CODE, FEEDBACK, AND QUESTION GENERATION ON PROGRAMMING TOPICS USING CHATGPT API

Jawad Alshboul[1], Erika Baksa-Varga[2]
[1]PhD Student, [2]Associate Professor
[1,2]Institute of Informatics, Faculty of Mechanical Engineering and Informatics
[1]alshboul.jawad@student.uni-miskolc.hu, [2]erika.b.varga@uni-miskolc.hu

ABSTRACT

This paper provides interested readers with an introduction to automatic generation of code, question, and feedback for topics related to programming using OpenAI API and ChatGPT API. Chat Generative Pre-trained Transformer is a member of the Generative Pre-trained Transformer family of large language models that use Artificial Intelligence to create human-like text answers in natural language discussions. Automatic code generation helps during learning and brainstorming and speeds up the delivery of the solution. Automatic question generation helps create questions on programming topics to test student comprehension. Automatic feedback offers explanation and guidance for better practices in programming. The main purposes of this paper are to provide a short overview of using ChatGPT and OpenAI for teaching programming and to show how easy and beneficial it is to use such brand-new tools.

1. INTRODUCTION

Automatic Question Generation (AQG) is the method of automatically generating questions from various sorts of data utilizing computational approaches and algorithms [1]. This field lies at the intersection of natural language processing (NLP), artificial intelligence (AI), and machine learning (ML) aspects to produce questions that test and asses comprehension or aid learning. Automatic question generation aims to simplify the process of developing questions for educational purposes, exams, training modules, and other applications [2]. Computers can create questions from textual passages, photos, audio, or other types of data by leveraging algorithms and language patterns [3].

Automatic Code Generation (ACG) creates program codes automatically based on some specification or source code. Rather than spending much time writing code from scratch, developers and programming learners use software tools to generate the code based on specific instructions [4-5].

Automatic Feedback Generation (AFG) creates feedback automatically based on inputs, actions, or criteria without direct human intervention. This process is commonly practiced in several fields, including education and software development. In educational settings, AFG can provide students with timely feedback on their assignments and exercises, and it is commonly used in e-learning platforms. For the software engineering context, AFG can provide evaluations and insights for learners and software developers regarding quality, performance, and correcting bugs [6-7].

Large language models (LLMs) represent sophisticated AI systems that interpret and generate human-like written content. They undergo training on expansive text collections through deep learning techniques to grasp the intricacies of language, context, and various patterns. These models facilitate several tasks, such as language translation, summarization, and answering questions. Despite comprising

millions or even billions of parameters, they can produce coherent and contextually fitting responses based on the information they analyze [8].

Chat Generative Pre-trained Transformer (ChatGPT) is a dialogue-based Artificial Intelligence (AI) model designed by OpenAI to create human-like text answers in natural language discussions. It is a member of the GPT (Generative Pre-trained Transformer) family of large language models [8].

2. OPENAI API AND CHATGPT API

The application programming interface (API) is a method for computer applications to communicate with each other. It is a form of software interface that provides a set of services to other programs. OpenAI API is a sophisticated platform that gives access to cutting-edge AI models and technology from OpenAI. The API allows users to access some of OpenAI's most advanced models, such as GPT (Generative Pre-trained Transformer). The OpenAI API operates using diverse models, each with distinct capabilities and pricing tiers. GPT-4 and Turbo GPT-4 are models that improve GPT-3.5 by understanding and generating natural language or code. GPT-3.5 and GPT-3.5 Turbo are models that improve GPT-3 by understanding and generating natural language or code. GPT-3 is a set of models that can understand and generate natural language, and OpenAI currently considers this set a legacy. Among the GPT-3.5 models, the most proficient and cost-effective is the gpt-3.5-turbo variant [9].

Understanding the context is crucial for ChatGPT API to deliver accurate and pertinent responses. By guiding the AI through user queries and system prompts, it becomes adept at addressing user intentions, requirements, and constraints. Ensuring contextualization in interactions helps the AI comprehend the user's objectives, situation, and any specific constraints or preferences. Offering adequate context can lead to more precise, pertinent, and beneficial AI responses. There are two ways to provide context: user queries and system messages. User queries offer context by formulating clear and direct questions from users that include relevant details, constraints, and requirements. System messages offer context by defining the AI's role, setting the initial context, or offering specialized support within a particular domain.

It is crucial to thoroughly review the generated code snippet for accuracy, completeness, and alignment with best practices. Testing this code in your development environment is essential to ensure it works as intended and to catch any possible errors. If the code does not fully meet your requirements, adjust it as needed or provide additional context and constraints in your query.

