Prompt Engineering for Debugging with ChatGPT

Shreyes Joshi

Problem and Motivation

- ChatGPT is used to debug code
- Those who use it do not know enough about its capabilities and limitations
- ChatGPTs responses change based on the prompt
- People try to cater the request, but it is often unpredictable
- Understanding it is key to using it responsibly
  - Maximize the assistance
  - Avoid over use
- Write safe code in simple ways

Relations with Software Engineering

- A relevant topic because of the potential for automated code repair
- Quite a bit of research already exists
- Debugging with GPT
  - GPT-3-Powered Type Error Debugging: Investigating the Use of Large Language Models for Code Repair.
- Prompt engineering for code development
  - An Approach for Rapid Source Code Development Based on ChatGPT and Prompt Engineering

Key Insight

- Determine how the framing of a query helps GPT debug
- A large scoped issue
- Would be best solved by creating algorithms that are trained to produce the best query on a case by case basis
- To reduce the scope we simplify to 3 manually tested variables
    - Context
    - Action word
    - Error information

Assumptions

- ChatGPT (GPT 3) is a tool that can be used for debugging
- Debugging responses and accuracy depend on the prompt
- There exists a generally optimal prompt framing
- It assumes that the solutions relates to one of the three variables

Research Question

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

Evaluation

- Solution will be evaluated using a dataset of Python code with bugs
- Datasets available online for testing
- We will generate an “optimal query” asking GPT to fix the code
- The fix will replace the buggy Python code
- After testing, Hypothesis will be used to find what code is debugged
- The success of the study depends on what percentage of bugs were fixed