Capstone Project Proposal: Enhancing and Extending Sionna Dataset for Comprehensive Code Logic Comprehension Evaluation

Ahmed Alhammadi, Hang Zou

Digital Science Research Center
Technology Innovation Institute

July 10, 2024

1 Introduction

This project aims to extend the previously generated SIONNA dataset, which initially focused on generating telecom simulation scenarios, by including logic comprehension tasks. The primary objective is to enhance Large Language Models' (LLMs) comprehension abilities in handling logically equivalent yet distinct code structures. This will involve evaluating the models using the original SIONNA [1] dataset codes augmented with obfuscated and line-shuffled variations. This proposal outlines the objectives, methodology, and expected outcomes of the project.

2 Background

The initial phase of the project successfully created a dataset comprising 100 pairs of human instructions and corresponding telecom simulation code using the SIONNA library. The primary focus of that phase was to facilitate telecom simulation scenario development by using the dataset to potentially fine-tune LLMs for generating and modifying simulation code.

Despite these considerable achievements, we recognize a gap in the models' ability to comprehend the underlying logic of code variations—such as obfuscated or line-shuffled code. This gap becomes particularly evident when attempting to go beyond mere syntactic understanding to genuinely grasp the functional and logical equivalence of different code representations.

Moreover, recent advancements in LLM research, such as those presented by Shaik et al. [2], emphasize the importance of using comprehensive datasets that incorporate not just source code but also associated metadata and documentation to enhance the understanding of scientific software. This underscores the need for datasets that can better train models for logical comprehension rather than simple pattern recognition.

Inspired by the study “Is Next Token Prediction Sufficient for GPT? Exploration on Code Logic Comprehension” by Mengnan Q. et al. [3], this project proposes new tasks—Logically Equivalent Code Selection—designed to evaluate the logical comprehension capabilities of LLMs. This extension will focus on creating versions of the original code that include logical variations such as obfuscation and line shuffling to evaluate and improve LLMs’ understanding of code logic.

3 Objectives

- To extend the existing SIONNA dataset by adding variations such as obfuscated and line-shuffled versions of the original code.
- To evaluate the performance of LLMs, like GPT-4 and Code LLAMA, on tasks requiring code logic comprehension.
- To refine and validate these extensions by analyzing model performance and improving dataset quality.

4 Related Works

4.1 LLMs in Code Generation and Comprehension

With the advent of models like Codex and GPT-4, there has been significant progress in the field of code generation. Codex, for instance, fine-tuned from GPT-3, demonstrated impressive capabilities in translating natural language instructions into code. However, its underlying mechanism, primarily driven by next token prediction, revealed limitations in understanding code logic beyond syntactic structures. It often struggles with tasks that require maintaining logical consistency across multiple lines or understanding the functional equivalence of code variations.

4.2 Advanced Pretraining Techniques

Recent works have explored the limitations of traditional next token prediction tasks. For example, “Logically Equivalent Code Selection” introduced by Mengnan Q. et al., highlights the importance of training models not just to predict the next token but to distinguish between logically equivalent and non-equivalent code. This paper proposed perturbing the code by obfuscating identifiers or shuffling lines to evaluate the LLMs’ understanding of code logic. This approach revealed significant gaps in current models’ capabilities, suggesting the need for more sophisticated pretraining techniques, such as “Next Token Prediction+”, which includes tasks designed to steer the model towards better logical comprehension.

4.3 Evaluation Frameworks

Evaluation frameworks like CodeXGLUE and benchmarks like HumanEval have been pivotal in assessing the performance of code generation models. However, these assessments are limited to syntactic and line-level code completion tasks. The new task of logically equivalent code selection, proposed by Mengnan Q. et al., adds a new dimension by requiring models to not just generate but to comprehend code logic, thereby providing a more rigorous and comprehensive evaluation metric.

5 Methodology

5.1 Dataset Extension

5.1.1 Extension with Original Code and Obfuscated Code

To extend the existing SIONNA dataset, we will:

1. **Original Code Review:** Extract original telecom simulation codes from the existing SIONNA dataset.

2. **Obfuscation Process:** Utilize techniques such as identifier renaming, variable name abstraction, and function name changes to create semantically equivalent but textually distinct code. This involves using tools like Abstract Syntax Tree (AST) parsers to ensure the structural integrity of the code is maintained.

3. **Instruction-Obfuscated Code Pairing:** Pair each original instruction from the dataset with its obfuscated code counterpart, ensuring the logical equivalence is preserved while syntactically differing.

5.1.2 Extension with Original Code and Line Shuffled Code

1. **Original Code Review:** Extract original telecom simulation codes from the existing SIONNA dataset.

2. **Line Shuffling Process:** Randomly shuffle lines of code within methods or logical blocks, ensuring that logical coherence is disrupted but the syntax remains valid. This will involve using dependency analysis tools to identify lines that can be shuffled without causing syntax errors.

3. **Instruction-Shuffled Code Pairing:** Pair each original instruction from the dataset with its shuffled line code counterpart, ensuring that the logical flow is intentionally altered while maintaining syntactic correctness.

5.2 Evaluation

5.2.1 Experimental Setup

To evaluate the performance of LLMs on the extended dataset, we will use:

- **Models:** GPT-4 and a compact model like Code Llama.
- **Tasks:** Models will be required to determine the logical equivalence of provided code pairs. Each task will involve identifying whether the obfuscated or shuffled code maintains the original code's logic.
- **Evaluation Metrics:** Accuracy in selecting logically equivalent pairs, and understanding disruptions caused by obfuscation and line shuffling will be measured.

5.2.2 Performance Analysis

1. **Logical Equivalence Detection:** Measure how well models can detect equivalence between the original and obfuscated/shuffled code versions.
2. **Error Analysis:** Detailed examination of cases where models fail to recognize equivalence, to understand common pitfalls and areas needing improvement.
3. **Baseline Comparison:** Compare performance against baseline models to gauge improvement from dataset extensions.

6 Expected Outcomes

Upon completion, the extended dataset will encompass a broader array of telecom simulation scenarios with corresponding human instructions and code modifications (obfuscated and line-shuffled). Evaluations will yield insights into LLMs' abilities to comprehend code logic, thereby highlighting areas for further improvement in LLM training methodologies. The outcome will be a significant step forward in enhancing LLMs' practical applications in software engineering and automated code generation.

7 Conclusion

This project aims to extend the existing SIONNA dataset to include logically equivalent but distinct code variations, facilitating better LLM training and evaluation. By enhancing code logic comprehension capabilities, the project aims to bridge the gap between text-based code understanding and logic-based code execution, providing significant advancements in the field of AI-powered software development tools.

References

[1] J. Hoydis, S. Cammerer, F. A. Aoudia, A. Vem, N. Binder, G. Marcus, A. Keller, "Sionna: An open-source library for next-generation physical layer research," arXiv preprint arXiv:2203.11854, 2022, available at `https://arxiv.org/abs/2203.11854`.

[2] K. Shaik, D. Wang, W. Zheng, Q. Cao, H. Fan, P. Schwartz, Y. Feng, "S3LLM: Large-Scale Scientific Software Understanding with LLMs using Source, Metadata, and Document," arXiv preprint arXiv:2403.10588, 2024, available at `https://arxiv.org/abs/2403.10588`.

[3] M. Q, Y. Huang, Y. Yao, M. Wang, B. Gu, N. Sundaresan, "Is Next Token Prediction Sufficient for GPT? Exploration on Code Logic Comprehension," arXiv preprint arXiv:2404.10587, 2024, available at `https://arxiv.org/abs/2404.10587`.