Southeast Europe Journal of Soft Computing
Available online: http://scjournal.ius.edu.ba

ISSN 2233 – 1859

Vol 12, No. 1, March 2023, pp. 08-12

Methods and Applications of ChatGPT in Software Development: A Literature Review

[1*]Muna Abu Jaber, [1]Adna Beganović and [1,2]Ali Abd Almisreb

[1]Faculty of Engineering and Natural Sciences, International University of Sarajevo,
Hrasnicka Cesta 15, Ilidža 71210 Sarajevo, Bosnia and Herzegovina
International IT University, Department of Cybersecurity, Almaty, Kazakhstan
*Corresponding Author: munaabujaber@outlook.com

Article Info

Article history:

Article received on 08 05 2023

Received in revised form 12 05 2023

Keywords:

ChatGPT, OpenAI, AI model, Software Development

ABSTRACT: This research paper explores the use of large language models in software development that has gained significant attention in recent years. One of the most popular models is ChatGPT, a large language model trained by OpenAI. ChatGPT has been applied in some of the software development tasks, like programming Numerical methods, code generation, solving programming bugs, code completion and even in practicing for computer science exam. This paper presents a systematic review of the literature on the application of ChatGPT in software development. The review covers 12 studies that use ChatGPT in some fields of software development. We evaluate their results and the offering for the field. Overall, the findings of this review offer significant contributions to the current understanding and future direction of using ChatGPT in the domain of software development.

1. INTRODUCTION

Software development is the process of creating, testing, designing, and maintaining software. It involves a huge range of activities, also we have the side that include analyzing user requirements, developing software architectures, writing code, debugging. Considering all these aspects, having the assistance of large language model such as ChatGPT can be remarkably facilitative. Finding a suitable number of research papers was somewhat challenging due to the highly novel nature of this research topic, particularly in the field of software development. The aims of the research paper are to evaluate the performance of ChatGPT, a deep learning approach, for automated program repair, bug fixing, application in education, code quality, programming in parts such as Numerical Methods. In this paper we present the methods and applications of ChatGPT, as well as the results of its performance in software development. Our paper presents methodology and results involved in research articles that are presented in this paper.

2. CHATGPT FOR SOFTWARE DEBUGGING AND TESTING

For automated program repair and bug fixing, the authors used several methods to evaluate ChatGPT's performance compared to other deep learning approaches. Firstly, the authors examined ChatGPT's success rate in bug fixing, using a range of bug scenarios, and found that it was competitive with other deep learning approaches. They also explored the usefulness of ChatGPT's dialog system, which allows users to provide additional information, and found that it can increase the system’s success rate in bug fixing [1]. Secondly, the authors analyzed the features of ChatGPT and how they can aid in debugging tasks such as bug prediction and explanation [2]. They acknowledged the limitation of the program in programing bug solving and emphasized the need for other debugging methods to validate its predictions and explanations. Also, authors concluded that ChatGPT has potential as a component of comprehensive debugging toolkit and suggested that integrating its strengths with those of other debugging tools can help identify and resolve bugs more efficiently.

The paper "ChatGPT and Software Testing Education: Promises and Perils" [3] investigates the potential effects of ChatGPT on software testing education. The history of predictive language modeling for code is examined, as is the recent progress of massive language models built on neural transformer topologies, which hold immense potential for a wide range of applications. ChatGPT is introduced as a conversational agent trained to respond to end-user requests and questions. In the paper, a conducted study is mentioned in which the model was asked a series of questions on a well-known software testing curriculum in order to assess ChatGPT's performance in a software testing scenario. ChatGPT offered right or partially correct replies in around 44% of cases and correct or partially correct explanations in approximately 57% of cases. Surprisingly, the discovery of prompting the model in a shared question context raised the rate of right answers marginally. ChatGPT's potential was explored in software testing education, such as its capacity to deliver on-demand support and individualized feedback, as well as its ability to increase student engagement and encourage collaborative learning. However, many concerns were raised as well about the usage of ChatGPT, including the risk of over-reliance on the model, the model's propensity to perpetuate misconceptions or biases, and the ethical implications of employing AI in education. Overall, the study provides an intriguing viewpoint on the possible impact of AI and conversational bots like ChatGPT on education, particularly in technical disciplines like software testing. The emphasis is on the importance of additional research and reflection on the benefits and drawbacks of employing these technologies in educational contexts.

