

Bug Buster: Enhancing Unit Tests using ChatGPT-3.5

Stefan Creasta (Creasta@student.tudelft.nl)

Supervisors: Annibale Panichella, Mitchell Olsthoorn

1. Background

- Testing is crucial for ensuring that the program behaves as intended
- Achieving high mutation score can be difficult
- There have been certain Search-Based and Fuzz Testing (SBFT) tools which have showed promising results
- EvoSuite – based on evolutionary algorithms [1]
- LLMs have been used for various software applications, including test suites generation [2, 3]

2. Research Questions

Can LLMs be used to improve tests?

- To what extent can tests be improved using ChatGPT-3.5, utilizing the static approach?
- To what extent can tests be improved using ChatGPT-3.5, utilizing the dynamic approach?
- What is the most efficient number of prompts that must be sent to ChatGPT-3.5 such that the mutation score is maximized?

3. Methodology

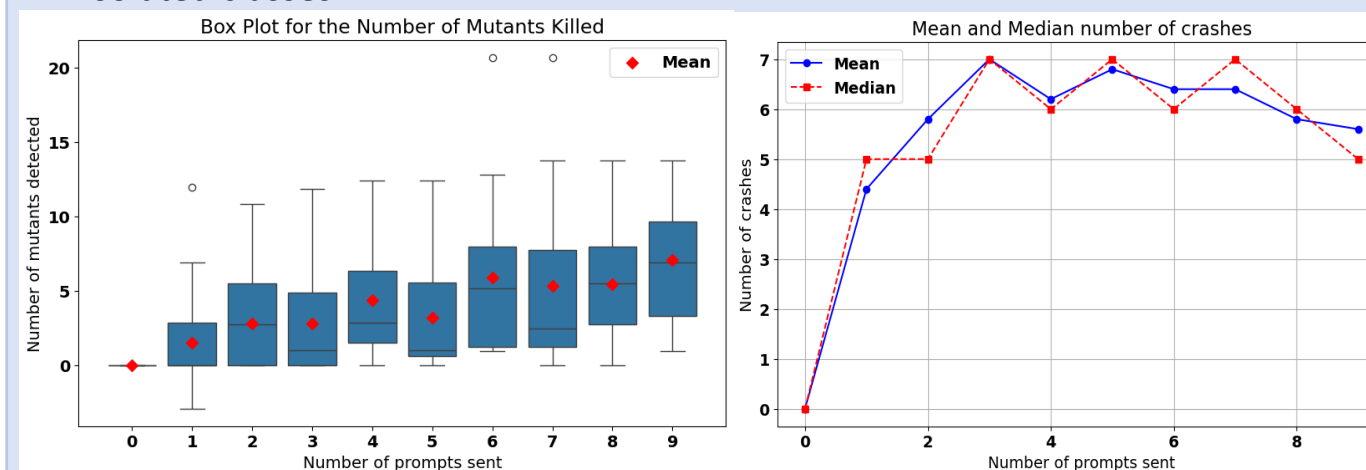
- Initially extract mutation score from original test suite
- Create the first prompt by using the classes, test cases and mutation score
- Send the prompt to GPT-3.5
- Extract the test suite from the response, while storing its mutation score or if it crashed
- Create the subsequent prompt by including the output of the suite
- Send the new prompt to the LLM and continue this iterative process until 9 total prompts have been sent
- If a test suite received from GPT-3.5 got errors, enumerate them in the next prompt to allow the LLM to fix them
- Be more explicit in results for better results: “include the correct package”, “provide the entire code”, “here are a list of mutants that I would like you to focus on” etc.

