

Simple Prompt Engineering for Debugging with ChatGPT

Shreyes Joshi
Oregon State University
Corvallis, Oregon
joshish@oregonstate.edu

I. INTRODUCTION

A. Problem statement

As it stands, ChatGPT is used as a tool to write, read, and debug code. However, most people do not know about its capabilities and limitations. How much can it code on its own? What information does it need? Moreover, every time ChatGPT responds to a question, its answer can change. How does the framing of the query influence the response the response? To properly and safely use ChatGPT as a debugging tool, we must answer some of these questions.

B. Motivation

ChatGPT is already used to debug software in many fields. Understanding its strengths and weaknesses can help developers use it responsibly. It can maximize the value a developer can take from ChatGPT while assuring it is not overused.

C. Relation with software engineering research

Research on the use of LLMs for debugging has been a well-investigated topic. Because it can help simplify code repair, it is a relevant topic for all software engineers. Papers exist that investigate the success of GPT in debugging, but not enough focus on prompt engineering, and maximizing the responses. Additionally, none of these focus on low-level communication with ChatGPT. They focus on development integrating GPT into debuggers.

D. Key Insight or Idea

I would like to explore the idea of prompt engineering for debugging with ChatGPT. The goal of this project is to find the best framing for a query that locates and resolves bugs in code. However, the scope of this project must be reduced to create a timeline that fits into five weeks. To do this, I would only focus on three variables: context, action words, and bug information. I would like to see how the different levels of context could assist ChatGPT in solving issues. At what point does the additional information become ineffective? Context could be measured through lines of code. I would also like to see the effect of various action words on the response. Do words such as "fix", "patch", "debug", and "show me" systematically generate better responses? Finally, I want to investigate how information about an error improves ChatGPT's response. Will an error code or error name help ChatGPT locate and repair errors?

E. Assumptions

This study assumes that ChatGPT is a tool that can and will be used for debugging. It assumes that debugging responses vary based on the query. Additionally, it assumes that there exists a generally optimal solution for debugging in all contexts. Finally, it assumes that the generally optimal solution relates to one of the three variables tested in this study.

F. Research questions

How can we write the most optimal prompts for debugging with ChatGPT by varying only the context, action word, and error information?

G. Evaluation Dataset

The solution will be evaluated on a dataset of Python code with bugs. These datasets can be found online, and simulate real issues found in real life.

H. Evaluation metrics

ChatGPT will be asked to solve solve the bugs in the Python code, using optimal queries. After evaluating the solution on the bugged Python code, we will use an automated checker such as Hypothesis to see how many bugs are fixed. The success of the study will be evaluated based on the percentage of bugs that were fixed.