学 习 进 展 & 学 期 计 划

2021 本周工作汇报

汇报人：王昭丹

5 Papers

Title	Publication source	Year
Improving Bug Detection and Fixing via Code Representation Learning	ICSE	2020
An Empirical Study of Fault Localization Families and Their Combinations	TSE	2021
A Developer Centered Bug Prediction Model	TSE	2018
Just-In-Time Defect Identification and Localization: A Two-Phase Framework	TSE	2020
Chaff from the Wheat: Characterizing and Determining Valid Bug Reports	TSE	2020

Improving Bug Detection and Fixing via Code Representation Learning

A deep learning framework to improve the software quality and reliability on these two detect-fix processes.

Used **advanced code modeling** and **AI models**.

Aim: to improve the existing state-of-the-art bug detection and auto-fixing (namely detect-fix) approaches via accurate, effective, and specialized code representation learning.

The **code representation learning** relies on the following pillars: code representations (i.e., data structures) obtained from advanced program analysis and deep neural network models.

A Developer Centered Bug Prediction Model

Fig. 1. Example of two developers having different levels of “scattering”

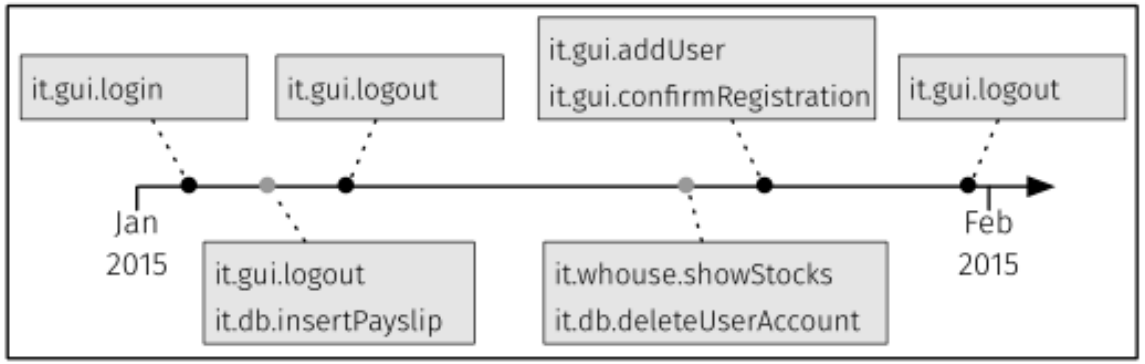
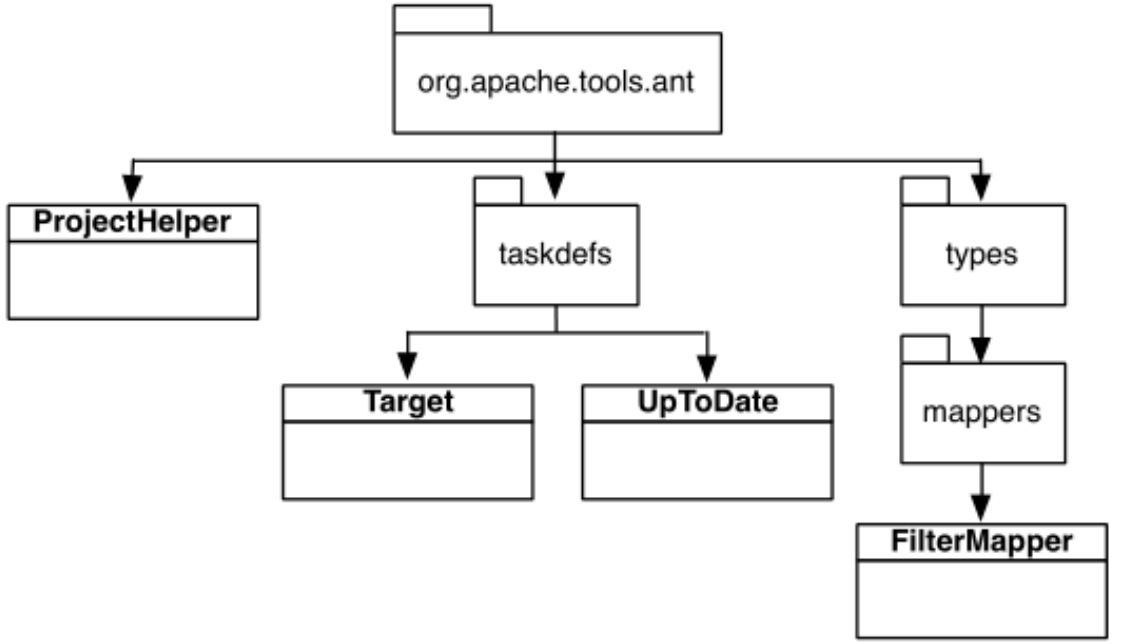


