CodeT5 and JaCoText Performance on Adversarially Perturbed vs. Original Code Repairs

Assignee Research

June 7, 2026

Abstract

This report synthesises findings from 13 peer-reviewed papers addressing the following research question: How does the performance of CodeT5 and JaCoText in repairing adversarially perturbed code from the QuixBugs benchmark compare to their performance on the original, unperturbed version, measured by. 19 claims were extracted from source literature; 1 was independently verified against retrieved documents. An automated multi-reviewer quality assessment produced a score of 4.3/10. This report is a machine-generated literature synthesis and does not constitute original research.

1 Introduction

This paper examines: Program Repair with Minimal Edits Using CodeT5. Research question: How does the performance of CodeT5 and JaCoText in repairing adversarially perturbed code from the QuixBugs benchmark compare to their performance on the original, unperturbed version, measured by correct fix rates and semantic preservation?.

2 Methodology

Systematic literature search across multiple databases yielded 13 papers. Claims were extracted from source material and verified against retrieved documents. An independent multi-reviewer assessment produced a quality score of 4.3/10.

3 Results

13 papers retrieved. 19 claims extracted; 1 independently verified. Quality review score: 4.3/10.

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 model has 220M parameters.	×	0.03
The transformers library is used to load the model and conduct the fine-tuning.	×	0.06
Generated programs are executed on an isolated judge system to validate functional correctness.	×	0.04
The judge system uses hidden test cases for the programming problem provided by AOJ.	×	0.05
Training and inference are conducted in a GPU environment with one NVIDIA A100 40GB GPU.	×	0.02
Experiments for naive models and evaluations by the judge systems are conducted in a CPU environment.	×	0.04
The best BLEU score for Seq2Seq is 88.84 at epoch 16.	×	0.02
The best BLEU score for CodeT5 is 94.68 at epoch 6.	×	0.02
The training takes 21.0 hours for 21 epochs in the Seq2Seq model.	×	0.02
The fine-tuning takes 11.5 hours for 11 epochs in the CodeT5 model.	×	0.04
The pass@100 of 91.95% by CodeT5 indicates that the fine-tuned CodeT5 can generate at least one correct program for 91.9	✓	0.28
The pass@100 of 62.58% by Seq2Seq indicates that the Seq2Seq can generate at least one correct program for 62.58% of wro	×	0.15
CodeT5 performs the best on BLEU and Exact Match, compared with all baseline models.	×	0.06
Gulwani et al. proposed CLARA using the syntactic difference (tree-edit-distance) as the cost function to find the progr	×	0.12
Lu et al. proposed a fast and accurate program repair tool, FAPR, that outperformed CLARA in suggesting the correct and	×	0.08
Parihar et al. applied program repair, enabling automatic grading of incorrect submissions that contain syntax errors us	×	0.05
CodeT5 has demonstrated the capability in program repair on the CodeXGLUE benchmark.	×	0.08
Using large language models (LLMs) trained on source code has shown the capability in APR.	×	0.09
Codex has shown the ability to fix security bugs, improve time performance, and enhance code readability.	×	0.08

References

- http://arxiv.org/abs/2309.14760v1
- http://arxiv.org/abs/2303.12869v1
- http://arxiv.org/abs/2111.03922v1