

技術的負債に関連する 課題票抽出手法の構築

オープンソースソフトウェア工学研究室

学籍番号：60210032

氏名：木村 祐太

技術的負債：Technical Debt^[1]

- 短期的には有用，長期的には問題がある実装や設計
 - 例) レガシーコード : かなり前に実装されたコード
 - コードスメル : 深刻な問題を抱えているコードの特徴
- 保守コストの増加を引き起こす^{[2][3][4][5]}
 - 開発の約23～35%が余分な作業
 - 既存のTDによる新たなTDの導入 (TDの蓄積)

開発資源（人的，金銭的，時間的）は有限
TDの蓄積量を管理する必要がある

[1] Cunningham, W., 1992. The wycash portfolio management system. Proc. Obj. Oriented Programm. Syst. Lang. Appl. 4 (2), pp. 29-30.

[2] C. Seaman, G. Yuepu, N. Zazworka, F. Shull, C. Izurieta, C. Yuanfang, et al., "Using technical debt data in decision making: Potential decision approaches," in Managing Technical Debt (MTD), 2012 Third International Workshop on, 2012, pp. 45-48.

[3] E. Tom, A. Aurum, and R. Vidgen, "An exploration of technical debt," Journal of Systems and Software, vol. 86, no. 6, 2013, pp. 1498-1516.

[4] T. Besker, A. Martini and J. Bosch, "Technical Debt Cripples Software Developer Productivity: A Longitudinal Study on Developers' Daily Software Development Work," 2018 IEEE/ACM International Conference on Technical Debt (TechDebt), 2018, pp. 105-114.

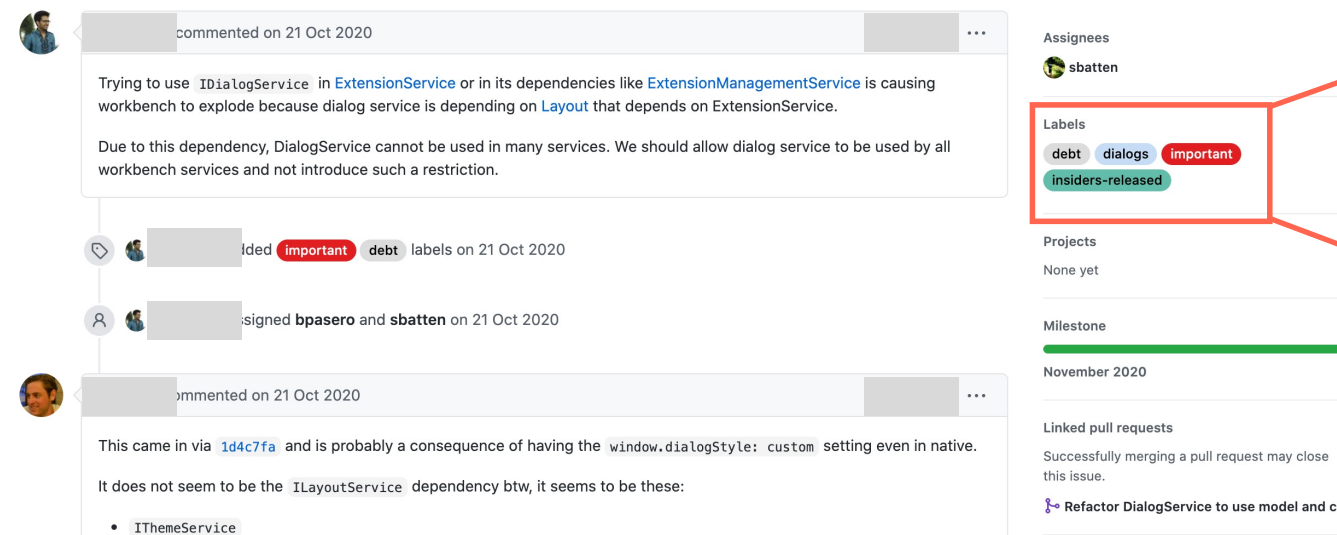
[5] T. Besker, A. Martini, and J. Bosch, "The pricey Bill of Technical Debt: When and by whom will it be paid?," in IEEE International Conference on Software Maintenance and Evolution (ICSME), Shanghai, China, 2017, pp. 13-23.

Self-Admitted TD

- 開発者によって明示的に管理されるケース
 - SATD-Issue [6]
 - 課題票 (Issue) にTDラベルを割り当てて明示されるSATD

Cannot use IDialogService in some services due to custom dialog service dependencies
#109056

 Closed sandy081 opened this issue on 21 Oct 2020 · 13 comments



The screenshot shows a GitHub issue page for #109056. The issue title is "Cannot use IDialogService in some services due to custom dialog service dependencies". The issue is closed and was opened by sandy081 on 21 Oct 2020. There are 13 comments. The first comment, from sandy081, describes a problem with IDialogService and its dependencies. The second comment, from bpassero, mentions that the issue is related to a custom dialog service dependency. The third comment, from sbatten, mentions that the issue is related to a custom dialog service dependency. The right sidebar shows the issue's metadata, including assignees (sbatten), labels (debt, dialogs, important, insiders-released), projects (None yet), milestone (November 2020), and linked pull requests (Refactor DialogService to use model and c...).



