Uncertainty estimation from multi-model ensembles for improved selection accuracy on HumanEval+

Assignee Research

June 11, 2026

Abstract

Large language models (LLMs) have significantly improved code generation, particularly in one-pass code generation. However, most existing approaches focus solely on generating code in a single programming language, overlooking the potential of leveraging the multi-language capabilities of LLMs. LLMs have varying patterns of errors across different languages, suggesting that a more robust approach could be developed by leveraging these multi-language outputs. In this study, we propose Multi-Programming Language Ensemble (MPLE), a novel ensemble-based method that utilizes code generation across

1 Introduction

This paper examines: Multi-Programming Language Ensemble for Code Generation in Large Language Model. Research question: Can uncertainty estimates derived from multi-model ensembles improve selection accuracy over single-model sampling strategies on the HumanEval+ dataset?.

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 7.4/10.

3 Results

14 papers retrieved. 17 claims extracted; 13 independently verified. Quality review score: 7.4/10.

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
The MPLE framework was evaluated on the HumanEval and HumanEval-plus benchmarks.	×	0.11
HumanEval is designed for text-to-code generation tasks specifically for the Python programming language.	×	0.15
HumanEval evaluates generated code based on its ability to pass unit tests with specified requirements.	✓	0.20
HumanEval-plus extends the HumanEval dataset by incorporating additional valid unit test cases.	✓	0.22
Pass@1 accuracy is computed using hidden test cases to assess the performance of generated code.	✓	0.18
Pass@1 measures the percentage of tasks for which the model's top output passes all hidden test cases.	✓	0.22
Experiments were conducted using the proprietary LLMs GPT3.5-turbo (gpt-3.5-turbo-0125), GPT-4o-mini (gpt-4o-mini-2024-0	✓	0.34
Experiments were conducted using the open-source LLMs Llama3.1-8b-instruct, Llama3.1-70b-instruct, and Llama3.1-405b-ins	✓	0.24
Performance results on the HumanEval benchmark are presented in Table 1.	×	0.11
Performance results on the HumanEval-plus benchmark are presented in Table 2.	×	0.12
The MPLE framework iteratively refines code by leveraging the strengths of different programming languages.	✓	0.22
The MPLE framework integrates reflection algorithms and MCTS to enhance robustness and accuracy.	✓	0.18
The code generation task is formulated as a triplet (Q, Tv, Th), where Q is the task description, Tv represents visible	✓	0.31
In the problem formulation, the LLM generates an initial program P0 using Q and Tv.	✓	0.19
The generated program P0 is refined iteratively to produce a sequence of programs until one passes all visible tests in	✓	0.32
The final output program P* is evaluated on hidden test cases Th to verify correctness.	✓	0.29
Hidden test cases (Th) are used only once for pass@1 evaluation and remain hidden during the code generation and refinem	✓	0.22

References

- http://arxiv.org/abs/2410.12381v3
- http://arxiv.org/abs/2605.16567v1
- http://arxiv.org/abs/2409.04114v1