JaCoText Performance on HumanEval Under Varying Java-Python Pretraining Ratios

Assignee Research

June 7, 2026

Abstract

This report synthesises findings from 11 peer-reviewed papers addressing the following research question: How does varying the proportion of Java vs. Python code in pretraining data affect the performance of JaCoText on the HumanEval benchmark, measured by pass@1 accuracy. 14 claims were extracted from source literature; 1 was independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 4.5/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: JaCoText: A Pretrained Model for Java Code-Text Generation. Research question: How does varying the proportion of Java vs. Python code in pretraining data affect the performance of JaCoText on the HumanEval benchmark, measured by pass@1 accuracy?.

2 Methodology

Systematic literature search across multiple databases yielded 11 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 4.5/10.

3 Results

11 papers retrieved. 14 claims extracted; 1 independently verified. Quality review score: 4.5/10.

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
JaCoText is a model based on Transformers neural network designed to generate Java source code from natural language tex	✓	0.36
CodeGPT is trained from scratch on the CodeSearchNet dataset.	×	0.02
CodeGPT-adapted is initialized from GPT-2 pretrained weights.	×	0.03
PLBART uses the same architecture as BART-base.	×	0.01
PLBART utilizes three noising strategies: token masking, token deletion, and token infilling.	×	0.00
CoTexT uses the same architecture as T5base.	×	0.01
CoTexT is trained on both unimodal and bimodal data using the CodeSearchNet Corpus and GitHub Repositories.	×	0.06
On the CONCODE dataset, the T5-base model achieved a BLEU score of 32.74, an EM score of 18.65, and a CodeBLEU score of	×	0.03
On the CONCODE dataset, the PLBART model achieved a BLEU score of 36.69, an EM score of 18.75, and a CodeBLEU score of 3	×	0.04
On the CONCODE dataset, the CoTexT-1CC model achieved a BLEU score of 37.40, an EM score of 20.10, and a CodeBLEU score	×	0.03
On the CONCODE dataset, the JaCoText-L-2CC-PL model achieved a BLEU score of 39.87, an EM score of 22.45, and a CodeBLEU	×	0.06
JaCoText-L-2CC-PL achieved the highest BLEU, EM, and CodeBLEU scores among all models listed in Table II.	×	0.04
Reference [17] used a BiLSTM encoder and an RNN decoder to generate syntactically valid parse trees.	×	0.02
Reference [26] used a Bi-LSTMs encoder to compute contextual representations of natural language and an LSTM-based RNN d	×	0.07

References

- http://arxiv.org/abs/2410.12381v3

- http://arxiv.org/abs/2303.12869v1
- http://arxiv.org/abs/2412.21199v2