4. Benchmark and Results

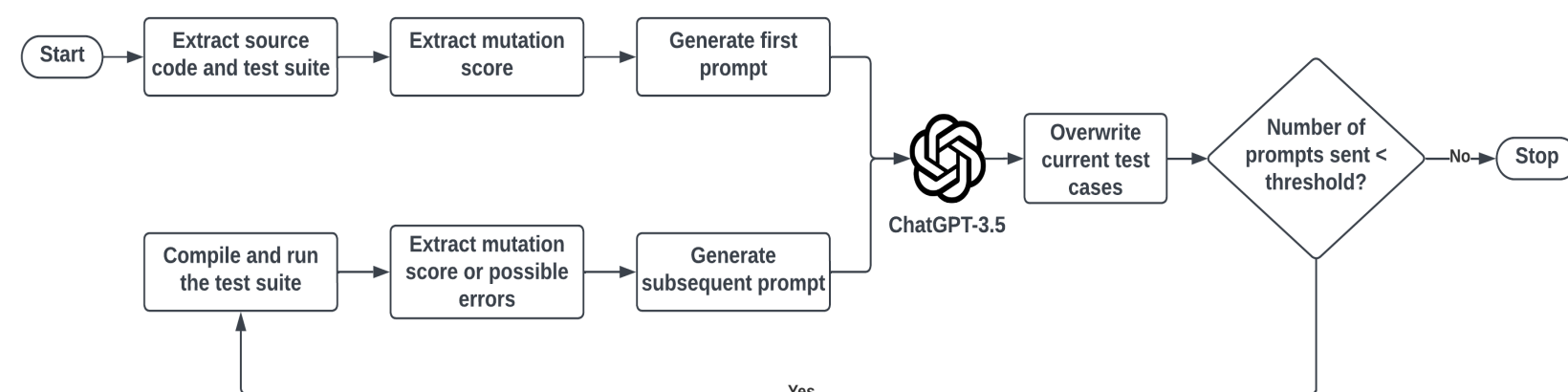
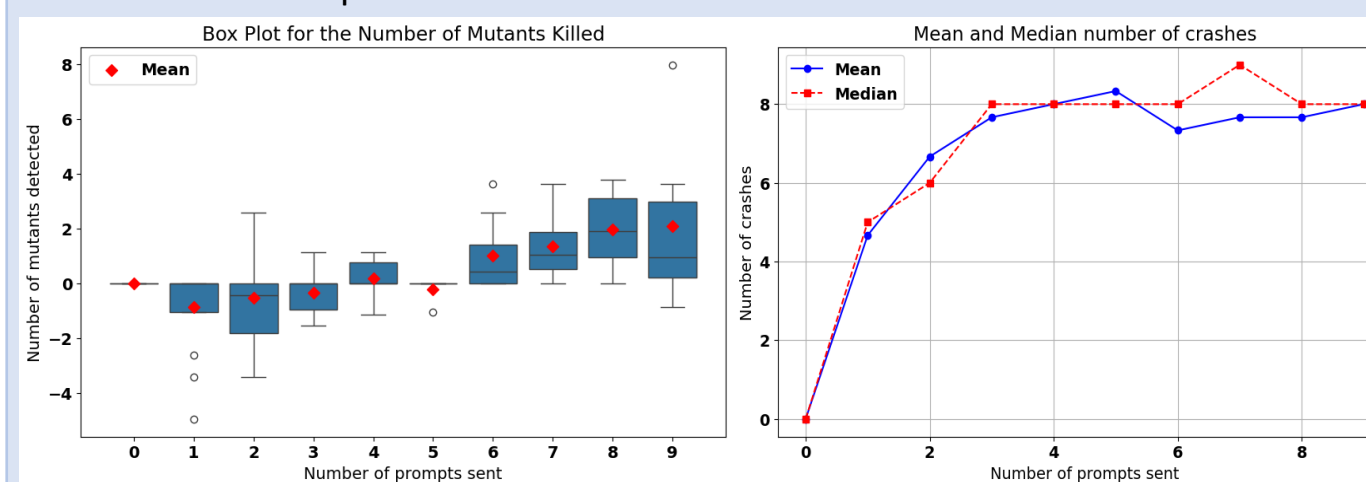
- Extracted classes from Apache Commons library and SF110

Class	Number of lines	Initial mutation score	Total number of mutants	Manual written tests	SF110
ByteVector	294	15	138		✓
Utils	175	21	24	✓	✓
BooleanComparator	190	35	24	✓	
CommandLine	198	41	32		✓
Queue	232	38	34		✓
Group 1	593	27	29	✓	
Group 2	462	29	38	✓	
Group 3	404	10	52	✓	

- Isolated classes



- Classes with dependencies



References:

- [1] Jahangirova, Gunel, and Valerio Terragni. "SBFT tool competition 2023-Java test case generation track." 2023 IEEE/ACM International Workshop on Search-Based and Fuzz Testing (SBFT). IEEE, 2023.
- [2] A. Fan, B. Gokkaya, M. Harman, M. Lyubarskiy, S. Sengupta, S. Yoo, and J. M. Zhang, "Large language models for software engineering: Survey and open problems," 2023
- [3] J. Wang, Y. Huang, C. Chen, Z. Liu, S. Wang, and Q. Wang, "Software testing with large language models: Survey, landscape, and vision," 2024