Learning to Repair Codes via Code Coverage Testing

Hyeongjun Jeon°, Gyeongju Lee, Aditi and Sang-Ki Ko

Computational Intelligence & Data Analytics Laboratory, Department of AI, University of Seoul

UNIVERSITY OF SEOUL

Research Question

"Can the coverage-based fault localization method effectively be used for deep learning-based automated program repair?"

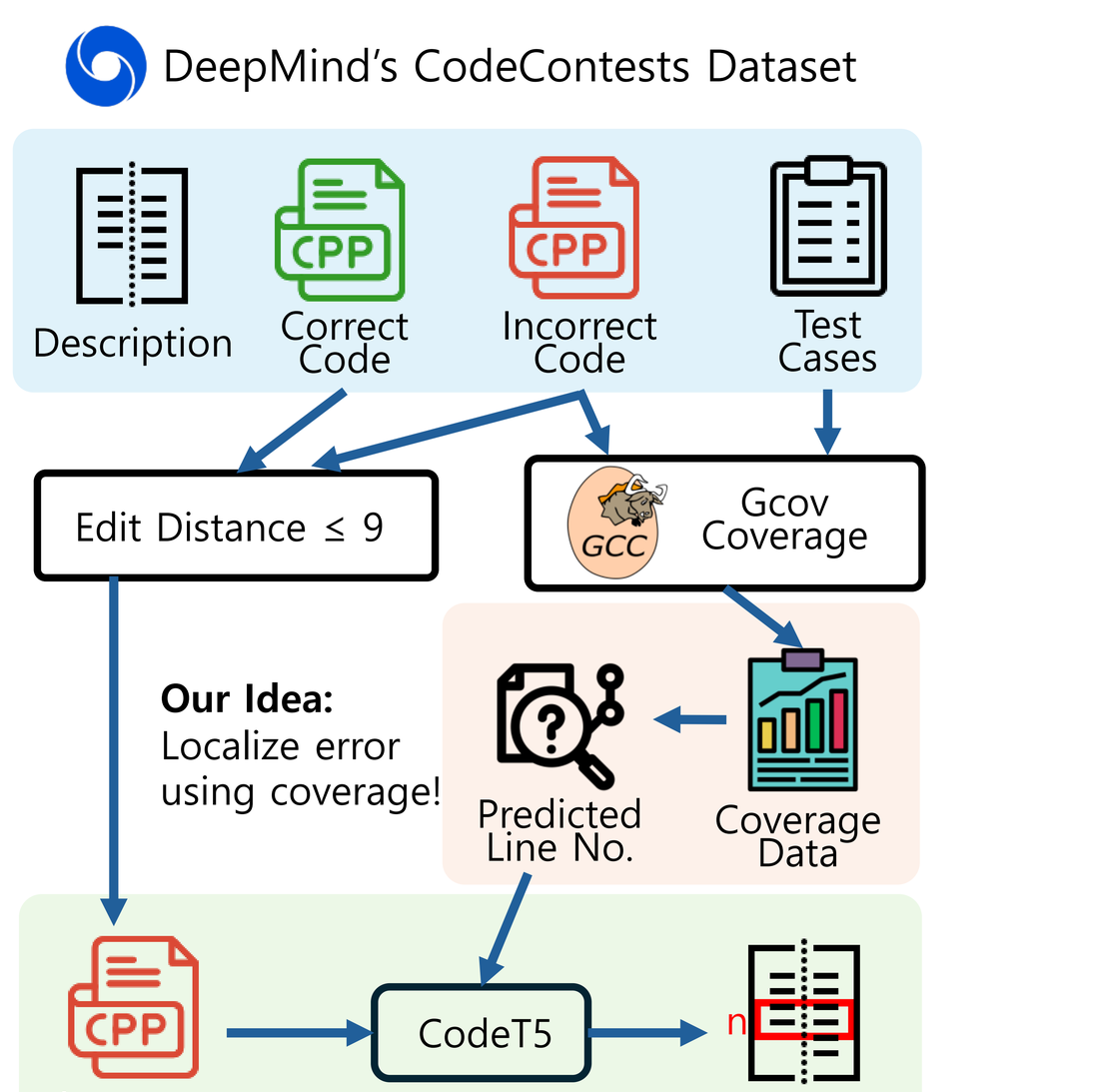
Contributions

- Enhanced the **CodeT5 model** with **code coverage analysis** to identify and correct logical errors more effectively.
- Emphasized **precise error localization** for improved correction accuracy.
- Validated with the **CodeContests dataset** of correct and incorrect solutions.
- Combined **traditional debugging methodologies** with machine learning for **APR**.

