Mesopotamian journal of Computer Science
Vol. **(2023)**, 2023, **pp** 9-15
DOI: https://doi.org/10.58496/MJCSC/2023/002 ; ISSN: 2958-6631
https://mesopotamian.press/journals/index.php/cs

Research Article

Role of ChatGPT in Computer Programming.

Som Biswas [1,*]

[1] *Department of Radiology, UTHSC College of Medicine Memphis: The University of Tennessee Health Science Center College of Medicine, Memphis, TN, 38103, USA*

ARTICLE INFO

Article History

Received 06 Dec 2022
Accepted 14 Jan 2023
Published 05 Feb 2023

Keywords

ChatGpt

Computer programming

AI

ABSTRACT

Purpose: The purpose of this abstract is to outline the role and capabilities of ChatGPT, a language model developed by OpenAI for computer programming. **Methodology**: ChatGPT is a large language model that has been trained on a diverse range of texts and can perform a variety of programming-related tasks. These tasks include code completion and correction, code snippet prediction and suggestion, automatic syntax error fixing, code optimization and refactoring suggestions, missing code generation, document generation, chatbot development, text-to-code generation, and answering technical queries. **Results**: ChatGPT can provide users with explanations, examples, and guidance to help them understand complex concepts and technologies, find relevant resources, and diagnose and resolve technical problems. Its use can improve overall satisfaction with support services and help organizations build a reputation for expertise and reliability. **Conclusions**: ChatGPT is a powerful and versatile tool for computer programming that can support developers and users in a wide range of tasks.

1. INTRODUCTION

Computer programming is a complex field that requires a deep understanding of programming languages, algorithms, and data structures. It can be challenging for developers to keep up with the latest technologies and best practices, as well as diagnose and resolve technical problems. This is where ChatGPT, a language model developed by OpenAI, comes in. ChatGPT offers a wide range of capabilities for computer programming, including code completion, correction, prediction, error fixing, optimization, document generation, chatbot development, text-to-code generation, and technical query answering. With its ability to provide explanations, examples, and guidance, ChatGPT is a valuable resource for technical support and its programming-related capabilities can improve efficiency and accuracy for developers and organizations. This paper will outline the role and capabilities of ChatGPT in computer programming [1][2].

2. STATE OF THE ART

ChatGPT can be used as a tool in computer programming for various tasks such as:

1. Code Completion and Correction
2. Document generation
3. Chatbot development
4. Text-to-Code Generation
5. Answering technical queries.

It can be integrated into programming environments to improve developer productivity and speed up the coding process. However, ChatGPT is not a full-fledged programming language and still requires human intervention for more complex tasks.

2.1 Role of ChatGPT in Code Completion and Correction

1. In Code Completion and Correction, ChatGPT can be used as a tool to assist developers by:
2. Predicting and suggesting code snippets based on the input provided
3. Automatically fixing syntax errors and common mistakes
4. Providing suggestions for code optimization and refactoring
5. Generating missing code based on the context of the project

[*] *Corresponding author. Email: ssbinmemphis@gmail.com*

By using ChatGPT, developers can save time and improve the accuracy of their code, allowing them to focus on more important tasks.

2.2 Role of ChatGPT in Predicting and suggesting code snippets

In Predicting and suggesting code snippets, ChatGPT uses its language generation capabilities to:

1. Analyze the context of the code and suggest relevant code snippets to the user
2. Learn the coding style and preferences of the user to make better suggestions
3. Use its vast knowledge base of programming languages and libraries to provide accurate suggestions
4. Consider the project's dependencies and constraints to make context-aware suggestions

By utilizing ChatGPT in this manner, developers can speed up the coding process, avoid common mistakes, and increase code quality.

2.3 Role of ChatGPT in Automatically fixing syntax errors and common mistakes

In Automatically fixing syntax errors and common mistakes, ChatGPT can be used to:

1. Detect syntax errors in real-time and suggest fixes
2. Offer suggestions to resolve common programming mistakes such as missing semicolons or brackets
3. Provide warnings for code that may cause future issues
4. Verify code against best practices and suggest improvements

By utilizing ChatGPT in this manner, developers can improve the accuracy and quality of their code, reducing the time spent on manual error correction and debugging.

