Abstract

Imagine you are a novice programmer who has just written some code that you hope works as intended. You are able to describe what your code should do in words, but you struggle to think of test cases and implement them syntactically. You look online for help, but the test cases you find are either too complicated, not adapted to your messy beginner code, or do not compile and pass. Since you do not know the syntax well enough to fix any of these problems, you feel like giving up.

Instead of looking online, you ask a large language model (LLM) to generate test cases. However, this results in complicated tests that are not consistent or suitably documented, since LLMs are trained using a corpus that is not necessarily suited for novice programmers. Furthermore, the tests do not "wrap" around your messy beginner code correctly, since you are unable to prompt the LLM with the necessary context from your limited understanding. Finally, the tests fail at compilation and runtime since LLMs face hallucinations, which cause incorrect tests. From this, you realize that LLMs face the same issues as standard online searches.

The aim of this thesis is to develop TestPilot, a Visual Studio Code extension for Java, one of the most common programming languages used by novice programmers. A novice programmer can input a beginner natural language description of what their code should do, and receive high-quality JUnit test cases that are semantically and syntactically correct, at an appropriate level for a novice programmer, and written in a consistent style that is suitably documented.

The extension interacts with a server that completes a strategic sequence of LLM inferences to generate a suite of test cases using research-driven prompt engineering, fine-tuning, and embeddings. The resulting test cases are displayed to the novice programmer using natural language and code implementations that follow strict testing practices. The code coverage of the tests is displayed to encourage the novice programmer to explore the problem space instead of blindly copying code. It presents a "testing" and "discovery" mode of operation to help a novice programmer depending on their learning goals.

Compared to a single prompt to ChatGPT, TestPilot generates tests that are 28% more likely to compile, 26% more likely to test the intended logic from a vague prompt, and 30% more likely to include documentation through comments. A novice programmer who initially struggled to find help using online searches or standard LLMs is now able to understand tests that are catered for their level and more correct to help them learn about software testing.