

Clone Detection in Test Code: An Empirical Evaluation

Brent van Bladel, Serge Demeyer

*2020 IEEE 27th International Conference on
Software Analysis, Evolution and
Reengineering (SANER), 2020, 492-500*

背景

- 単体テストにおいてもコードクローンが頻繁に作成される
- 単体テストにおけるクローン作成は冗長性を増長させる要因となる[8]
- 単体テストのクローンに着目した研究は存在していない

[8] V. Garousi and B. Kucuk, "Smells in software test code: A survey of knowledge in industry and academia," *Journal of Systems and Software*, vol. 138, pp. 52 – 81, 2018.

目的

- 単体テストにおけるクローンの特徴を明らかにする
- 既存のクローン検出手法で単体テストのクローンを十分に検出できるかを明らかにする

リサーチクエスチョン

1. 単体テストにおけるクローンの特徴は何か
2. 単体テストで作成されるクローンに対して既存手法はどれほどの検出能力を発揮するか

データセット

- 扱うクローンタイプは1-4まで
 - タイプ4クローン：振る舞いが同じコード片のペア
 - タイプ3かつタイプ4はタイプ4クローンとみなす

TABLE I
DATASET DESCRIPTIVE STATISTICS.

Name	Files	Tests	LOC
Apache Common's Math	360	2,782	32,483
Google Guave	109	1,229	10,929
Java Design Patterns	133	313	3,747

This table only lists the JUnit test code of the respective projects.

使用するクローン検出手法

- NiCad[24]

- 字句ベースで検出，識別子を特殊文字化後最長共通シーケンスアルゴリズムを使用してタイプ3クローンの検出

- CPD[14]

- 字句ベースで検出，ラビン-カープアルゴリズムを使用して検出

- iClones[26]

- 字句，ASTのハイブリッドな手法で検出，AST変換後字句にシリアル化し，接尾辞木検出アルゴリズムを使用

- TCORE[28]

- AST，プログラムグラフのハイブリッドな手法で検出，AST変換後シンボリック実行でPDG化し部分木マッチングアルゴリズムを使用

[24] C. K. Roy and J. R. Cordy, "Nicad: Accurate detection of near-miss intentional clones using flexible pretty-printing and code normalization," in *2008 16th IEEE international conference on program comprehension*. IEEE, 2008, pp. 172–181.

[14] C. K. Roy, J. R. Cordy, and R. Koschke, "Comparison and evaluation of code clone detection techniques and tools: A qualitative approach," *Science of computer programming*, vol. 74, no. 7, pp. 470–495, 2009.

[26] R. Koschke, R. Falke, and P. Frenzel, "Clone detection using abstract syntax suffix trees," in *2006 13th Working Conference on Reverse Engineering*. IEEE, 2006, pp. 253–262.

[28] B. van Bladel and S. Demeyer, "A novel approach for detecting type-iv clones in test code," in *2019 IEEE 13th International Workshop on Software Clones (IWSC)*. IEEE, 2019, pp. 8–12.

RQ1のアプローチ

- クローン密度を計算

- $clone\ density = \frac{f_c * 100}{f_{tot}}$ f_c : クローン片数 f_{tot} : テストコード総数

- タイプ毎のクローン密度やクローンクラスの大きさを調査

RQ1の結果

- タイプ2のクローン密度が他に比べて大きい
 - 単体テストの典型的な構造によると考えられる
- クローンクラスに含まれるクローン片が多く，クローンペア数がクローン片数を上回ることがあった
 - 1つのテストが複数回複製され，軽微な変更が加えられる

TABLE II
OVERVIEW OF DATASETS.