3. AUTOMATIC QUESTION GENERATION USING CHATGPT API

The ChatGPT API is helpful for automatically generating questions from code snippets. Figure 1 shows the code snippet used as the context to ask questions. Figure 2 shows the system message defining AI's role as a Python coding teacher, and it also shows a user query requesting generating questions about the code snippet shown in Figure 1. Figure 3 shows the text generated by ChatGPT API in response to the query shown in Figure 2. It suggests several reasonable questions on the code snippet.

```python
def Fibonacci(n):
  # Check if input is 0 then it will
  # print incorrect input
  if n < 0:
    print("Incorrect input")

  # Check if n is 0
  # then it will return 0
  elif n == 0:
    return 0

  # Check if n is 1,2
  # it will return 1
  elif n == 1 or n == 2:
    return 1

  else:
    return Fibonacci(n-1) + Fibonacci(n-2)
```

Fig. 1.
Fibonacci.py file

```python
import os
import openai
import secret

openai.api_key=secret.api_key

def chatgpt_code_review(file_name):
  try:
    file_path = file_name
    with open(file_path,'r') as file:
      code = file.read()

    response=openai.ChatCompletion.create(model="gpt-3.5-turbo",
    messages=[{"role": "system", "content": "You are a programming teacher with expertise in Python. pls write
    questions on the following code"},{"role": "user", "content": code}])

    #return response.choices[0].message.content
    answer = response['choices'][0]['message']['content'].strip()
    return answer

  except Exception as e:
    return(f"An error occurred: {e}")

print(chatgpt_code_review("fibonacci.py"))
```

Fig. 2.
A user query asking for generating questions on the code shown in Figure 1

```
1. What does the function `Fibonacci` do?
2. What is the purpose of the `if n < 0` condition?
3. What values are returned when `n` is equal to 0,
1, or 2?
4. What happens when `n` is not equal to 0, 1, or 2?
5. Which mathematical sequence is this function
calculating?
6. How does the function calculate the Fibonacci
sequence?
```

Fig. 3.
ChatGPT API response to the query shown in Figure 2

4. AUTOMATIC CODE GENERATION USING CHATGPT API

The ChatGPT API is a valuable tool for producing code snippets in several programming languages, helping users to optimize their development workflow. Figure 4 shows a user query specifying the programming language (Python), the desired function (reverse), and both the input and output requirements. The system message establishes the context that defines AI's role as a proficient Python coding assistant.

Figure 5 shows the text generated by ChatGPT API in response to the query shown in Figure 4. It suggests a working solution with an explanation.

```
import os
import openai
import secret

openai.api_key=secret.api_key

response = openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[{"role": "system", "content": "You are a coding assistant with expertise in Python."},
        {"role": "user", "content":"Generate a Python code snippet. The Python function you need should take in a string as
        an input and return the same string but reversed" }])

print(response['choices'][0]['message']['content'].strip())
```

Fig. 4.
A user query asking for generating a specific function to reverse input

```
Sure! Here's a Python code snippet that defines a
function to reverse a string:

```python
def reverse_string(string):
    return string[::-1]

# Example usage
input_string = input("Enter a string: ")
reversed_string = reverse_string(input_string)
print("Reversed string: ", reversed_string)
```

In this code, the `reverse_string` function takes a
string as an input, using slicing with a step of -1
to reverse the string, and then returns the reversed
string. It uses negative indexing to iterate from the
end of the string towards the beginning.

The example usage asks the user to input a string,
calls the `reverse_string` function to reverse it,
and then prints the reversed string.
```

Fig. 5.
ChatGPT API response to the query shown in Figure 4

5. AUTOMATIC FEEDBACK GENERATION USING CHATGPT API

Utilizing the ChatGPT API for debugging can assist users in pinpointing and fixing coding errors, leading to more robust and more effective code. Enhancing your development process and code quality can be achieved by employing effective methods to seek debugging support and feedback with the provided solutions. Likewise, when writing code, seeking assistance, or conveying information, it is crucial to offer precise details and context and to leverage system messages to ensure clarity and effectiveness. The displayed code excerpt in Figure 6 contains a user query that specifically highlights the issue (IndexError) and includes the relevant code portion. The system message establishes the AI's role as a Python-savvy coding assistant, framing the context accordingly. Figure 7 shows the text generated by ChatGPT API in response to the query shown in Figure 6. It explains the issue and suggests a fix for this problem.

```
import os
import openai
import secret
openai.api_key=secret.api_key

message=[{"role": "system", "content": "You are a coding assistant with expertise in Python."},
{"role": "user", "content": "I have this Python code that throws an IndexError: `def get_element
(lst, index): return lst[index]`. Can you help me understand the issue and suggest a fix?"}]

response=openai.ChatCompletion.create(model="gpt-3.5-turbo",messages= message)

print(response['choices'][0]['message']['content'].strip())
```