2.4 Role of ChatGPT in Predicting and suggesting code snippets

Here's an example of ChatGPT predicting and suggesting code snippets:
Suppose a developer is writing a Python script and wants to add a list of numbers. They type the following code:

```
python
numbers = [1, 2, 3]
sum = 0
for num in numbers:
sum += num
```

ChatGPT, recognizing the context of the code, can suggest the following code snippet to the developer:

```
python
print(f"The sum of numbers is {sum}")
```

The developer can then simply copy and paste this snippet into their code to print the result of the sum. In this manner, ChatGPT can help developers write code more efficiently and quickly.

2.5 Role of ChatGPT in Automatically fixing syntax errors and common mistakes with examples

Here's an example of ChatGPT automatically fixing syntax errors and common mistakes:

Suppose a developer is writing a JavaScript function and has made a syntax error:

```
javascript
function addTwoNumbers(a, b) {
return a + b
}
```

ChatGPT can detect the missing semicolon at the end of the return statement and suggest the following correction:

```
javascript
function addTwoNumbers(a, b) {
  return a + b;
}
```

By making this correction, the developer can avoid potential syntax errors that may cause problems in their code. In this manner, ChatGPT can help developers write error-free code more efficiently.

2.6 Role of ChatGPT in Providing suggestions for code optimization and refactoring In Providing suggestions for code optimization and refactoring

ChatGPT can be used to:

1. Analyze code for performance bottlenecks and suggest optimization techniques
2. Identify code duplication and suggest ways to refactor the code for improved readability and maintainability
3. Recommend design patterns and best practices for structuring code
4. Provide suggestions for reducing code complexity

By utilizing ChatGPT in this manner, developers can improve the performance and maintainability of their code, making it easier to scale and update in the future.

2.7 Role of ChatGPT in Providing suggestions for code optimization and refactoring with examples

Here's an example of ChatGPT providing suggestions for code optimization and refactoring:

Suppose a developer has written a function to search for an item in a list of dictionaries:

```
python
def search_item(item_name, item_list):
  for item in item_list:
    if item['name'] == item_name:
      return item
  return None
```

ChatGPT can analyze the code and suggest the following refactoring to improve performance:

```
python
def search_item(item_name, item_list):
  item_dict = {item['name']: item for item in item_list}
  return item_dict.get(item_name)
```

By refactoring the code in this manner, the function will be much faster for large lists since it uses a dictionary to store the items. In this manner, ChatGPT can help developers optimize their code for better performance.

2.8 Role of ChatGPT in Generating missing code based on the context of the project

In Generating missing code based on the context of the project, ChatGPT can be used to:

1. Analyze the existing code and generate missing code snippets to complete the project
2. Consider the dependencies, constraints, and desired outcome of the project to generate relevant code
3. Use its vast knowledge base of programming languages and libraries to generate accurate and efficient code
4. Learn from the coding style and preferences of the user to make better suggestions

By utilizing ChatGPT in this manner, developers can save time and effort on repetitive tasks, allowing them to focus on more complex and creative aspects of the project. It can also assist in training new developers and provide a starting point for new projects. Generating missing code based on the context of the project with examples. Here's an example of ChatGPT generating missing code based on the context of the project:

Suppose a developer is building a website using Django and wants to create a view to display a list of articles. They have written the following code:

```
python
# views.py
from django.shortcuts import render
from .models import Article

def article_list(request):
  articles = Article.objects.all()
  # missing code here
```

ChatGPT can analyze the code and the context of the project and suggest the following missing code to complete the view:

```
python
# views.py
from django.shortcuts import render
from .models import Article
def article_list(request):
articles = Article.objects.all()
return render(request, 'article_list.html',
{'articles': articles})
```

By generating the missing code in this manner, the developer can complete the view more efficiently and quickly.

