Semantic Literature Retrieval and Code Context Engineering for Multi-File Project Accuracy

Assignee Research

May 30, 2026

Abstract

This report synthesises findings from 15 peer-reviewed papers addressing the following research question: What is the impact of combining semantic literature retrieval (Elicit) with code-focused context engineering on the accuracy of generated code for niche domains in multi-file projects, measured by. Large Language Models (LLMs) have shown promise in automating code generation and software engineering tasks, yet they often struggle with complex, multi-file projects due to context limitations and knowledge gaps. We propose a novel context engineering workflow that combines. 15 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 2.5/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Context Engineering for Multi-Agent LLM Code Assistants Using Elicit, NotebookLM, ChatGPT, and Claude Code. Research question: What is the impact of combining semantic literature retrieval (Elicit) with code-focused context engineering on the accuracy of generated code for niche domains in multi-file projects, measured by pass@k on Re-Code?.

2 Methodology

Systematic literature search across multiple databases yielded 15 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 2.5/10.

3 Results

15 papers retrieved. 15 claims extracted; 0 independently verified. Quality review score: 2.5/10.

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.

5 Extracted Claims

Claim	Verified	Confidence
The context-engineered assistant was evaluated on a sample of 5 non-trivial development tasks in the RainMakerz codebase	×	0.03
The multi-agent approach succeeded in more tasks than the baseline single-agent Claude.	×	0.14
The multi-agent approach required fewer iterations than the baseline single-agent Claude.	×	0.12
The multi-agent system often produced working solutions on the first attempt, whereas the baseline frequently needed fol	×	0.05
In the 'CustomBlock' case study, the Planner agent broke the task into 4 steps.	×	0.03
In the 'CustomBlock' case study, an initial test failure occurred because the new block type was not added to a serializ	×	0.02
In the 'CustomBlock' case study, the second test run passed all unit and integration tests after the serialization confi	×	0.02
The 'CustomBlock' feature was completed in one automated session by the multi-agent system.	×	0.05
When prompted for the same 'CustomBlock' task, the single Claude instance produced only the React component and forgot t	×	0.02
The single Claude instance's output for the 'CustomBlock' task resulted in runtime errors.	×	0.03
Every function or class used by the code generated by the multi-agent system existed in the repository.	×	0.05
The baseline system guessed function or variable names, such as referring to a non-existent getEvents() API.	×	0.06
The system employs GPT-5 as an Intent Translator to reformulate user queries into structured specifications.	×	0.06
The retrieval pipeline utilizes a dual-index backend consisting of Chroma and Zilliz.	×	0.03
The retrieval pipeline includes code-aware chunking and a unified adapter.	×	0.06

References

- http://arxiv.org/abs/2402.12317v2
- http://arxiv.org/abs/2212.10264v1
- http://arxiv.org/abs/2508.08322v1