1. Evaluating Representation Learning of Code Changes for Predicting Patch Correctness in Program Repair

Haoye Tian (1); Kui Liu (2); Kabore, A.K. (1); Koyuncu, A. (1); Li Li (3); Klein, J. (1); Bissyande, T.F. (1)

Source: *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, p 981-92, 2020; **ISBN-13:** 978-1-4503-6768-4; **DOI:** 10.1145/3324884.3416532; **Conference:** 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), 21-25 Sept. 2020, Melbourne, VIC, Australia; **Publisher:** IEEE Computer Society, Los Alamitos, CA, USA

Author affiliation: (1) University of Luxembourg, Luxembourg (2) Nanjing University of Aeronautics and Astronautics, China (3) Monash University, Melbourne, VIC, Australia

Abstract: A large body of the literature of automated program repair develops approaches where patches are generated to be validated against an oracle (e.g., a test suite). Because such an oracle can be imperfect, the generated patches, although validated by the oracle, may actually be incorrect. While the state of the art explore research directions that require dynamic information or that rely on manually-crafted heuristics, we study the benefit of learning code representations in order to learn deep features that may encode the properties of patch correctness. Our empirical work mainly investigates different representation learning approaches for code changes to derive embeddings that are amenable to similarity computations. We report on findings based on embeddings produced by pre-trained and re-trained neural networks. Experimental results demonstrate the potential of embeddings to empower learning algorithms in reasoning about patch correctness: a machine learning predictor with BERT transformer-based embeddings associated with logistic regression yielded an AUC value of about 0.8 in the prediction of patch correctness on a deduplicated dataset of 1000 labeled patches. Our investigations show that learned representations can lead to reasonable performance when comparing against the state-of-the-art, PATCH-SIM, which relies on dynamic information. These representations may further be complementary to features that were carefully (manually) engineered in the literature. (0 refs)

Inspec controlled terms: learning (artificial intelligence) - neural nets - program diagnostics - regression analysis - software maintenance

Uncontrolled terms: code changes - predicting patch correctness - automated program repair - oracle - generated patches - dynamic information - manually-crafted heuristics - code representations - representation learning approaches - learning algorithms - machine learning predictor - BERT transformer-based embeddings - labeled patches - learned representations - PATCH-SIM

Classification Code: C6150G Diagnostic, testing, debugging and evaluating systems - C1140V - C6264 - C6110B Software engineering techniques - C6130 Data handling techniques

IPC Code: G06F7/00 - G06F9/44 - G06F11/36 - G06N20/00

Treatment: Bibliography (BIB) - Practical (PRA) - Theoretical or Mathematical (THR)

Database: Inspec

Data Provider: Engineering Village