Effective Test Generation Using Pre-trained Large Language Models and Mutation Testing

Riddhi Mistry - 202201238

Hardi Naik - 202201477

Introduction to LLM-Based Test Generation

Manual Testing Challenges

Manual test case creation is time-consuming and error-prone, motivating automation.

LLM Approaches

Gemini LLM generates tests using zero-shot (no examples) and few-shot (with examples) prompting.

Mutation Testing

Used to measure test effectiveness by checking if tests detect small code changes called mutants.

Goals and Research Questions

Goal

Evaluate Gemini LLM's test generation quality using mutation testing and MUTAP refinement.

Research Questions

- Can LLMs match or exceed mutation scores of traditional tools like Pynguin?
- Is few-shot prompting more effective than zero-shot?
- Does MUTAP refinement significantly improve test effectiveness?

Experiment Steps

1

Test Generation

Gemini LLM generates tests using zero-shot and few-shot prompts for two Python programs.

2

Mutation Testing

Initial mutation scores are calculated by applying tests to mutated code versions.

3

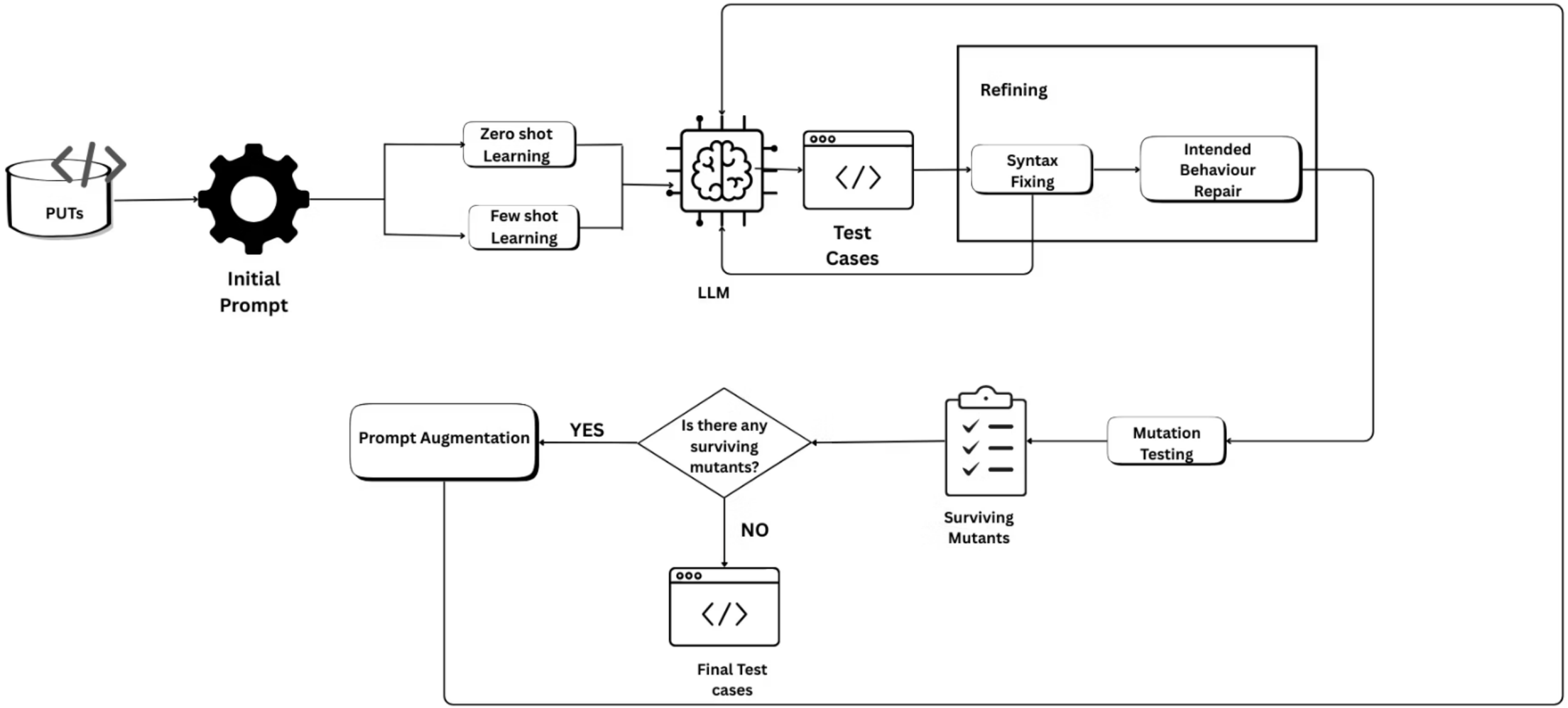
MUTAP Refinement

Surviving mutants are targeted with enhanced prompts to generate additional tests.

4

Comparison

Pynguin-generated test suites are also evaluated for mutation scores for benchmarking.