Fig. 6.
A user query that specifically highlights the issue (IndexError)

```
The IndexError occurs when the given index is out of
range for the provided list. This means that the
index value is either less than 0 or greater than the
length of the list minus one.

To fix this issue, you can check if the given index
is valid before accessing the element in the list.
Here's an example of how you can do that:

```python
def get_element(lst, index):
    if index < 0 or index >= len(lst):
        return None  # or raise an exception if
desired
    return lst[index]
```

In this case, the function will return `None` if the
index is invalid. You can replace `None` with any
value that suits your needs, or you can choose to
raise a custom exception if you prefer.
```

Fig. 7.
ChatGPT API response to the query shown in Figure 6

6. CONCLUSION

Using the ChatGPT API to optimize coding practices allows learners and developers to hone their coding skills and improve the quality of their code. Using efficient ways of requesting advice and applying suggested tactics will help you improve your programming skills, resulting in more effective, maintainable, and robust code. The ChatGPT API is a valuable tool for debugging codes, supporting users in finding and correcting common coding difficulties, and recommending improvements to existing code. Contextual awareness is critical in providing meaningful and precise responses in ChatGPT API interactions. Users may assist the AI in developing responses that appropriately meet their goals, requirements, and constraints by providing context through user queries and system messages. Finally, the paper introduces the users to the benefits of ChatGPT API by giving three examples of automatic question generation from program codes, automatic code generation, and automatic feedback generation.

REFERENCES

[1] J. ALSHBOUL and E. BAKSA-VARGA: **A Review of Automatic Question Generation in Teaching Programming**, International Journal of Advanced Computer Science and Applications (IJACSA), vol. 13, no. 10, pp. 45–51, 2022, https://doi.org/10.14569/IJACSA.2022.0131006.

[2] G. KURDI, J. LEO, B. PARSIA, U. SATTLER, and S. AL-EMARI: **A Systematic Review of Automatic Question Generation for Educational Purposes**, *Int. J. Artif. Intell. Educ.*, vol. 30, no. 1, pp. 121–204, Mar. 2020, https://doi.org/10.1007/s40593-019-00186-y.

[3] N. MULLA and P. GHARPURE, **Automatic Question Generation: A Review of Methodologies, Datasets, Evaluation Metrics, and Applications**, *Prog. Artif. Intell.*, vol. 12, no. 1, pp. 1–32, Jan. 2023, https://doi.org/10.1007/s13748-023-00295-9.

[4] J. ALSHBOUL and E. BAKSA-VARGA: **A Hybrid Approach for Automatic Question Generation from Program Codes**, International Journal of Advanced Computer Science and Applications (IJACSA), vol. 15, no. 1, pp. 10–17, 2024, http://dx.doi.org/10.14569/IJACSA.2024.0150102.

[5] J. SHIN and J. NAM: **A Survey of Automatic Code Generation from Natural Language**, *J. Inf. Process. Syst.*, vol. 17, no. 3, pp. 537–555, Jun. 2021, https://doi.org/10.3745/JIPS.04.0216.

[6] F. H. WANG: **Efficient Generation of Text Feedback in Object-Oriented Programming Education Using Cached Performer Revision**, *Mach. Learn. Appl.*, vol. 13, p. 100481, Sep. 2023, https://doi.org/10.1016/j.mlwa.2023.100481.

[7] S. SARSA, P. DENNY, A. HELLAS, and J. LEINONEN: **Automatic Generation of Programming Exercises and Code Explanations Using Large Language Models**, presented at the International Computing Education Research, Lugano, Switzerland: ACM, Aug. 2022, pp. 27–43. https://doi.org/10.1145/3501385.3543957.

[8] P. P. RAY: **ChatGPT: A Comprehensive Review on Background, Applications, Key Challenges, Bias, Ethics, Limitations and Future Scope**, Internet Things Cyber-Phys. Syst., vol. 3, pp. 121–154, 2023, https://doi.org/10.1016/j.iotcps.2023.04.003.

[9] **OpenAI API Models,** OpenAI API Documentation. Accessed: Dec. 15, 2023. [Online]. Available: https://platform.openai.com/docs/models