3. CHATGPT FOR SOFTWARE OPTIMIZATION AND ALGORITHMS PROGRAMMING

Some of the papers were written to demonstrate the potential of ChatGPT as a tool for optimizing the Gcode generation process in 3D printing [4], and to propose a new way of approaching AGI using LLMs (Large Language Models) as controllers for AI tasks across different domains and modalities [5]. The proposed solution of those papers can advance the field of AI in significant ways. The first paper explores the use of ChatGPT to optimize the Gcode generation process in 3D printing. The authors explain that ChatGPT can be trained on existing Gcode data to generate optimized Gcode and analyze various printing parameters, resulting in noteworthy reductions in both time and material usage. The authors conducted performance tests using the ability to revolutionize the 3D printing industry through enhancement in effectiveness and precision. In the second paper, the authors discuss the challenges of achieving artificial general intelligence (AGI) and propose a solution that involves LLMs such as ChatGPT as a controller to manage existing AI models to solve complex AI tasks. The authors present HuggingGPT, which integrates various AI models from HuggingFace into a unified system using ChatGPT for task planning, model selection, and result summarization. This powerful combination allows HuggingGPT to handle complex AI tasks across multiple domains and modalities, leading to impressive results and paving the way for AGI. The authors also highlighted several concerns related to the use of ChatGPT in education [6], including the risk of over-reliance on the model, the potential for the model to reinforce misconceptions or biases, and the ethical implications of using AI in education. The paper offers an interesting perspective on the potential impact of AI and conversational agents in ChatGPT on education, particularly in the technical field. The authors emphasized the need for future research and reflection on the pros and perils of using those tools in educational settings.

The capability of ChatGPT for programming numerical algorithms is explored for the first time in this research paper [7]. The study examines the ability of ChatGPT to generate codes for numerical algorithms in different programming languages, to debug and improve written codes by users, to complete missed parts of numerical codes, to rewrite available codes in other programming languages, and to parallelize serial codes. Additionally, the study of those authors assesses whether ChatGPT can distinguish code written by humans from those written by machines. Mathematical problems such as the Poisson equation, the incompressible Navier-Stokes equations, and storing sparse matrices are considered. Scientific machine learning techniques such as physics-informed neural networks and convolutional neural networks with applications to computational physics are also explored. Through various examples, successes, failures, and challenges of ChatGPT are investigated. The study concludes that ChatGPT can successfully program numerical algorithms in different programming languages, but certain limitations and challenges exist that require further improvement of this machine learning model.

4. CHATGPT AND CREATIVITY OF SOFTWARE DEVELOPMENT

When we are talking about Navigation Complexity in software engineering, the paper [8] discusses the importance of creativity in software engineering and the limitations of current chatbot tools for programmers, which often only present a single solution. In this research, a prototype tool called “GPTCOMPARE was introduced, which allows programmers to visually compare multiple source code solutions generated by GPT-n models for the same programming-related query. The tool highlights the similarities and differences between the solutions, making it easier for programmers to evaluate and select the best option for their specific requirement and constraints. This work-in-progress prototype has potential to improve the decision-making process for software engineers.

The article "Towards human-robot collaboration in software architecture with ChatGPT" [9] outlines a novel method of software architecture that makes use of ChatGPT. In this paper, claims have been made that by generating ideas, offering solutions, and offering feedback on design choices, ChatGPT can be utilized as a collaborative tool to support human software architects in the design process. The importance of software architecture and the challenges faced during the design process are covered at the outset of the discussion. It is also outlined that software architecture requires expertise in a range of areas, including software engineering, systems design, and human-computer interaction, and involves complicated decision-making. By offering on-demand knowledge and feedback during the design process, ChatGPT could aid human software designers. In this case study, it shows how ChatGPT helped a group of software architects create a new virtual assistant system. Using ChatGPT, teams may generate ideas, discuss design options, and obtain feedback on their decisions. According to the authors, the collaboration was fruitful because ChatGPT provided excellent ideas and assistance throughout the design process. The end of the study discusses the possible benefits of adopting ChatGPT as a collaboration tool for software architecture. ChatGPT can help advance expertise in software architecture by providing on-demand support to designers with varying levels of experience. It is mentioned that ChatGPT may promote innovation and creativity by generating innovative ideas and disproving preconceived views. Demonstrations have been shown on how ChatGPT may help human software architects by generating ideas, proposing solutions, and providing feedback on design decisions. Such strategies can democratize knowledge in software architecture, foster creativity, and innovation, and increase software design quality. More research in this area could result in significant advancements in software development and beyond.

5. CHATGPT AND PROMPT ENGINEERING

“A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT” [10] is a study in which a catalog of fast patterns was used as a way improve the performance of ChatGPT. Rapid development is an important phase in using ChatGPT because it entails creating prompts to elicit the necessary responses from the model. The authors suggest that by presenting excellent cue design examples, the model catalog can be valuable in attracting cue designers. The importance of advice development and the challenges of good advice development are discussed in this paper. The process of creating a pattern catalog is described in detail, including how it can entail reviewing many cue-response pairings and locating patterns that efficiently provide the requisite responses. Several examples of prompt models were provided, such as question-answer pairs, fill-in-the-blank prompts, and prompts that use multiple-choice questions. In the study, it is shown how to use schema catalogs during advice development. A case study was described in which a catalog was used to develop messages for a response task involving natural language processing. Then, the performance of ChatGPT was compared with and without hints to schema folders and found that folders increased model performance by about 4%. The researchers believe that this catalog can help simplify the rapid construction process, reduce the time and effort required to develop effective prompts, and improve the performance of ChatGPT. They also point out that the catalog can be updated and expanded over time to include new models and improve the overall efficiency of the just-in-time construction process. While the research focuses on tasks related to natural language processing, the method can be applied to other tasks and domains. The study findings highlight the potential benefits of incorporating rapid model catalogs into the rapid construction process.

