2023.