[ICML 2020] Graph-based, Self-Supervised Program Repair from Diagnostic Feedback. [paper]

Node/Graph Tasks: repairing programs based on diagnostic feedback

Training Type: pre-training and fine-tuning

Pretext task data: problem specific (program repairing, recovering the error index and repaired version)
The pretext task here is to train model beforehand to recover the error lines of the dataset generated by corrupting existing programs.

Initial short summary here
This paper proposes the DrRepair, a novel approach to program repair that addresses two problems, which are first system needs to connect and jointly reason over the broken source code and the diagnostic feedback. Second, existing works rely on manual effort to curate labeled datasets for program repair. Given a broken program with L lines $x = (x_1, ..., x_L)$ and diagnostic feedback provided by a compiler, $f = (i_{\text{err}}, m_{\text{err}})$ where i_{err} denotes the reported line number and m_{err} the error message (a sequence of tokens). The task is to identify the index of an erroneous line $k \in \{1, ..., L\}$ and generate a repaired version of the line y_k.

First a program-feedback graph to model the reasoning process is proposed. The intuition is that traditional previous seq2seq or AST-based models are difficult to capture long-range dependencies of tokens. To enable more efficient information flow, a program-feedback graph G that directly connects tokens relevant to the reasoning of program repair is required. A program-feedback graph $G = (V, E)$ has nodes V that consist of tokens in the diagnostic arguments, their occurrences in the source code, and all remaining identifiers in the code. Then identical tokens are connected in V with undirected edges E to capture the semantic correspondence. The resulting graph is a set of cliques, one for each symbol. Our program has an encoder that takes in a program x and feedback f, and a decoder that predicts a distribution over which line is erroneous k and a repaired line y_k. The encoder first encodes each input token at the line level and then utilizes the graph attention which propagates information across tokens on a program-feedback graph, and recontextualization which contextualizes token representations at the line level again to produce an embedding for each line. Finally, the decode outputs a distribution over the erroneous line index and a repaired line. A training example consists of a broken program x, feedback f, an erroneous line index k, and the repaired line y_k. The loss on a given example is the standard negative log-likelihood, $-\log p(k, y_k | x, f)$. The error localization and repair components are learned jointly.

In self-supervised learning, they utilize unlabeled, working programs to create a large amount of training data for program repair by automatically corrupting programs.

Bibtex:

@inproceedingsyasunaga2020graph, title=Graph-based, self-supervised program repair from diagnostic feedback, author=Yasunaga, Michihiro and Liang, Percy, booktitle=International Conference on Machine Learning, pages=10799–10808, year=2020, organization=PMLR