15th Conference on

DATA ANALYSIS METHODS for Software Systems

November 28–30, 2024

Druskininkai, Lithuania, Hotel “Europa Royale”
https://www.mii.lt/DAMSS

Contacts:

Dr. Jolita Bernatavičienė
jolita.bernataviciene@mif.vu.lt
Tel. (+370 5) 2109 315
Prof. Olga Kurasova
olga.kurasova@mif.vu.lt

https://doi.org/10.15388/DAMSS.15.2024
ISBN 978-609-07-1112-5 (digital PDF)

Quality Assessment of LLM Models Generated Unit Tests: Quality Metrics Completeness from Code-Aware Perspective

Dovydas Marius Zapkus, Asta Slotkienė

Institute of Data Science and Digital Technologies
Vilnius University

marius.zapkus@mif.stud.vu.lt

Unit testing is a fundamental aspect of software testing, which ensures the correctness and robustness of code implementations, but their creation requires considerable time and resources from developers. It has already been proven that special software can generate test cases using conventional methods such as SBST or random testing (Tang et al., 2024). However, the generated test's code reached high code coverage metrics but was highly unreadable by developers. Large Language Models (LLMs) can solve readability issues by learning from training data containing real human-written test code examples. However, another challenge arises, such as unit tests reaching better coverage, but they are independent of functional context (Ryan et al., 2024). Researchers suggest solving this issue by additionally introducing the context of the code fragment into LLM's training set, improving overall results and its quality metrics. Recent research with LLM-generated unit tests focuses on code coverage as a unit test quality measure (Pan et al., 2024; Lops et al., 2024; Bhatia et al., 2024). However, this is not enough, and we suggest involving additional ways in which unit tests will be reliable and understandable. These additional two ways: comparison measures based on abstract syntax trees such as CodeBLEU, RUBY, and measurements based on machine-translation metrics such as ROUGE, METEOR, chrF. According to this, this paper proposed to research and analyze how to measure the quality of the LLM-generated unit tests. In this research, three LLM models were applied, which were used for unit test generation according to the provided source codes. The generated unit tests

were evaluated by test quality metrics such as coverage and machine translation-based metrics. Our research results allow us to highlight several results of generated unit tests with several LLM models. The first observation was that LLM models generated unit test coverage that achieved an average of 76%. The second research result was that semantic and syntactic similarity based on AST was achieved up to 0,99 between LLM-generated unit tests.

Acknowledgment: The conference participation is funded by EPAM.

References

Tang, Y., Liu, Z., Zhou, Z., & Luo, X. (2024). Chatgpt vs SBST: A comparative assessment of unit test suite generation. IEEE Transactions on Software Engineering.

Ryan, G., Jain, S., Shang, M., Wang, S., Ma, X., Ramanathan, M. K., & Ray, B. (2024). Code-Aware Prompting: A Study of Coverage-Guided Test Generation in Regression Setting using LLM. Proceedings of the ACM on Software Engineering, 1(FSE), 951-971.

Pan, R., Kim, M., Krishna, R., Pavuluri, R., & Sinha, S. (2024). Multi-language Unit Test Generation using LLMs. arXiv preprint arXiv:2409.03093.

Bhatia, S., Gandhi, T., Kumar, D., & Jalote, P. (2024, April). Unit test generation using generative AI: A comparative performance analysis of autogeneration tools. In Proceedings of the 1st International Workshop on Large Language Models for Code (pp. 54-61).

Lops, A., Narducci, F., Ragone, A., Trizio, M., & Bartolini, C. (2024). A System for Automated Unit Test Generation Using Large Language Models and Assessment of Generated Test Suites. arXiv preprint arXiv:2408.07846.