

Learn programming languages using ChatGPT

Name Fahmi Rizky Nugraha

Universitas Utpadaka Swastika, Tangerang, Indonesia, 15112
E-mail: fahmi.rizky@utpas.ac.id,

Accepted: April 25, 2024 | Published: April 30, 2024

ABSTRACT

Technology is currently developing rapidly, where every year there are lots of programming languages which appears to provide solutions in creating applications or as business support. Many of these programming languages require one to learn to be able to do them competing in a world of work that is now dominated by AI and machine learning so we must be able to take advantage of learning quickly using ChatGPT. However, someone who is truly an expert in this field is still needed to verify the correctness of the code produced.

KEYWORDS: ChatGPT, OpenAI, Artificial Intelligence, machine learning

1. Introduction

ChatGPT is a free programming language that everyone can use and register to use the application however, as time goes by, ChatGPT also provides solutions or additional tools for the enterprise version, usually this is needed by companies that want to get more features, so companies that need to learn new programming skills need customized guidance. With the emergence of advanced Natural Language Generation models such as the ChatGPT API, it is now possible to create convenient and personalized tutoring systems with AI for computer science education. This paper presents GPTutor, a ChatGPT-powered programming tool, which is a Visual Studio Code extension using the ChatGPT API to provide programming code annotation. By integrating the Visual Studio Code API, GPTutor can comprehensively analyze the provided code by referring to the relevant source code. As a result, GPTutor can use hints designed to explain the selected code with pop-up messages. GPTutor is now published on the Visual Studio Code Extension Marketplace, and its source code is openly accessible on GitHub. Initial evaluation shows that GPTutor provides the most concise and accurate explanations compared to vanilla ChatGPT and GitHub Copilot.

2. General Instructions

In using ChatGPT we must still have skills or abilities where we also have to understand the algorithm of programming that we will write using ChatGPT, as we know ChatGPT is just an application that learns how to understand what we want and what ChatGPT provides. not necessarily all of them are true.

Therefore, the role of a professional is to use ChatGPT to proofread every line of code generated by ChatGPT.

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

1. Code Completion and Correction
2. Document creation
3. Chatbot development
4. Text to Code Generation
5. Answering technical questions.

Can be integrated into programming environments to increase developer productivity and speed up processes coding. However, ChatGPT is not a complete programming language and still requires human intervention for more complex tasks.

You
please create login using golang

ChatGPT
Creating a simple login system in Go involves setting up a web server, handling HTTP requests, managing user sessions, and integrating with a data store for user authentication. Below is a basic example of how you might implement a login system using Go with a focus on simplicity and clarity.

First, ensure you have Go installed on your system. You can check this by running `go version` in your terminal.

Next, create a new directory for your project and navigate into it:

```
bash
mkdir go-login-example
cd go-login-example
```

Inside this directory, create a new Go module:

```
bash
go mod init login
```

Now, create a `main.go` file for your Go code:

```
go
// main.go
package main
import (
    "fmt"
    "net/http"
    "html/template"
)
// user represents a basic user structure
type user struct {
    Username string
    Password string
}
// Simulated user database
var users = []user{
    { "user1", "password1" },
    { "user2", "password2" },
}
var sessions = make(map[string]string)
func main() {
    http.HandleFunc("/", home)
    http.HandleFunc("/login", login)
```

Figure 1: Sample Creat Login use golang

ChatGPT there gives what we command as we can see in the login, there are only a few standard validations in the login process.

Therefore, we need a professional to check or add validation that is safe for the application when used later

example, adding validation before logging in, we will add an OTP to ensure that it is the user who is logging in

You
also add logic to sent otp before login

ChatGPT
To enhance the login system by adding OTP (One-Time Password) functionality before allowing users to log in, we can integrate a mechanism to generate and validate OTPs via email or SMS. In this example, I'll demonstrate how to generate and verify a simple OTP using email.

We'll use the `net/smtp` package in Go to send emails. For this example, you'll need an SMTP server to send emails. You can use a third-party service like Gmail or set up your SMTP server.

Let's modify the previous example to include OTP generation and validation:

```
// Generate and send OTP via email
otp := generateOTP()
otps[username] = otp
sendOTPByEmail(foundUser.Email, otp)
// Redirect to OTP verification page
http.Redirect(w, r, "/verify?username="+username, http.StatusFound)
}
func verifyOTP(w http.ResponseWriter, r *http.Request) {
    username := r.FormValue("username")
    otp := r.FormValue("otp")
    if storedOTP, ok := otps[username]; ok && storedOTP == otp {
        sessionID := generateSessionID()
        sessions[sessionID] = username
        http.SetCookie(w, &http.Cookie{Name: "session", Value: sessionID})
        delete(otps, username) // Clear OTP after successful login
        http.Redirect(w, r, "/dashboard", http.StatusFound)
        return
    }
    http.Redirect(w, r, "/", http.StatusUnauthorized)
```

Figure 2: Add logic used ChatGPT

2.1 Research and method

In ChatGPT, apart from being able to create code, you can also look for problems in a code.

Usually programmers search from websites like StackOverflow or a blog to find solutions to these documents. In fact, there are many ways, such as looking at the documentation on the language website.

However, to make it easier and faster, it is now better to use ChatGPT because it is answered directly by the system, no need to use comments from people or read documentation.

