CodeT5 Cross-Domain Generalization from Python to Low-Resource Languages via Pretraining Data Ratios

Assignee Research

June 7, 2026

Abstract

This report synthesises findings from 14 peer-reviewed papers addressing the following research question: How does the ratio of code-to-natural-language pretraining data influence CodeT5's cross-domain generalization from Python to low-resource languages like Rust or Go when evaluated using the MBPP. 14 claims were extracted from source literature; 0 were independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 3.8/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: How Does Code Pretraining Affect Language Model Task Performance?. Research question: How does the ratio of code-to-natural-language pretraining data influence CodeT5's cross-domain generalization from Python to low-resource languages like Rust or Go when evaluated using the MBPP benchmark?.

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 3.8/10.

3 Results

14 papers retrieved. 14 claims extracted; 0 independently verified. Quality review score: 3.8/10.

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 paper measures performance on three compositional generalization benchmarks and Big-Bench tasks.	×	0.08
Performance is quantified by calculating lines of best fit between performance and code mixture.	×	0.07
Compositional generalization is a measure of how well a learner can generate and interpret novel, licit combinations of	×	0.02
The presence of source code in pretraining data may aid models in making compositional generalizations.	×	0.11
Source code often contains sequences in which a finite set of primitives (e.g., variable and method identifiers) are bro	×	0.03
The paper evaluates whether increased code mixture enables compositional generalization by finetuning pretrained models	×	0.06
COGS and COGS-vf both divide their generalization split into two parts based on generalization type: lexical and structu	×	0.03
The paper uses causally-masked decoder-only transformer language models for pretraining.	×	0.07
The models have roughly 374 M parameters.	×	0.03
All models in the competitive setting were trained with Ntotal = 132 B tokens.	×	0.03
Models in the additive setting were trained with Nlang = 132 B tokens, and hence Ntotal varying between 132 B tokens and	×	0.03
The paper uses a batch size of 128, meaning that models were trained for between 1 M and 2 M steps, depending on the mix	×	0.02
For each combination of code mixture and setting, the paper pretrains models from five different random seeds.	×	0.05
Full replication of the pretraining procedure outlined in the paper would take roughly 750 TPU-days of compute.	×	0.02

References

- http://arxiv.org/abs/2412.10008v1
- http://arxiv.org/abs/2409.04556v2
- http://arxiv.org/abs/2303.12869v1