ISCAS 中国科学院软件研究所学术年会'2022
暨计算机科学国家重点实验室开放周

学术论文

Are We Building on the Rock? On the Importance of Data Preprocessing for Code Summarization
论数据质量在代码注释自动生成中的重要性

石琳 沐方文 陈啸 王松 王俊杰 杨叶 李戈 夏鑫 王青

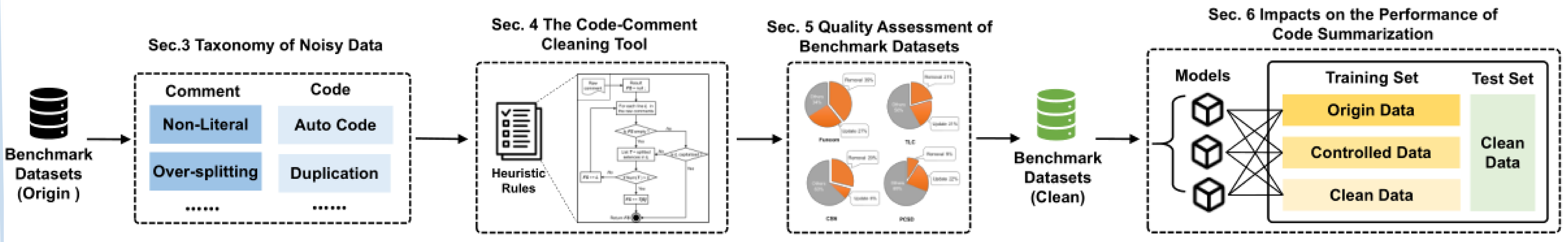
the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022)

主要联系人：石琳（15001193593，shilin@iscas.ac.cn）

Introduction

- Code summarization models require large-scale and high-quality training datasets. To that end, multiple benchmark datasets for code summarization tasks have been constructed. Although these datasets are expected to be of good quality, noise is inevitable due to the differences in coding conventions and assumptions employed in modern programming languages and IDEs.
- To investigate the aforementioned concerns of data quality for code summarization, we conduct a systematic study to assess and improve the quality of four widely-used benchmark datasets.

Methodology



Noisy Data Taxonomy

Comment-Related Noisy Data		Code-Related Noisy Data	
Partial Sentence	`/* Returns the high-value * for an item within a series. */`	Empty Function	`/*Specifies the behaviour of the automaton in its end state*/ protected void end(){}`
Verbose Sentence	`Generate a CSV file containing a summary of the xBlock usage Arguments:course_data`	Commented-Out	`/* for now try mappig full type URI */ // public String transformTypeID(URI typeuri){ // return typeuri.toString();}`
Content Tampering	`<p> Builds the JASPIC application context.</p>`		
Over-Spliting	`/* This method initializes jTextField. */`	Block-Comment	`/* Get GPS Quality Data */ public int getFixQuality(){ checkRefresh(); // TODO: Why is he using Math.round?`
Non-Literal	`/* 将JSONArray转换为Bean的List，默认为ArrayList`		
Interrogation	`/* Do we need to show the upgrade wizard prompt? public boolean isDue() {`	Auto Code	`/* Test the constructor */ public void testConstructor() {`
Under-Developmen	`/* Description of the Method */ protected void openFile(File f) {`	Duplicated Code	Developers often reuse code by copying, pasting and modifying to speed up software development

Effectiveness Evaluation

	Category	Dataset: #Annotations (100%)	Dataset: Rule-Build (80%)	Dataset: Rule-Test (20%)	Performance (%): P	Performance (%): R	Performance (%): F1
Comment	Partial Sentence	176	135	41	97.5	95.1	96.3
	Verbose Sentence	129	111	18	94.7	100.0	97.3
	Content Tampering	147	120	27	92.9	96.3	94.6
	Over-Splitting	84	63	21	90.9	95.2	93.0
	Non-Literal	38	30	8	100.0	100.0	100.0
	Interrogation	16	7	9	100.0	88.9	94.1
	Under-Development	57	92	57	91.5	94.7	93.1
	Total	647	558	181	95.4	95.8	95.5
Code	Empty Function	21	14	7	100.0	100.0	100.0
	Commented-Out Method	4	2	2	100.0	100.0	100.0
	Block-Comment Code	44	31	13	100.0	92.3	96.0
	Auto Code	179	133	46	97.7	93.5	95.6
	Duplicated Code	22	16	6	100.0	100.0	100.0
	Total	270	196	74	99.6	97.2	98.3

Our code-comment cleaning tool can accurately filter noisy data, with all the F1 scores of over 90.0%

Distribution of noisy data

	Category of Noisy Data	Funcom (%)	TLC (%)	CSN (%)	PCSD (%)
	Total	65.8	41.9	37.2	31.2
Comment	Partial Sentence	17.1	0.0	7.8	15.9
	Verbose Sentence	0.0	22.8	0.0	7.8
	Content Tampering	9.7	3.2	24.4	0.5
	Over-Splitting	24.1	0.0	0.0	0.0
	Non-Literal	0.5	0.0	7.8	0.2
	Interrogation	0.7	0.9	0.7	0.3
	Under-Development	3.7	1.2	1.2	2.3
	Total	40.9	25.4	36.1	26.5
Code	Empty Function	1.6	1.1	0.0	0.0
	Commented-Out Method	0.2	0.0	0.0	0.0
	Block-Comment Code	11.1	0.0	0.0	0.0
	Auto Code	29.8	4.6	1.6	4.3
	Duplicated Code	0.6	18.4	0.0	1.5
	Total	40.7	22.6	1.6	5.8
	Removed noisy data	38.7	21.1	29.2	9.3
	Updated noisy data	27.1	20.8	8.0	21.9

Finding 1: Noisy data extensively exist in the four benchmark datasets, ranging from 31.2% to 65.8%.

