Evaluating Self-Correcting LLM Agents for Robust Test Assertion Generation

by Horia Galitianu, supervised by Annibale Panichella Mitchell Olsthoorn

Background

The Oracle Problem:
Automated test case generation tools excel at creating inputs but rely on implicit oracles or regression checks that miss deep semantic failures.

The Static Limit:
Single-pass LLM assertion generation frequently suffers from low usability, high error rates, and compilation issues because static models do not inspect execution feedback.

Research Question:
To what extent does an iterative, multi-agent framework improve the compilation, execution, and Test Strength of LLM-generated test assertions compared to static, one-shot prompting?

Approach

Agentic Architecture:
A multi-agent LangGraph workflow for context summarization, strategic planning, and assertion generation to isolate performance bottlenecks.

Iterative Feedback Loop:
A self-correcting mechanism that compiles and executes generated test candidates, using compile-time and mutation testing feedback from PITest to iteratively refine assertion logic.

Experimental Methodology:
Extracted 112 focal test prefixes from real-world Java projects (twilio-java and liqp), injected LLM-generated oracles, and measured performance using Valid Runs and Test Strength.

Results & Discussion

Agentic Improvements:
The iterative feedback loop significantly enhances reliability and quality. The Summarizer + Refining Loop configuration achieved 84.8% valid runs and an average Test Strength of 56.1%, representing a 45.9% relative increase in reliable executions and a 12.4% improvement in Test Strength compared to the static baseline.

Ablation Findings:
Execution feedback is the primary driver of reliability. Upfront strategic planning improved static prompting but added little computational value inside the loop.

Future Work:
Expanding evaluation to larger datasets and dynamically typed languages like Python or JavaScript to test the versatility of the runtime execution feedback loop.