Fig. 2. Example of structural scattering



An Empirical Study of Fault Localization Families and Their Combinations

This paper reports on an empirical study of fault localization techniques.

Fault localization techniques from different families

- **SBFL**
- **MBFL**
- **program slicing**
- **predicate switching**
- **stack trace analysis**
- **information retrieve-based fault localization**
- **history-based fault localization**

Based on 357 real-world faults from the Defects4J dataset.

Just-In-Time Defect Identification and Localization: A Two-Phase Framework

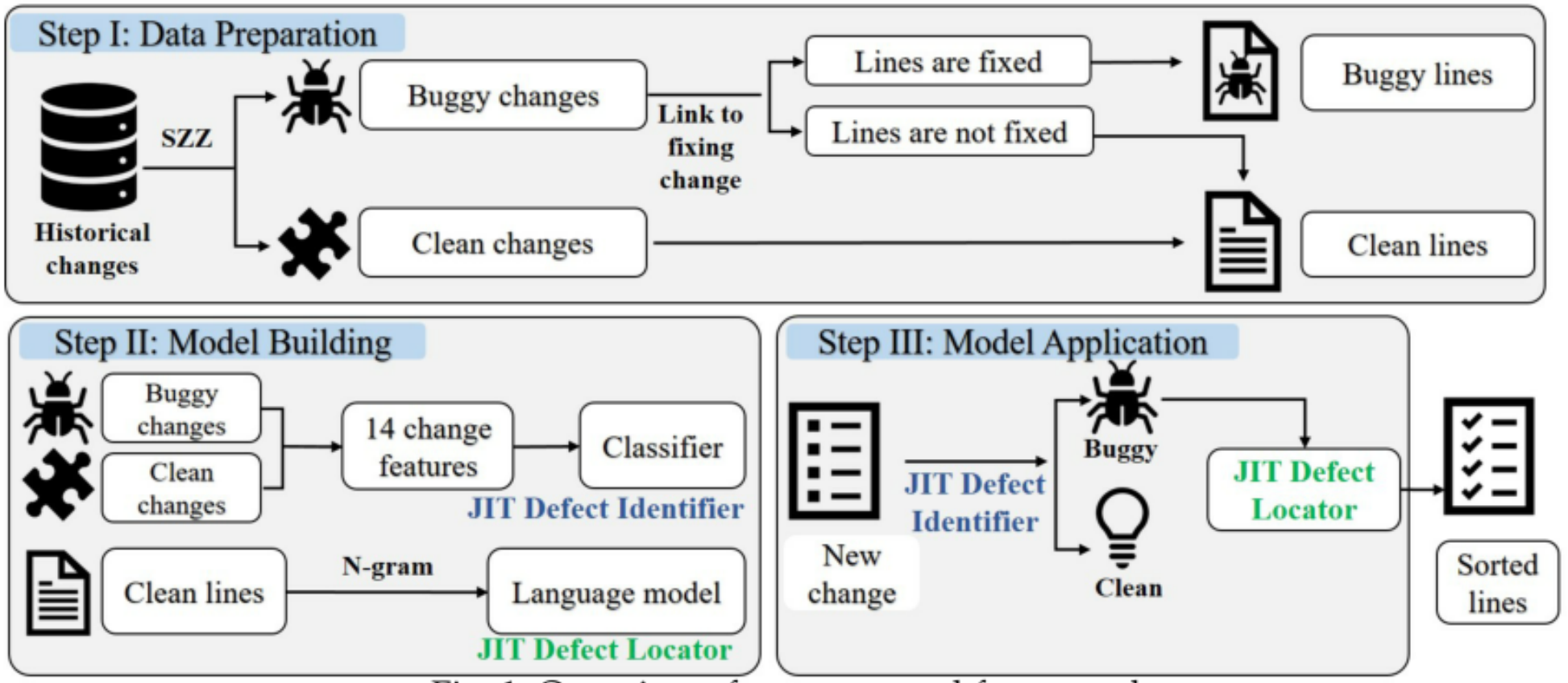


Fig. 1: Overview of our proposed framework

Chaff from the Wheat: Characterizing and Determining Valid Bug Reports

Dimension	Feature Name	Description
Reporter Experience	bug-num	Number of prior bug reports submitted by the reporter of this bug report
	recent-bug-num	Number of prior bug reports submitted by the reporter of this bug report in 90 days
	valid-rate	Valid rate of prior bug reports with known labels submitted by the reporter of this bug report
Collaboration Network	lcc-membership	These metrics are used to quantify a bug reporter's degree of activity in his/her bug handling community [99]
	in-degree	
	out-degree	
	total-degree	
	clustering-coefficient	
	k-coreness	
	closeness-centrality	
	betweenness-centrality	
	eigenvector-centrality	
Completeness	has-stack	Whether description of this bug report contains stack traces
	has-step	Whether description of this bug report contains steps to reproduce the bug
	has-code	Whether description of this bug report contains code examples
