Generalization of Llama3-70B and Codestral-34B to Low-Resource Programming Languages

Assignee Research

May 30, 2026

Abstract

This report synthesises findings from 14 peer-reviewed papers addressing the following research question: How do Llama3-70B and Codestral-34B generalize to low-resource programming languages beyond Java and Python, such as Rust or Go, when fine-tuned on limited domain-specific datasets, as measured by. Domain-specific languages that use a lot of specific terminology often fall into the category of low-resource languages. Collecting test datasets in a narrow domain is time-consuming and requires skilled human resources with domain knowledge and training for the annotation task. 13 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 3.4/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Automated Collection of Evaluation Dataset for Semantic Search in Low-Resource Domain Language. Research question: How do Llama3-70B and Codestral-34B generalize to low-resource programming languages beyond Java and Python, such as Rust or Go, when fine-tuned on limited domain-specific datasets, as measured by perplexity and completion accuracy?.

2 Methodology

Systematic literature search across multiple databases yielded 14 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 3.4/10.

3 Results

14 papers retrieved. 13 claims extracted; 0 independently verified. Quality review score: 3.4/10.

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
Combining multiple encoders with a generative LLM (GPT-4o) to reassess relevance scores increases inter-coder agreement	×	0.11
The approach of combining multiple encoders with GPT-4o improves the F1-score by 1.5 times.	×	0.04
Ensemble learning improves machine learning performance by combining predictions from multiple models, enhancing accurac	×	0.07
Standalone LLMs perform worse than human annotators in data annotation tasks.	×	0.06
In the implementation described, documents (text logs) ranged in size from a sentence to a paragraph and were truncated	×	0.03
Queries were generated using GPT-4o from randomly selected documents containing at least 100 characters.	×	0.02
In Dataset A, the Combined method achieved a Krippendorff's alpha of 67.03, compared to 50.30 for the Ensemble method.	×	0.04
In Dataset A, the Combined method achieved an F1-score of 53.42, while the GPT-4o-SE method achieved 38.63.	×	0.02
In Dataset B, the Combined method achieved a Krippendorff's alpha of 68.69.	×	0.02
The average relevance score for the Combined method across categories 0-3 is 50.2.	×	0.04
The azure-text-embedding-3-large model achieved an nDCG@10 score of 69.	×	0.03
The sentence-transformers/multi-qa-mpnet-base-cos-v1 model achieved an average score of 35.29 across evaluated metrics.	×	0.04
The dataset used for evaluation contains 79.6K documents, 20 queries, and 406 relevant documents.	×	0.03

References

- http://arxiv.org/abs/2412.10008v1
- http://arxiv.org/abs/2306.06371v1

- http://arxiv.org/abs/2311.01767v2