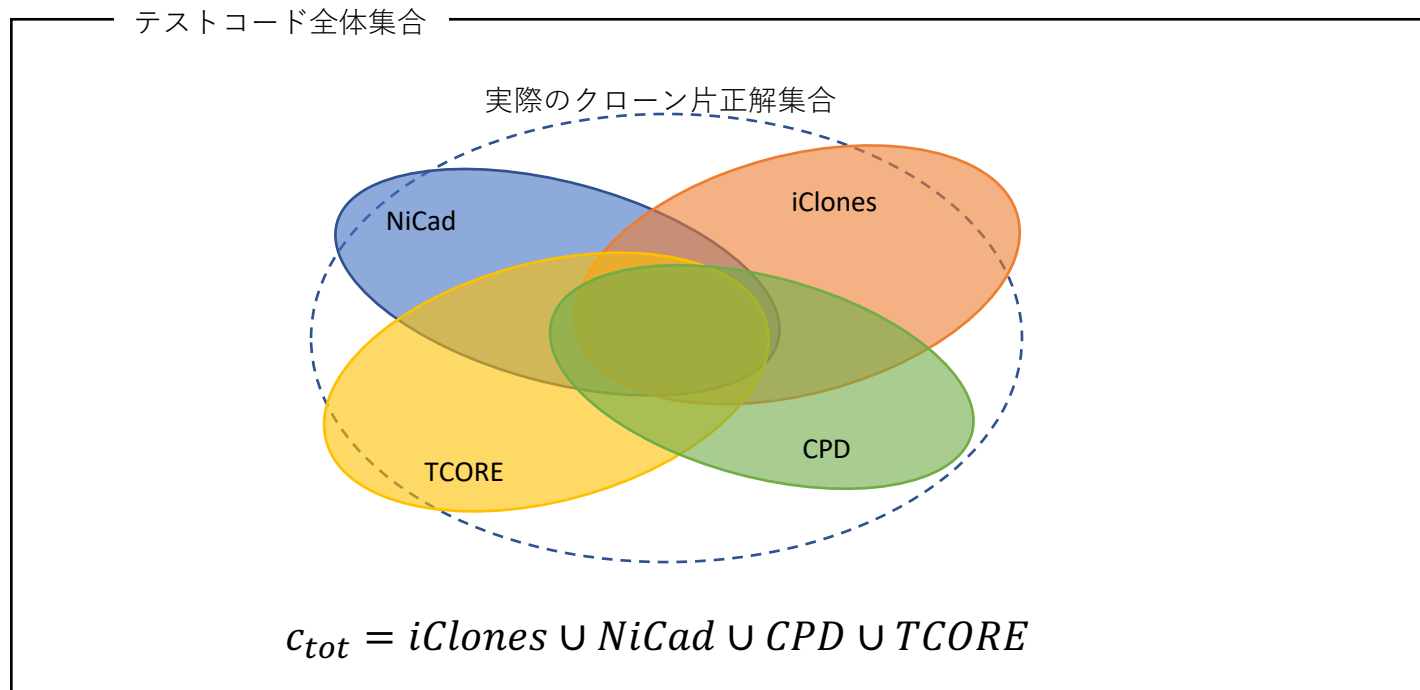
Clone Type	Clone Pairs			Clone Fragments (Clone Density)		
	Apache	Google	Patterns	Apache	Google	Patterns
Type 1	48	24	4	55 (1.9%)	40 (3.2%)	5 (1.5%)
Type 2	708	310	36	373 (13.4%)	181 (14.7%)	40 (12.7%)
Type 3	954	32	13	214 (7.6%)	25 (2.0%)	15 (4.7%)
Type 4	354	40	21	170 (6.1%)	40 (3.2%)	20 (6.3%)
Total	2064	406	74	812 (29.1%)	286 (23.2%)	80 (25.5%)

RQ2のアプローチ

- 各手法に対して以下の式で求められる適合率と再現率を計算
- 適合率： $precision = \frac{c_{TP} * 100}{c_{all}}$
 - c_{all} : 評価対象手法が検出したクローン
 - c_{TP} : 検出したクローンのうち正解と認められたもの
- 再現率： $recall = \frac{c_{all} * 100}{c_{tot}}$
 - c_{tot} : 全手法で検出したクローン

再現率の計算について

- クローンの正解集合を全て見つけ出すのは現実的でないため、全手法で検出したクローンを擬似的な正解集合としている



RQ2の結果

- 適合率は全ての手法において高い水準である
 - クローン検出手法は適合率が高くなるように設計されている
- 再現率はデータセットや手法によっては低い水準である
 - 重複テストの除去が不完全になる

TABLE III
OVERVIEW OF CLONE DETECTOR PERFORMANCE.