2.9 Role of ChatGPT in Document generation for programming

In Document generation for programming, ChatGPT can be used to:

1. Automatically generate documentation for code, including descriptions of functions, classes, and variables.
2. Provide explanations and examples of how to use code, making it easier for other developers to understand.

3. Generate API reference documentation from code.
4. Create technical documents, such as project reports, with ease.

By utilizing ChatGPT in this manner, developers can save time and effort on documenting their code and projects, allowing them to focus on other important tasks. It can also help to ensure that documentation is up-to-date, accurate, and comprehensive.

Here's an example of ChatGPT generating documentation for programming:

Suppose a developer has written the following code for a simple calculator:

```
python
class Calculator:
  def add(self, a, b):
    """
    Add two numbers
    """
    return a + b

  def subtract(self, a, b):
    """
    Subtract two numbers
    """
```

```
return a - b
def multiply(self, a, b):
"""
Multiply two numbers
"""
return a * b
def divide(self, a, b):
"""
Divide two numbers
"""
return a / b
```

ChatGPT can generate the following API reference documentation for the code:

```
python
class Calculator:
  def add(self, a, b):
    """
    Add two numbers.
:param a: The first number.
    :param b: The second number.
    :return: The sum of the two numbers.
    :rtype: float
    """
    def subtract(self, a, b):
    """
    Subtract two numbers.
```

```
def multiply(self, a, b):
    """
    Multiply two numbers.
    :param a: The first number.
    :param b: The second number.
    :return: The product of the two
numbers.
    :rtype: float
    """
    def divide(self, a, b):
    """
    Divide two numbers.
   :param a: The first number.
    :param b: The second number.
    :return: The quotient of the two
numbers.
    :rtype: float
    """
```

By generating the API reference documentation in this manner, other developers will have a clear understanding of how to use the code, making it easier for them to work with the calculator class.

2.10 Role of ChatGPT in Chatbot development

In Chatbot development, ChatGPT can be used to:

1. Generate natural language responses for a chatbot, allowing it to have meaningful conversations with users.
2. Train a language model for the chatbot, fine-tuning it to better understand the context of conversations and generate more accurate responses.
3. Generate code snippets for common chatbot functionality, such as handling user inputs or generating responses based on specific keywords.
4. Suggest improvements to existing chatbot code, helping developers to optimize and refine their chatbot's functionality.

By utilizing ChatGPT in this manner, developers can speed up the process of developing a chatbot, as well as improve its overall performance and user experience. It can also help to ensure that the chatbot's language and behavior are consistent, making it easier for users to interact with the chatbot.

Here's an example of ChatGPT in Chatbot development:

Suppose a developer is building a chatbot to assist customers with ordering food. They want the chatbot to handle questions like "What are the options for a vegetarian meal?" and "Can I get a salad with chicken?"

ChatGPT can help in the following ways:

1. Generating natural language responses: ChatGPT can generate the following responses for the chatbot:

- "We have several vegetarian options, including a veggie burger, a grilled vegetable wrap, and a Mediterranean salad. Which one would you like to order?"
- "Yes, you can add chicken to the salad. Would you like to make any other changes to your order?"

2. Training a language model: ChatGPT can be fine-tuned on examples of restaurant menu items and customer orders to better understand the context of conversations and generate more accurate responses.
3. Generating code snippets: ChatGPT can generate code snippets to handle user inputs, such as recognizing when a user wants to order a vegetarian meal or add chicken to a salad.
4. Suggesting improvements: ChatGPT can suggest improvements to the chatbot's code, such as adding more options for vegetarian meals or handling requests for gluten-free options.

By using ChatGPT in these ways, the developer can save time and effort in developing the chatbot and improve its overall performance and user experience.

2.11 Role of ChatGPTin Text-to-Code Generation

In Text-to-Code Generation, ChatGPT can be used to:

1. Convert natural language descriptions of desired software functionality into code snippets or complete code solutions.
2. Generate code in a specific programming language, such as Python or Java, based on the input text.
3. Suggest improvements to existing code based on the input text, helping developers to optimize and refine their code.