Impacts on the performance of models

Benchmark	Model	Train set	BLEU-1	BLEU-2	BLEU-3	BLEU-4	ROUGE	METEOR
Funcom	NNGen	Origin	23.87	14.28	11.4	10.05	26.88	12.59
		Controlled	21.93	12.19	9.34	8.09	24.84	11.39
		Filtered	**24.58** 3.0% ↑	**15.26** 6.9% ↑	**12.49** 9.6% ↑	**11.2** 10.3% ↑	**27.08** 0.7% ↑	**13.24** 5.2% ↑
	NCS	Origin	29.95	17.79	10.2	6.42	34.84	16.14
		Controlled	29.33	17.06	9.78	5.31	34.05	15.65
		Filtered	**30.53** 1.9% ↑	**18.79** 5.6% ↑	**11.47** 12.5% ↑	**7.64** 16.0% ↑	**35.42** 1.7% ↑	**16.32** 1.1% ↑
	Rencos	Origin	27.23	15.97	9.62	6.43	31.97	14.32
		Controlled	26.90	15.71	9.48	6.42	31.79	14.16
		Filtered	**27.92** 2.5% ↑	**16.8** 5.2% ↑	**10.61** 10.3% ↑	**7.44** 13.6% ↑	**32.51** 1.7% ↑	**14.50** 1.3% ↑
TLC	NNGen	Origin	32.58	24.16	21.92	20.74	36.07	18.14
		Controlled	39.84	32.01	29.24	27.51	43.57	23.22
		Filtered	**46.88** 43.9% ↑	**39.27** 62.5% ↑	**36.81** 67.9% ↑	**35.19** 41.1% ↑	**49.08** 36.1% ↑	**25.53** 40.7% ↑
	NCS	Origin	42.09	32.95	29.09	27.09	46.30	24.18
		Controlled	39.28	29.61	25.83	23.89	43.49	22.11
		Filtered	**46.52** 10.5% ↑	**37.19** 12.9% ↑	**33.41** 14.9% ↑	**31.38** 13.7% ↑	**49.40** 6.7% ↑	**24.67** 2.0% ↑
	Rencos	Origin	43.66	34.82	31.29	29.19	47.87	24.95
		Controlled	43.71	34.89	31.21	28.93	47.85	25.37
		Filtered	**51.54** 18.0% ↑	**42.90** 23.2% ↑	**39.22** 25.3% ↑	**37.00** 21.1% ↑	**54.25** 13.3% ↑	**28.21** 13.1% ↑
CSN	NNGen	Origin	14.86	6.08	4.07	3.42	18.04	8.54
		Controlled	13.95	5.09	3.21	2.62	17.08	7.97
		Filtered	**19.89** 33.8% ↑	**8.28** 36.2% ↑	**5.72** 40.5% ↑	**4.96** 31.0% ↑	**23.17** 28.4% ↑	**9.67** 13.2% ↑
	NCS	Origin	25.47	12.34	5.81	3.02	30.47	12.48
		Controlled	25.45	12.29	5.68	2.88	31.17	12.30
		Filtered	**28.68** 12.6% ↑	**14.01** 13.5% ↑	**6.96** 19.8% ↑	**3.87** 22.0% ↑	**34.29** 12.5% ↑	**13.84** 10.9% ↑
	Rencos	Origin	16.99	7.65	4.09	2.64	20.91	8.33
		Controlled	16.30	7.09	3.75	2.43	20.00	8.13
		Filtered	**24.72** 45.5% ↑	**11.36** 48.5% ↑	**6.51** 59.2% ↑	**4.56** 42.1% ↑	**29.35** 40.4% ↑	**11.52** 38.3% ↑
PCSD	NNGen	Origin	22.52	15.48	12.63	10.45	24.90	12.97
		Controlled	21.81	14.77	11.99	9.91	24.16	12.49
		Filtered	**25.96** 15.3% ↑	**18.91** 22.2% ↑	**16.27** 28.8% ↑	**14.00** 25.4% ↑	**27.68** 11.2% ↑	**15.09** 16.3% ↑
	NCS	Origin	28.14	18.69	14.28	11.36	32.95	16.30
		Controlled	26.85	17.42	13.05	10.17	31.77	15.42
		Filtered	**37.33** 32.7% ↑	**24.74** 32.4% ↑	**19.49** 36.5% ↑	**16.48** 31.1% ↑	**40.93** 24.2% ↑	**18.67** 14.5% ↑
	Rencos	Origin	30.37	21.27	16.42	12.93	33.66	17.40
		Controlled	29.73	20.55	15.71	12.37	33.05	16.96
		Filtered	**33.59** 10.6% ↑	**24.14** 13.5% ↑	**19.63** 19.5% ↑	**16.10** 19.7% ↑	**36.15** 7.4% ↑	**19.18** 10.2% ↑

Finding 2: Removing noisy data from the training set in the four datasets has a positive influence on the performance of the models (improving BLEU-4 by 21%-27%).

Conclusion

- We propose a taxonomy of data preprocessing noises in four popularly used benchmark datasets for code summarization, which contains 12 different types of noise.
- We develop an automated data cleaning tool for code summarization datasets, which can help distill high-quality code-comment data.
- We perform a comprehensive assessment on data quality of datasets, which provides practical insights for future research.
- We conduct a comparative analysis on the performance of code summarization models, our results show that removing noises yields significant model perfomance improvement.