Title: Capabilities of Large Language Models in Program Analysis Tasks

Introduction: In recent years, after the emergence of the transformer, Large Language Models(LLMs) have not only become the forefront of text generation tasks but have started to redefine the realm of static labor-heavy program analysis tasks. Program analysis tasks are crucial to ensure software security, robustness, and efficiency. Failing to ensure the quality of the software costs 2.26 trillion dollars annually for the tech giants in the US[1]. Though effective, traditional approaches often involve static and dynamic analysis techniques and come with inherent limitations. With their ability to understand and generate human-like text, LLMs offer a different perspective on program understanding. Their capacity to comprehend vast amounts of code and natural language opens avenues for more nuanced analyses, providing insights into program behavior that were previously inaccessible[2]. However, despite their potential, utilizing LLMs in program analysis tasks still needs to be explored. While existing literature acknowledges their capabilities in natural language understanding and generation, their adaptation to code comprehension poses challenges. There exists a gap in understanding how LLMs can be effectively used in existing program analysis workflows, which is the primary motivation behind this work. This research aims to bridge the gap by exploring the capabilities of LLMs in program analysis tasks. By developing novel methodologies and frameworks, we strive to utilize the power of LLMs to enhance program comprehension, bug detection, and optimization. We seek to demonstrate the effectiveness and efficiency of integrating LLMs into existing program analysis pipelines through empirical studies and real-world applications.

References:

[1] K. Wiegers, “The cost of quality in software,” Medium, https://medium.com/geekculture/the-cost-of-quality-in-software-5f113a26e759 (accessed Mar. 19, 2024).

[2] L. Zabala, V. Liventsev, J. Febres, and R. Sterling, Comparison between program synthesis with large language models and model predictive control for buildings optimal operation, Dec. 2023. doi:10.21203/rs.3.rs-3735947/v1