By utilizing ChatGPT in this manner, developers can reduce the time and effort required to manually write code, allowing them to focus on more complex tasks. It can also help to ensure that code is more consistent, reducing the number of bugs and errors that need to be fixed later on. Additionally, Text-to-Code Generation can make programming more accessible to non-developers by allowing them to describe desired functionality using natural language. Here's an example of ChatGPT in Text-to-Code Generation:

Suppose a developer is tasked with creating a function that calculates the average of a list of numbers. They describe the desired functionality to ChatGPT in natural language as follows:

"I need a function in Python that takes a list of numbers as input and returns the average of those numbers."

ChatGPT can then generate the following code in response:

```
python
def average(numbers):
    total = sum(numbers)
    count = len(numbers)
    return total / count
```

In this example, ChatGPT has taken the natural language description of the desired functionality and generated the corresponding code. The developer can then use this code as-is or make modifications as needed. Another example is a non-developer who wants to perform text analysis on a set of documents. They describe the desired functionality to ChatGPT in natural language as follows:

"I want to find the most common words in a set of text documents and create a bar chart to visualize the results." ChatGPT can then generate the following code in response (in Python):

```
python
import nltk
from nltk.probability import FreqDist
import matplotlib.pyplot as plt

def analyze_text(documents):
    words = []
    for document in documents:
        words+= nltk.word_tokenize(document)
    fdist = FreqDist(words)
    fdist.plot(30, cumulative=False)
    plt.show()
```

In this example, ChatGPT has taken the natural language description of the desired functionality and generated the corresponding code in Python to perform text analysis and visualize the results.

2.12 Role of ChatGPT in answering technical queries

1. ChatGPT can play a role in answering technical queries by:
2. Providing accurate and relevant answers to questions related to programming, software development, and other technical topics.
3. Offering explanations and examples to help users better understand complex concepts and technologies.
4. Guiding users to relevant resources, such as documentation or tutorials, to help them find the information they need.
5. Assisting users in diagnosing and resolving technical problems, such as error messages or software bugs.

By using ChatGPT in this way, organizations can reduce the time and effort required to answer technical queries and help users more quickly find the information they need. This can improve overall satisfaction with support services and help organizations build a reputation for expertise and reliability.

3. CONCLUSION

ChatGPT is a cutting-edge language model developed by OpenAI that offers a wide range of capabilities for computer programming. It can assist in code completion, correction, prediction, error fixing, optimization, document generation, chatbot development, text-to-code generation, and technical query answering [3][4]. The ability of ChatGPT to provide explanations, examples, and guidance makes it a valuable resource for technical support, while its ability to perform programming-related tasks can improve efficiency and accuracy[5]. Overall, ChatGPT is a powerful tool for the programming community that has the potential to make a significant impact on the field.

Conflict of interest

No competing relationships or interests that could be perceived as influencing the research are reported in the paper.

Funding

No financial grants or awards related to the research are disclosed in the paper, signifying a lack of external funding.

Acknowledgment

The author acknowledges that this article was partially generated by ChatGPT (powered by OpenAI's language model, GPT-3.5; http://openai.com). The editing was performed by the author.

References

[1] Phillips, Tanner, et al. "Exploring the use of GPT-3 as a tool for evaluating text-based collaborative discourse." *Companion Proceedings of the 12th* (2022): 54.
[2] Lund, Brady, and Wang Ting. "Chatting about ChatGPT: How May AI and GPT Impact Academia and Libraries?." *Lund, BD, & Wang* (2023).
[3] Winston, Patrick Henry. *Artificial intelligence*. Addison-Wesley Longman Publishing Co., Inc., 1984.
[4] Charniak, Eugene, et al. *Artificial intelligence programming*. Psychology Press, 2014.
[5] Tanimoto, Steven L. *The elements of artificial intelligence: an introduction using LISP*. Computer Science Press, Inc., 1987.