OpenAI's GPT-3/ChatGPT: Suggests & Corrects

Pritish Bhave[1], Rushikesh Chopade[2], Aditya Stanam[3], & Shrikant Pawar[4*]

[1]Department of Computer Science, Western Kentucky University, 1906 College Heights Blvd, Bowling Green, KY 42101, USA

[2]Department of Geology and Geophysics, Indian Institute of Technology, Kharagpur, West Bengal 721302, India

[3]University of Iowa, Iowa City, IA, 52242-5000, USA

[4]Department of Computer Science & Biology, Claflin University, Orangeburg, SC 29115, USA

Introduction

- ChatGPT (Chat Generative Pre-Trained Transformer) is a chatbot developed by OpenAI built on OpenAI's GPT-3 family of large language models which has been fine-tuned using supervised and reinforcement learning techniques.
- Both approaches used human trainers to improve the model's performance.
- With supervised learning, the model was trained on data from users and the artificial intelligence assistant, while in reinforcement learning, human trainers first ranked responses that the model had created in previous conversations which was further fine-tuned on using several iterations of Proximal Policy Optimization (PPO). The training data includes information about internet phenomena and programming languages like Python.
- A recent Nature technology feature suggested that such chatbots work best for small, discrete programming tasks, such as loading data, performing basic data manipulations, and creating visualizations and websites. It further suggests considering 4 guidelines while using such chatbots: verification, safety, iteration, and to anthropomorphize its performance.
- The objective of this study is to test different use cases specially to understand the effectiveness of ChatGPT in syntax auto-correction and its output generation. Considering performance of its 184 programming exercises test from Piccolo et.al, we hypothesize it to have at least 75.5% sensitivity across 8 different domain unrelated programming languages.

Method

- Following use cases were tested using module *openai* and engines “code-cushman-001” (code model) & “text-davinci-002” (general model). Engine “code-cushman-001” is a code base specific model as its was specifically trained on codebases including but not limited to python programming language, while “text-davinci-002” is a general model trained on codebase and any other associated programming language information (language history, past versions, application domains, etc.).
- So, it’s important to test use-cases against both trained codebase specific and non-specific models of ChatGPT. Use cases have been divided into 2 categories, syntax error identification and syntax error correction.
- The 2 categories are further subdivided into 8 sub-categories (data structures, conditional statements, functions, etc.) as follows. We also have provided an example code of actual query ran through *openai* module for recorded response.
- Testing code can be found here: *https://github.com/Claflin-Machine-Learning/ChatGPT-Testing/*

Results

Test Type	*Identification Sensitivity (100%)*	*Correction*
Misspelled words	+	+
Missing open, or closing brackets	+	+
Missing commas, or semicolons	+	+
Missing variables	+	-
Incorrect types	+	+
Missing operator	+	-
Incorrect Boolean values	+	+
If else statement	+	+
While loop	+	+
For loop	+	+
Incorrect functions	+	+
Incorrect list	+	+
Incorrect dictionary	+	+
Incorrect tuple	+	+
Incorrect sets	+	+
Incorrect arrays	+	+
Incorrect stack	+	+
Incorrect queue	+	+
Incorrect linked lists	+	+

Discussion

- Based on the context and the desired result, ChatGPT can analyze the provided requirements and generate pertinent test data and test cases.
- The testing procedure could be considerably accelerated as a result, giving testers more time for other activities.
- ChatGPT can be used to create more sophisticated and realistic test cases, which can aid in finding a larger variety of software faults and issues.
- ChatGPT is a useful tool that can assist in getting above the limitations brought on by having little familiarity with a certain technology. It could produce starting code for users if users are unsure about where to begin.

Limitations

- However, ChatGPT can be constrained in its ability to develop creative and sophisticated test cases because it was only trained on data that was already accessible. ChatGPT won't be able to produce reliable test cases if the training data used was not reflective of the software under test.
- Additional results validation is necessary since ChatGPT may produce test cases that are not always precise and correct. The results can occasionally be incomplete.
- Mistakes in missing variables & operator were correctly identified but not corrected in our study, such missed corrections can be overcome with transfer learning if model can be retrained with corrections in subsequent versions.
- Although this is a limited study to understand testing performance, we see a significant potential of ChatGPT for code testing and debugging.
- Further, other functional programming languages need to be tested individually. A thorough replication of this study on additional iterations and more use cases is presently being tested.