How does sliding window truncation impact the defect localization precision of LLaMA 3.2 compared to fixed-hea

Assignee Research

June 10, 2026

Abstract

In recent years, JavaScript has become the most widely used programming language, especially in web development. However, writing secure JavaScript code is not trivial, and programmers often make mistakes that lead to security vulnerabilities in web applications. Large Language Models (LLMs) have demonstrated substantial advancements across multiple domains, and their evolving capabilities indicate their potential for automatic code generation based on a required specification, including automatic bug fixing. In this study, we explore the accuracy of LLMs, namely ChatGPT and Bard, in finding a

1 Introduction

This paper examines: A Study of Vulnerability Repair in JavaScript Programs with Large Language Models. Research question: How does sliding window truncation impact the defect localization precision of LLaMA 3.2 compared to fixed-head or fixed-tail truncation when evaluated on JavaScript codebases with similar complexity to BugsInPy?.

2 Methodology

Systematic literature search across multiple databases yielded 13 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 3.3/10.

3 Results

13 papers retrieved. 0 claims extracted; 0 independently verified. Quality review score: 3.3/10.

4 Limitations

This report is a machine-generated literature synthesis and does not constitute original research. Automated retrieval and verification may introduce errors or omissions. Review scores reflect automated assessment, not human peer review. Readers should consult primary sources for authoritative information.

References

- http://arxiv.org/abs/2306.09896v5
- http://arxiv.org/abs/2403.13193v1
- http://arxiv.org/abs/2511.07364v1