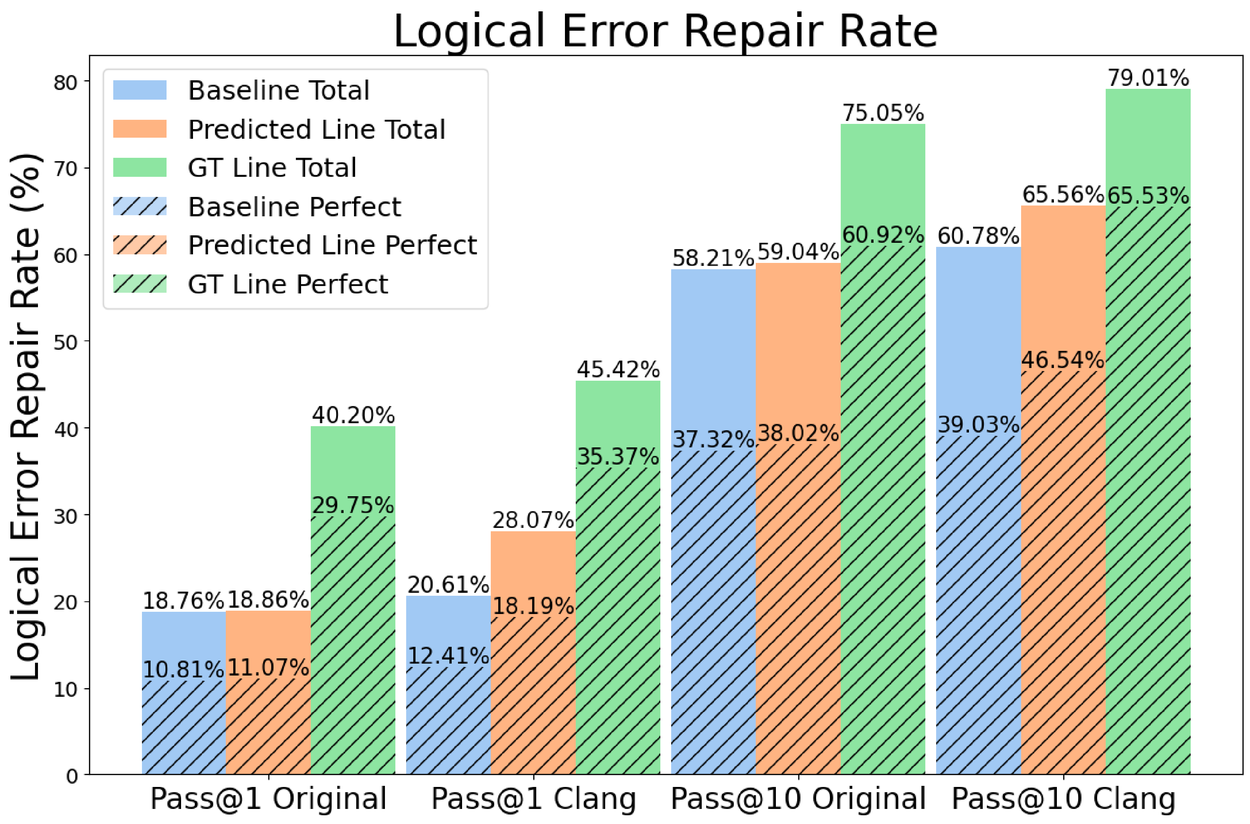
Dataset Construction

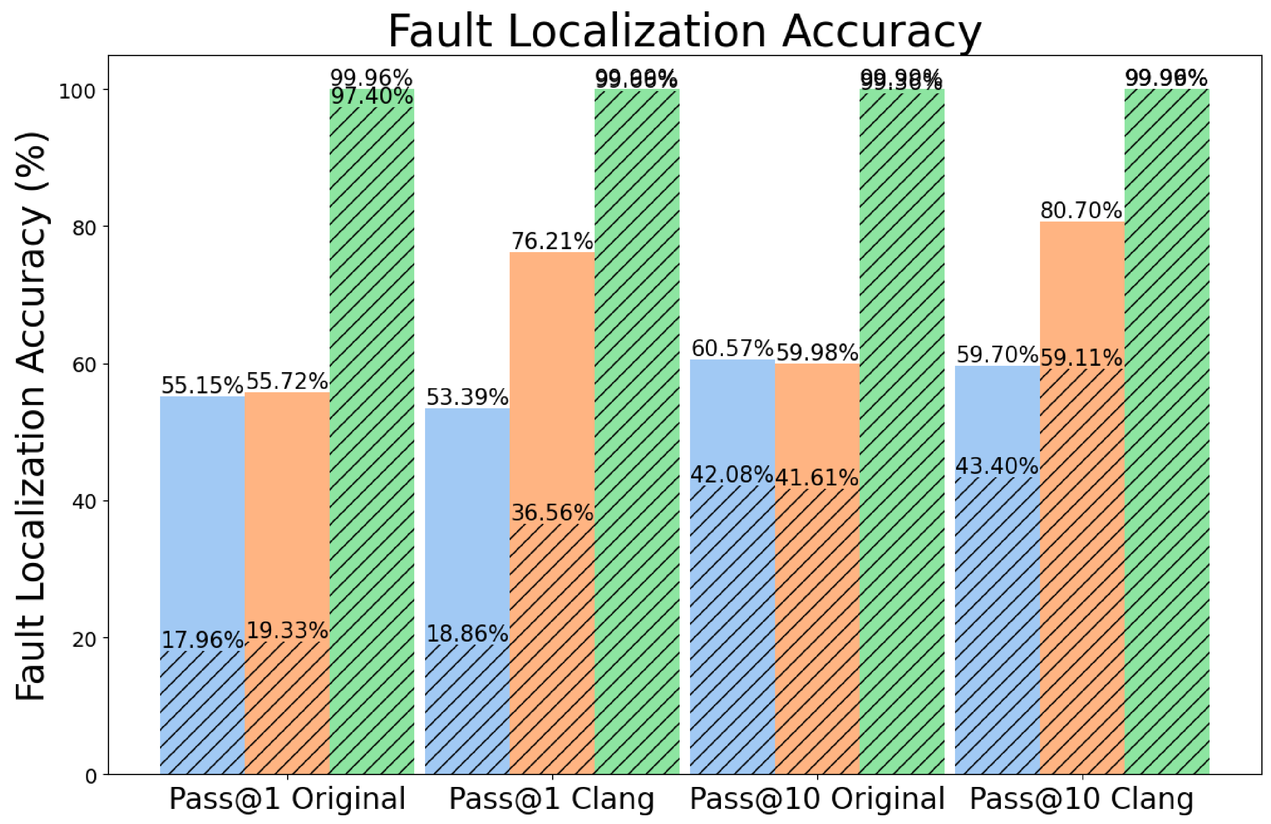
- Dataset Creation: **Curated pairs of C++ code snippets** with correct and incorrect versions, ensuring a minimal edit distance for focused learning.
- Dataset Formatting: Utilized **Clang Format** to refine the dataset, ensuring each line contains a single statement to **improve coverage accuracy**.
- Error Location Prediction with Coverage: Used **Gcov** to collect coverage data and preprocessed it by **assigning higher weights** to executed lines and their adjacent lines. Summarized coverage information was used to predict error lines for **30% of the total lines of code**, serving as hints to enhance APR performance..

Model Pipeline



Results





Perfect Repair: Modified code that passes all test cases.
Partial Repair: Modified code that passes more test cases than the original erroneous code but not all test cases.

Perfect Localization: Predicted error line exactly matches error location.
Partial Localization: Actual error location is included within the predicted range, but is not an exact match.

Conclusions

- **Effectiveness of Clang Format**: **Clang-formatting** significantly enhances the suspicious line prediction process with Coverage, enabling the CodeT5 model to learn logical error correction more accurately and precisely predict erroneous lines.
- **Coverage as a Boost to APR**: Traditional computer science debugging methods, particularly **Coverage**, can enhance APR by improving error localization and repair accuracy.

uos/poster cida.uos.ac.kr 113bommy@uos.ac.kr