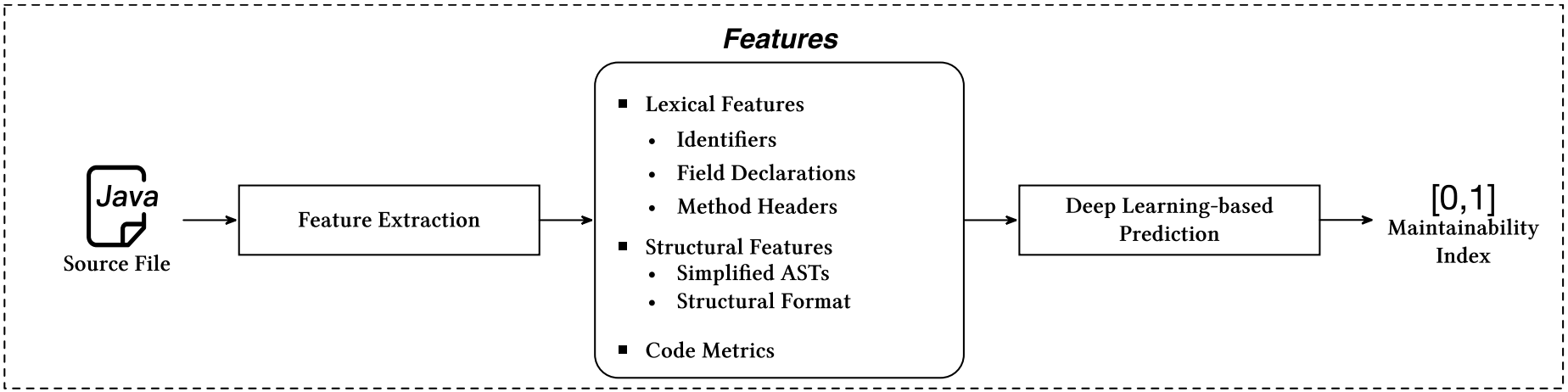
Measuring Code Maintainability with Deep Neural Networks

Yamin HU, Hao JIANG, Zongyao HU

Frontiers of Computer Science, DOI: 10.1007/s11704-022-2313-0

Problems & Ideas

- Problems of conventional approaches to measuring code maintainability :
  - Existing approaches rely heavily on code metrics, e.g., the number of Lines of Code and McCabe's Cyclomatic Complexity.
  - However, natural language in source code, especially identifiers, is rarely exploited by existing approaches.
- Ideas: The lexical semantics of text in source code should be exploited for code maintainability measurement.



An overview of the proposed approach (DeepM). DeepM takes a Java class as input and generates a maintainability index for the given class. DeepM is composed of two steps: feature extraction and deep learning-based prediction. The extracted features include lexical features, structural features, and traditional code metrics.

Main Contributions

- Contributions:
  - A deep learning-based approach to the quantitative measurement of code maintainability, which is the first approach that exploits the semantics of identifiers for code maintainability measurement;
  - A public implementation of the proposed approach and two publicly available datasets for the training and evaluation of code maintainability models;
  - An initial evaluation of the proposed approach.

Approach	Tomcat	Log4j 2	Apollo	Druid	Sentinel	Average
DeepM	**90.0%**	**80.0%**	**90.0%**	**85.0%**	**92.5%**	**87.5%**
LRR	80.0%	70.0%	77.5%	75.0%	72.5%	75.0%
CHC	65.0%	55.0%	70.0%	45.0%	52.5%	57.5%

The accuracies of the proposed approach (DeepM) and the two baseline approaches (LRR and CHC). The evaluation results suggest that the proposed approach outperforms the state-of-the-art approaches, improving the accuracy of the results from 75.0% and 57.5% to 87.5%.