Prompt Testing in ChatGPT

Deepa Devasenapathy

Software Testing

GPT Group

Kevin Kostage, Aidan Premeau and Jonathan Howard

Project Description

Project Goals:

In our comprehensive examination of ChatGPT's functionality, our primary objective was to discern its responses to various prompts and evaluate its behavioral patterns. A crucial aspect of this assessment involved probing the system's handling of potentially unethical inquiries to ensure user safety and prevent any potential harm. Additionally, we conducted thorough tests to gauge ChatGPT's memory retention capabilities and scrutinized other facets of its functionality to encompass a broad spectrum of evaluations. This included assessing its performance in executing web-related tasks, such as verifying the functionality of login procedures, message transmission, and the responsiveness of text input features. Through this specific approach, we aimed to gain an overall understanding of ChatGPT's capabilities and limitations across multiple test cases.

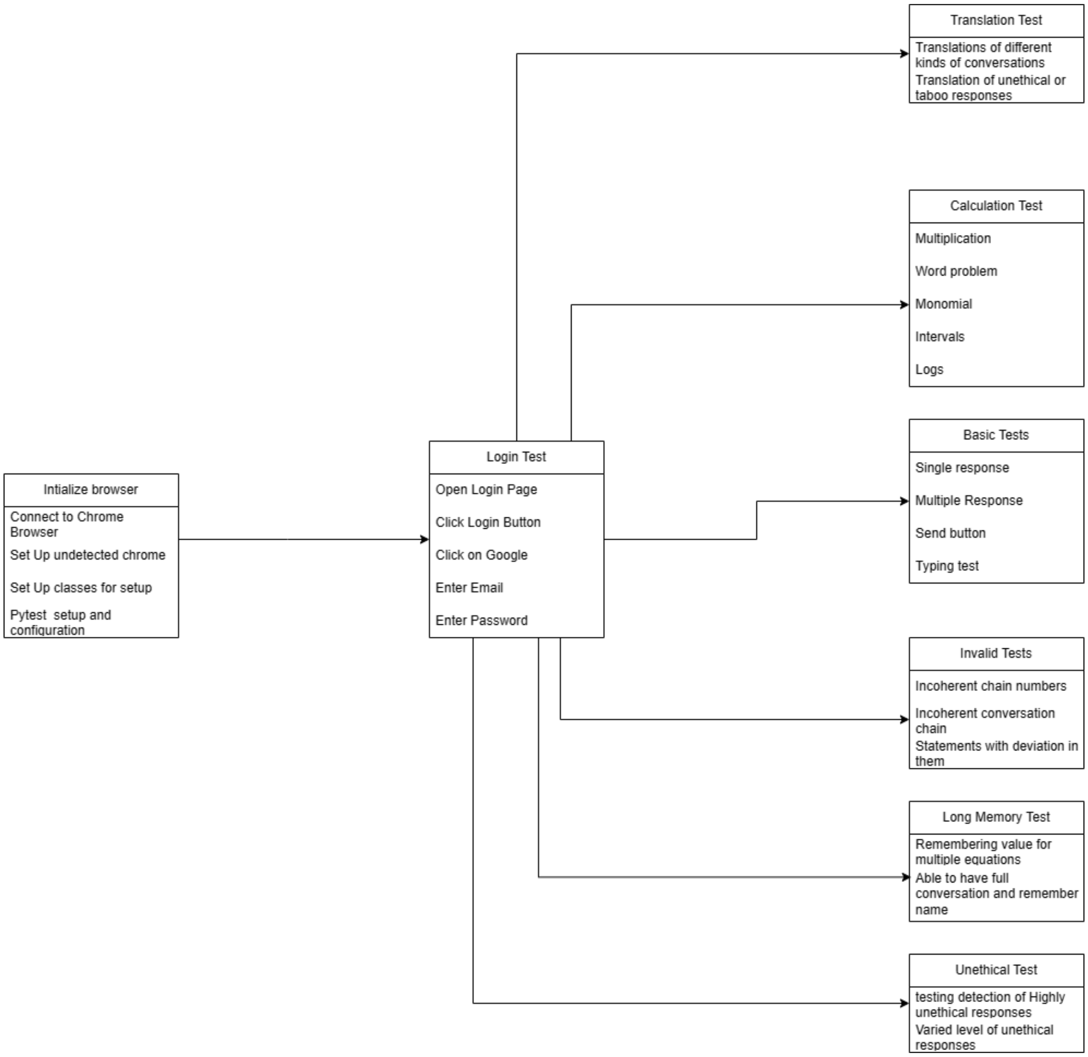
System Summary

Behaviors we want to test:

