

A Rule-based Self-healing Baseline for Hybrid AI Repair of Common Python Syntax Errors

Perera H.A.I.J. ^{1*} and Lekamge L.S. ²

¹Department of Software Engineering, Faculty of Computing,
Sabaragamuwa University of Sri Lanka, Sri Lanka

²Department of Computing and Information Systems, Faculty of Computing,
Sabaragamuwa University of Sri Lanka, Sri Lanka

*jayaruperera1998@gmail.com

Python has become a popular language of beginners due to its simplicity. However, syntax errors remain as a hurdle to productivity and learning. Although automated program repair (APR) has improved, in the majority of cases the existing solutions have difficulties in trading the deterministic accuracy of rule-based systems with the flexibility of AI-guided models. This paper deploys a rule-based repair agent in the form of a self-healing agent as the initial step in a proposed hybrid repair system, concerned with the automatic detection and correction of common Python syntax errors, including missing colons and indentation errors. The agent is modeled upon the MAPE-K (Monitor, Analyze, Plan, Execute, Knowledge) control loop and is tested on 37,639 real-world Python snippets with raw syntax errors that are mined on the public GitHub repositories. The agent based on rules is able to fix 14,149 snippets with a repair accuracy of 37.59% and an average fix time of around 0.0008 seconds. These results indicate that a rulebased layer can efficiently handle most low-complexity syntax errors with minimal delay, and represents a promising approach to rapid preprocessing. Meanwhile, the 62.41% of the unresolved cases also show the constraints of the use of static heuristics in context-sensitive or structurally complicated errors. This long abstract provides the empirical base to the rule-based stage and gives the impetus to the subsequent stage of the project, where a learning-based part will be added to create a hybrid repair architecture.

Keywords: *Python Syntax Errors, Self-healing Systems, Automated Program Repair, Hybrid AI, Rule-based Approaches*