	has-patch	Whether description of this bug report contains patches
	has-testcase	Whether description of this bug report contains test cases
	has-screenshot	Whether description of this bug report contains screenshots
Readability	flesch	These metrics are measured by the number of syllables per word and the length of sentences, which are used to quantify the readability of a text [3], [22], [25], [30], [45], [53], [69]
	fog	
	lix	
	kincaid	
	ari	
	coleman-liau	
	smog	
Text	summary-nb-score	Likelihood scores to be valid of this bug report calculated based on its summary: *summary-nb-score*, *summary-mnb-score*, *summary-dmnb-score* and *summary-cnb-score* are output by the naive Bayes, multinomial naive Bayes , discriminative multinomial naive Bayes and complement naive Bayes classifiers that are learned using summary of bug reports, respectively
	summary-mnb-score	
	summary-dmnb-score	
	summary-cnb-score	
	desc-nb-score	Likelihood scores to be valid of this bug report calculated based on its description: *desc-nb-score*, *desc-mnb-score*, *desc-dmnb-score* and *desc-cnb-score* are output by the naive Bayes, multinomial naive Bayes , discriminative multinomial naive Bayes and complement naive Bayes classifiers that are learned using description of bug reports, respectively
	desc-mnb-score	
	desc-dmnb-score	
	desc-cnb-score	

学期计划

11.8 -11.21

- 整理可用的机器学习模型构建方法（e.g. CNN / DNN)

12.13 – 12.19

- 确定最后使用何种模型或方法

Start 1 2 3 4 5 Finish

10.18 - 11.7

- 每周整理三种特征（包括实现方法以及复现可能存在的问题）

11.22 – 12.12

- 每周复现跑通一个技术
  - Buglocator
  - Blizzard
  - LR

12.20 – 12.26

- 串通整个实验步骤

学 习 进 展 & 暑 期 计 划

感 谢 您 的 聆 听

汇报人：王昭丹