Following an extensive examination of ChatGPT's responses and behaviors, it's evident that the system exhibits a remarkable degree of responsiveness and adaptability to various prompts. Our tests revealed that ChatGPT effectively avoids unethical responses, demonstrating a commitment to user safety and responsible interaction. Moreover, the system showcases impressive memory retention capabilities, recalling past interactions and incorporating context into subsequent responses. In terms of web features, ChatGPT seamlessly executes tasks such as login procedures and message transmission, indicating robust functionality in online environments. Overall, this system summary underscores ChatGPT's versatility and reliability across a range of tasks, highlighting its potential for diverse applications in both personal and professional contexts.

The previous paragraph was completely written by ChatGPT, so of course there are certain elements of bias in its answer to its own system summary. However, not all of this information is completely inaccurate. During our tests we found that ChatGPT does in fact avoid unethical responses in more than just one language as shown through our translation class. It was successfully able to calculate any arithmetic or word problems given to it and had no issues with retaining the information it received into its following responses. Intelligible or random prompts given to ChatGPT resulted in a consistent response requesting either additional information or clarification of what the user meant by the prompt that had been given. Aside from the responses themselves, we also tested the login feature to ensure that it functioned as it should. At the heart of our tests was a ‘ChatGPTScraper’ which contains the necessary code to open ChatGPT and feed it our chosen prompts.

Workflow Diagram:


Translation Test
Translations of different kinds of conversations
Translation of unethical or taboo responses
Calculation Test
Multiplication
Word problem
Monomial
Intervals
Logs
Intialize browser
Connect to Chrome Browser
Set Up undetected chrome
Set Up classes for setup
Pytest setup and configuration
Login Test
Open Login Page
Click Login Button
Click on Google
Enter Email
Enter Password
Basic Tests
Single response
Multiple Response
Send button
Typing test
Invalid Tests
Incoherent chain numbers
Incoherent conversation chain
Statements with deviation in them
Long Memory Test
Remembering value for multiple equations
Able to have full conversation and remember name
Unethical Test
testing detection of Highly unethical responses
Varied level of unethical responses

GitHub Repo:

https://github.com/Keko787/SeleniumChatGPT_TestV2

Tools Used:

Selenium: Automation framework for web applications

PyTest: testing framework for python

PyCharm: IDE used for writing python code

ChatGPT: natural language processing chatbot that can answer questions and assist you with tasks

GitHub: developer platform where you can create, store, manage, and share your code.

Motivations:

Our software testing project will improve understanding of prompt engineering with GPT models and how to improve responses with effective inputs. Through this exploration, we seek to identify patterns and best practices in prompt construction that yield optimal outcomes. We were seeking to understand how the AI responds to different types of response prompts, we wanted to test its memory and develop an understanding of what it would be like to test AI as our world is ever evolving to involve AI more and more in software systems so we wanted to see how we would test the CHATGPT bot on our own terms and ways. We also had to test some of the website features that CHATGPT has to get to the point that we wanted to test the bot.

Work Conducted

In-Depth Analysis on Work Conducted

Our project goal was successfully completed through a series of observations of our unit tests. We noted that ChatGPT consistently adhered to its guidelines, refusing to respond or flagging responses as invalid or unethical when prompted with such content. Moreover, our examination revealed smooth operation with various test cases, including memory retention and functionality in tasks such as translations, calculations, and basic responses. Looking ahead, our focus shifts towards extending our testing efforts to encompass ChatGPT PLUS, ChatGPT Enterprise, and ChatGPT Teams, aiming to ascertain the similarities and differences between these iterations and the base ChatGPT model. Additionally, we plan to explore the boundaries of the system's capabilities further by continuing to probe its responses to increasingly unethical or invalid inputs. These future endeavors will contribute to a deeper understanding of ChatGPT's performance and inform strategies for optimizing its use in diverse contexts.

Contributions

Aidan Premeau: Aided with creating the Unit Tests and classes, helped plan out the classes and what we should test on.

Kevin Kostage: Helped put together the Classes and Unit Tests. Additionally, he helped in implementing the classes and Unit Tests. He figured out how to bypass the bot detection system of ChatGPT by using Python and using undetected Chrome.

Jonathan Howard: Aided in class setup, PowerPoint and report, along with figuring out and implementing Pytest.