Tool	Precision			Relative Recall		
	Apache	Google	Patterns	Apache	Google	Patterns
NiCad	100%	99%	100%	14.1%	27.8%	22.9%
CPD	100%	100%	100%	11.1%	30.5%	44.5%
iClones	98%	99%	100%	86.5%	47.5%	29.7%
TCORE	97%	95%	100%	30.8%	50.2%	36.4%

RQ2の結果

- ASTベースの手法の方が多くのクローンを検出できている
- 手法によっては高い再現率で検出できるクローンタイプがある

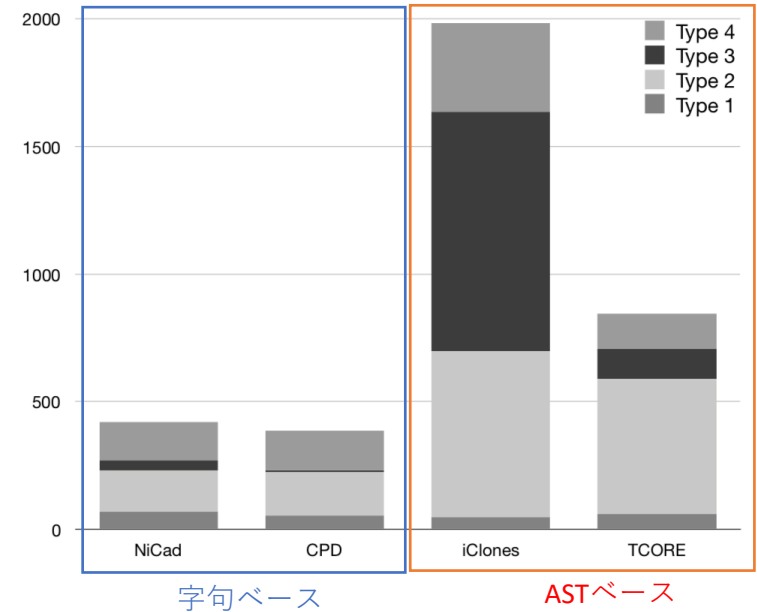


Fig. 4. Clone pairs detected by each tool per type.

TABLE IV
CLONE PAIR CLASSIFICATION AND RELATIVE RECALL PER CLONE DETECTOR.

Clone Type	Clone Pairs (Distribution)				Relative Recall			
	NiCad	CPD	iClones	TCORE	NiCad	CPD	iClones	TCORE
Type 1	69 (16.3%)	55 (14.1%)	48 (2.3%)	59 (6.7%)	90.7%	72.3%	63.1%	77.6%
Type 2	162 (38.3%)	171 (44.0%)	651 (32.5%)	529 (60.9%)	15.3%	16.2%	61.7%	50.1%
Type 3	41 (9.7%)	4 (1.0%)	935 (46.7%)	120 (13.8%)	4.1%	0.4%	93.5%	12.0%
Type 4	149 (35.3%)	158 (40.7%)	348 (17.3%)	136 (15.6%)	35.9%	38.0%	83.8%	32.7%
Total	422 (100%)	388 (100%)	2001 (100%)	868 (100%)	16.5%	15.2%	78.6%	34.1%

まとめ

- 単体テストにおけるクローンの特徴を分析
- タイプ2クローンが多いことが明らかになった
- 手法毎に正確に検出できるタイプのクローンがあることが明らかになった

所感

- プロダクトコードとは異なる特徴が明らかになった点が興味深かった
- 他の手法も追加して実験すると結果はどうなるのか気になった
- テストコードに特化したクローン検出手法が必要なのではないかと思った