PUTs
Initial Prompt
Zero shot Learning
Few shot Learning
LLM
Test Cases
Refining
Syntax Fixing
Intended Behaviour Repair
Mutation Testing
Surviving Mutants
Is there any surviving mutants?
YES
NO
Prompt Augmentation
Final Test cases

Data Collection and Mutation Scores

Average Mutation Scores for Zero-shot and Few- shot Prompting Strategies : MUTAP

Program Strategy	Program - 1	Program - 2
Zero shot	91.27%	84.00%
Few shot	95.71%	100%

Average Mutation score : Pynguin

Program Strategy	Program - 1	Program - 2
Zero shot	87.5%	44.4%
Few shot	87.5%	58.33%

Key Findings and Outcomes

LLM Effectiveness

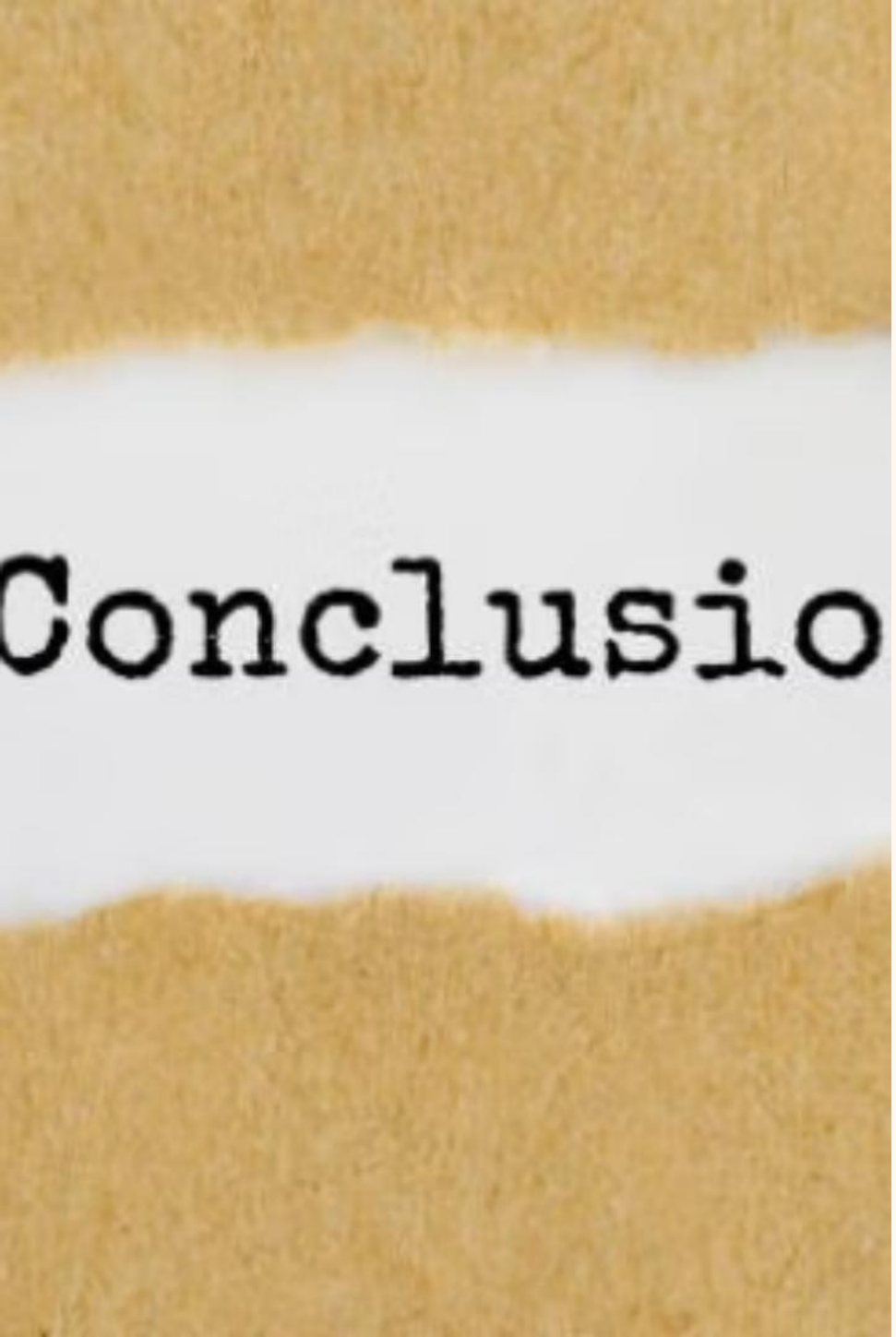
Gemini-generated tests achieve competitive or superior mutation scores compared to Pynguin.

Prompting Impact

Few-shot prompting produces better test cases than zero-shot prompting.

MUTAP Refinement

Refining tests with MUTAP significantly improves mutation scores by targeting surviving mutants.



Conclusions and Implications

Few-shot prompting combined with MUTAP refinement yields the highest mutation scores and most consistent results. Both zero-shot and few-shot LLM-generated tests outperform Pynguin.

In the future, MUTAP can be improved by testing on larger programs and supporting more languages.

Thank You