The paper “ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design” [11] proposes a set of prompt patterns that can be used to improve various aspects of software engineering using ChatGPT. ChatGPT is argued that it can be used as a tool to support software engineers in tasks such as code quality improvement, refactoring, requirements elicitation, and software design. The discussion begins with the importance of each of these tasks in software engineering and the challenges associated with them. Software engineers are facing significant difficulties in each of these areas: understanding complex codebases, identifying, and resolving code, eliciting requirements from stakeholders, and designing systems that meet the needs of users. The approach in this paper involves using ChatGPT to generate prompts that can assist software engineers in each of these tasks. A set of prompt patterns are presented, which can be used to generate questions, suggestions, and feedback that can help to improve code quality, guide refactoring efforts, elicit better requirements, and support software design decisions. For example, the prompt patterns include questions such as "What are the potential consequences of this design decision?" and suggestions such as "Consider using the Strategy pattern to decouple this code from specific implementations." A case study is also described in which the effectiveness of improving code quality approach is evaluated. A group of software engineers were asked to use ChatGPT to identify and resolve code issues in a set of Java code samples. It was reported that the engineers used the prompt patterns to identify many code issues and that the prompts were helpful in guiding their analysis and decision-making. The article is concluded with a discussion on the potential benefits of using ChatGPT to support software engineering tasks. It is discussed that ChatGPT can help to reduce the cognitive load on software engineers by providing on-demand assistance in complex tasks and that it can help to democratize expertise in software engineering by providing guidance and support to engineers with varying levels of experience. Overall, the paper offers a compelling approach to using ChatGPT to support software engineering tasks. Further research in this area could lead to significant advances in software engineering.

6. PROMPTING QUESTION TO CHATGPT

One of the ways to learn about the functionalities performed by ChatGPT and its capabilities is to ask ChatGPT questions about itself. One of the questions for ChatGPT that summarizes the general idea of its performance is the following:

“ChatGPT, as an AI model, where do you see yourself in the future of software development and testing?”

The answer ChatGPT gave is the following:

As an AI model, I see myself playing an increasingly important role in software development and testing in the future. With my ability to understand and generate natural language, I can help developers and testers communicate more efficiently and effectively, reducing the cognitive load on them and democratizing expertise. I can also be used to automate repetitive tasks, such as code review and documentation, freeing up developers and testers to focus on more creative and value-added work. Furthermore, I can be trained on vast amounts of data to improve my performance and adapt to new software development and testing paradigms as they emerge. Overall, I believe that I have the potential to revolutionize the software development and testing process and make it more accessible, efficient, and effective [12].

7. CONCLUSION

The papers discussed the potential of ChatGPT, a large language model based on the GPT-3.5 architecture, in various areas such as automated program repair and bug fixing, optimization of Gcode generation in 3D printing, achieving artificial general intelligence, programing numerical algorithms, and software engineering decision-making. The papers also highlighted concerns related to the use of ChatGPT, including the risk of over-reliance on the model, the potential for the model to reinforce misconceptions or biases, and the ethical implications of using AI in education and other fields. Overall, the research suggests that ChatGPT has the potential to be powerful tool in various domains, but also requires further research and development to address limitations and concerns.

REFERENCES

[1] D. Sobania, M. Briesch, C. Hanna, and J. Petke "An Analysis of the Automatic Bug Fixing Performance of ChatGPT", 2023..

[2] N. M. S. Surameery, M. Y. Shakor "Use Chat GPT to Solve Programming Bugs" 2023 .

[3] S. Jalil, S. Rafi, T. D. LaToza, K. Moran, W. Lam, "ChatGPT and Software Testing Education: Promises & Perils", 2023

[4] S. Badini, S. Regondi, E. Frontoni, and R. Pugliese "Assessing the capabilities of ChatGPT to improve additive manufacturing troubleshooting", 2023.

[5] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, Y. Zhuang "HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in HuggingFace", 2023.

[6] S. Bordt, and U. von Luxburgen "ChatGPT Participates in a Computer Science Exam", 2023

[7] A. Kashefi, and T. Mukejri "ChatGPT for Programming Numerical Methods",2023

[8] C. Treude, "Navigating Complexity in Software Engineering: A Prototype for Comparing GPT-n Solution", 2023.

[9] A. Ahmad, M. Waseem, P. Liang, M. Fehmideh, M. S. Aktar, and T. Mikkonen "Towards Human-Bot Collaborative Software Architecting with ChatGPT", 2023.

[10] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT", 2023.

[11] J. White, S. Hays, Q. Fu, J. Spencer-Smith, D. C. Schmidt "ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design", 2023

[12] https://chat.openai.com/