

TSE - Project Progress Report

Bora Elci be2246

April 10, 2023

AI-Driven Unit Test Generation

1 Overview

The goal of this project is to create a CLI tool for generating unit tests. The tool is designed for developers who prefer using built-in text editors over IDEs. It aims to improve code correctness and maintainability while requiring minimal installation and providing high portability. The project's scope is limited to the Java programming language, and the Methods2Test dataset [1] is used to fine-tune pre-trained models.

2 Progress

I have been exploring the contents of the dataset, which has different levels of context ranging from only “focal method” to “focal method + focal class name + constructor signatures + public method signatures + public fields.” To create a more practical and useful project, I plan to utilize the highest level of context, including object parameters. I began experimenting with generation models using AWS SageMaker. One of the main challenges identified in the academic papers I read is ensuring the correctness of assert statements in the generated unit tests. To address this issue, I will explore their approaches and adapt them to my project as necessary.

3 Research Questions

1. How do different model architectures impact the performance of unit test generation?
2. How do various hyperparameter settings influence the performance of the chosen model?
3. What is the user experience of developers using the CLI tool, and how can it be further improved?

To answer the first question, I will evaluate and compare the performance of several models, including CodeBERT, CodeGPT, as well as those suggested by Dr. Kaiser’s PhD student Robin: CodeGen, PolyCoder, and CodeParrot. I will fine-tune the selected models on the Methods2Test dataset and analyze their performance to determine the most suitable model for developing the CLI tool. To address second research question, I will explore the effects of various hyperparameters, such as learning rate, optimizers, number of epochs, and batch size, on the performance of the chosen model. To answer the third question, I will conduct a user study involving a small group of participants to gather feedback on their experience using the CLI tool. Based on their input, I will identify areas for improvement and discuss the potential for extending the CLI tool to support programming languages other than Java.

4 Value to User Community

Software developers, especially those using built-in text editors like Vim, Nano, or Emacs, will benefit from this tool as it helps them ensure code correctness and maintainability through automatically generated unit tests. This lightweight solution will require minimal installation, allowing for easy portability and compatibility with multiple systems. Moreover, companies can save time and resources by incorporating these tests into their development process. I will disseminate this tool by making it available on GitHub, along with a comprehensive README file explaining its usage, installation,

and the methodology behind it. I will also provide instructions on how to replicate the findings.

5 Demo

For the 1-minute elevator pitch, I will briefly explain the motivation behind the project, the target audience, and the benefits it offers to developers. In the 5-minute demo, I will showcase the following aspects:

- Installation and setup of the CLI tool
- Generating unit tests for a sample Java code repository
- Quality of the generated unit tests
- Walkthrough of the backend service (if applicable)

6 Delivery

I will deliver the code, data, documentation, and other software artifacts for this project through a public GitHub repository. The repository will contain the following:

- Source code for the CLI tool and the backend service (if applicable)
- Artifacts of fine-tuned models (for comparison)
- README file with instructions on installation, usage, and replicating the results
- Link to the Methods2Test dataset and pre-trained models

References

- [1] Tufano, Michele, et al. “Methods2Test: A dataset of focal methods mapped to test cases.” *Proceedings of the 19th International Conference on Mining Software Repositories*. 2022, <https://dl.acm.org/doi/abs/10.1145/3524842.3528009>.