You
how solved query rollback in golang

ChatGPT
In Go, handling query rollbacks typically involves working with database transactions. Transactions allow you to execute a sequence of database operations that either all succeed or are all rolled back if any one of them fails. Here's how you can handle query rollbacks in Go using database transactions:

- 1. Establish a Database Connection:**
First, you need to establish a connection to your database. Use the appropriate driver for your database (e.g., 'github.com/go-sql-driver/mysql' for MySQL, 'github.com/lib/pq' for PostgreSQL).
- 2. Begin a Transaction:**
Start a transaction using the 'Begin' method provided by your database connection. This marks the beginning of a transaction block.

```
go
tx, err := db.Begin()
if err != nil {
    // Handle error
}
defer tx.Rollback() // Ensure rollback is called if no explicit commit
```

- 3. Execute Queries Within the Transaction:**
Perform your database operations (queries, inserts, updates, deletes) within the transaction block using the transaction ('tx') object.

```
go
_, err := tx.Exec("INSERT INTO users (name, email) VALUES (?, ?)", "John Doe", "john.doe@example.com")
if err != nil {
    // Rollback the transaction if any operation fails
    tx.Rollback()
    // Handle error
}
// Handle error
```

- 4. Commit or Rollback the Transaction:**
After executing all necessary queries, decide whether to commit or rollback the transaction based on the success or failure of your operations.

- Commit:** If all operations were successful, commit the transaction to make the changes permanent.

```
go
err := tx.Commit()
if err != nil {
    // Handle error
}
```

- Rollback:** If any operation fails, explicitly call 'Rollback' to discard all changes made within the transaction.

2.2 Result

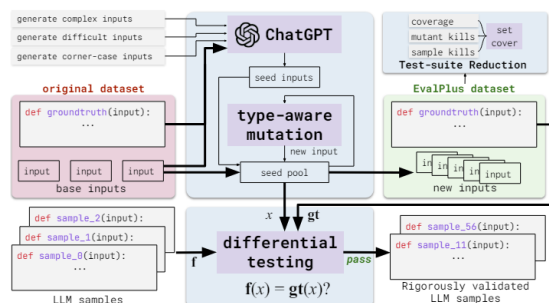


Figure 3: Overview of EvalPlus

To find out the truth produced by ChatGPT, many people carry out tests with several tools like the one in the picture

	#Tests				#Tasks
	Avg.	Medium	Min.	Max.	
HUMANEVAL	9.6	7.0	1	105 ²	
HUMANEVAL ⁺	764.1	982.5	12	1,100	164
HUMANEVAL ⁺ -MINI	16.1	13.0	5	110	

Figure 4: Overview of EvalPlus

The pros between using ChatGPT and Google	
ChatGPT Pros	Google Pros
ChatGPT uses a very effective and reliable way of implementing natural language processing, understanding the required task.	Google allows users to use voice search, which can be more effective and faster than typing the required task.
The developer can input the code as a prompt and ask ChatGPT to change a portion of it.	Moreover, Google is giving results in the most-suitable form — text, images, videos, QnA, products, etc., providing a more desirable solution for users.
ChatGPT offers flexibility for programmers to choose the language of the choice. For instance, developers can ask to write code in Python, C/C++/C#, JavaScript etc.	As an internet search engine, Google is offering multiple options to choose from for the search results, giving the user the possibility to choose the most fitting answer.

The cons between using ChatGPT and Google	
ChatGPT Cons	Google Cons
The code generated is based on specific patterns and structures present in training data, meaning it usually will not be flexible enough to meet specific requirements of the project.	As with ChatGPT, the code found on internet may be just a general example, so the programmer will spend more time on modifying it to fit the requirements of the project.
Usually, the user must re-generate the response to get the code he needs, as the first attempt might fail due to the servers being overloaded (especially during peak hours). Thus, writing a basic code with ChatGPT might be more time-wasting than writing it yourself.	The majority of the code took from Google either has some bugs or doesn't fully suit the user's needs, which will require the programmer to spend time adjusting the code to his own requirements.
Security of the code is weak, as the generated code is not rigorously tested for security vulnerabilities, as it may give the hackers a full access to private functions which receives confidential data from users, such their names and private data, and send users data from company's database.	The security of the code may be weak, because it is found in open-source and may be easily used to break the product by hackers by leaving them a backdoor to your source code, private functions and confidential information (passwords or entire databases).

So ChatGPT really helps programmers in making their work easier and in terms of accuracy it is quite close but only globally

6. Conclusion

It became clear that ChatGPT would not replace programmer. On the other hand, the use cases will tend to make things easier for a programmer questions/bugs, in this case, it could be related to the impact of the industrial revolution in the 18th century. It changes the way people work by automating part of their workload, drastically increasing productivity and wages.

Rather than replacing programmers, what is more likely to happen is ChatGPT and other AI language models will impact the programming industry by helping the work of developers, potentially improving their work

The workflow and reduces stress when debugging. As for shortcomings like this, this may change standardize the way future programmers think, which may result in a

decline in critical thinking skills, abilities are important for determining how a problem should be approached. Instead of relying ChatGPT to solve our problems, we should try to learn to understand the root of the problem ourselves, so that we can avoid it in the future.

References

- [1] Biswas, S. (2023). Role of ChatGPT in Computer Programming.: ChatGPT in Computer Programming. *Mesopotamian Journal of Computer Science*, 2023, 8-16.
- [2] Surameery, N. M. S., & Shakor, M. Y. (2023). Use chat gpt to solve programming bugs. *International Journal of Information technology and Computer Engineering*, (31), 17-22.
- [3] Chen, E., Huang, R., Chen, H. S., Tseng, Y. H., & Li, L. Y. (2023, June). GPTutor: a ChatGPT-powered programming tool for code explanation. In *International Conference on Artificial Intelligence in Education* (pp. 321-327). Cham: Springer Nature Switzerland.
- [4] LUPAN, L., MIHALACHI, M., & BERCO, A. (2023). Will ChatGPT replace programmers?. *MASTERANZILOR ȘI DOCTORANZILOR Chișinău*, 5-7 aprilie 2023.
- [5] Liu, J., Xia, C. S., Wang, Y., & Zhang, L. (2024). Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36.