“debt”というTDラベルが
割り当てられている

引用 : <https://github.com/microsoft/vscode/issues/109056>

[6] Laerte Xavier, Fabio Ferreira, Rodrigo Brito, and Marco Tulio Valente. Beyond the Code: Mining Self-Admitted Technical Debt in Issue Tracker Systems. In Proceedings of the 17th International Conference on Mining Software Repositories (MSR '20). 2020, pp. 137–146.

SATD-Issue研究の問題点

- 調査における一般性の欠如
 - SATD-Issueを扱っているプロジェクトが少ないためデータの絶対数が少ない



本研究の目的とアプローチ

目的

- TD-Issueを自動抽出しデータを増やす

※ TD-Issue : TDに関連するIssue

アプローチ

- TD-Issueを抽出する分類モデルの構築
 1. モデル構築に利用する特徴量の選定
 2. 分類モデルの評価実験
 3. 他プロジェクトを利用した分類モデルの一般性の検証

データセット

- SATD-Issueを慣習的に扱っている4プロジェクトをGitHubから抽出
 - VSCode, Influxdb, Saleor, Server
- データ数が最も多かったプロジェクトをモデル構築に利用
 - 課題内容が確定しているClosed Issue（解決済み）を対象

プロジェクト	SATD-Issue [件]（正例）	通常のIssue [件]（負例）
vscode	1,612	102,301

- 残りの3プロジェクトは、今後の一般性の検証に用いる予定

特徴量の検討

- Closed Issueから取得できる特徴量を抽出
 - TD-Issue分類に関する研究が存在しないため、有効な特徴量が不明
- Issue発見から解決までの工程における構成要素に着目



Issue発見から解決までの工程

特徴量の抽出 (1/4)



- 報告者
 - Issueを発見・報告する開発者

(仮説) 開発経験が豊富な開発者ほどTDを認識している

SATD-Code研究 [7]で, SATDコメントはベテラン開発者に使われる傾向にある

→ 報告者のこれまでの開発経験

- 経験年数, Issue報告数, コミット数 (変更追加回数)
- 特徴量次元数: 3次元

特徴量の抽出 (2/4)



- テキスト
 - Issueに記述されるテキスト

(仮説) 報告されるIssue内容が異なる

→ Issue内容を記述した自然言語 (英語)

- Issueタイトル, 本文
- ~~特徴量次元数: 46989次元~~

特徴量の抽出 (3/4)



- プロセス
 - Issueの報告から解決までにおこなわれる活動情報

(仮説) 通常のIssueに比べて解決にコストがかかる

SATD-Issueの方が解決されるまでに時間がかかる [6]

→ Issue解決に至るまでの活動に伴う情報

- 本文の文字数, 開発者数, 解決までにかかる時間, など
- ~~特徴量次元数: 13次元~~

[6] Laerte Xavier, Fabio Ferreira, Rodrigo Brito, and Marco Tulio Valente. Beyond the Code: Mining Self-Admitted Technical Debt in Issue Tracker Systems. In Proceedings of the 17th International Conference on Mining Software Repositories (MSR '20). 2020, pp. 137–146.

特徴量の抽出 (4/4)



- ソースコード
 - Issue解決のためにおこなわれるソースコードの変更

(仮説) 主にリファクタリングがおこなわれる

既存のTD検出ツールはソースコードベースでおこなわれる [8]

→ 変更によるソースコードメトリクスの変化

- コード行数の増減, サイクロマティック複雑度の増減, など
- 特徴量次元数: 20次元

[8] Z. Li, P. Avgeriou, and P. Liang, "A systematic mapping study on technical debt and its management," Journal of Systems and Software, vol. 101, 2015, pp. 193-220.

特徴量の選択

- 報告者・プロセス・ソースコード
 - 仮説検定と効果量測定による特徴量選択
 - 量的データ
 - 仮説検定：マンホイットニーのU検定
 - 効果量測定：Cliff's d
 - 質的データ
 - 仮説検定：カイ二乗検定
 - 効果量測定：Cohen's w
 - VIF統計量による特徴量削減

有意差 ($p < 0.01$) があり効果量が小程度以上かつ
VIF統計量が3未満の特徴量を選択

特徴量一覧

Category	Metrics	Description
報告者 (3次元)	Experience	報告者がアカウントを登録してから課題票Iを報告するまでの時間
	OpenIssueNum	報告者が課題票Iを報告するまでにおこなった課題報告の数
	Type	報告者の種別 (1：プロジェクトメンバー, 2：コントリビュータ, 0：その他)
プロセス (6次元)	DescriptionLen	課題票の本文に含まれる単語の数
	CodeLink	課題票の本文に含まれるコード片のリンクの有無
	CommentNum	課題票のコメントの数
	SelfAssign	報告者が自身を課題票の担当に割り当てているか
	pCommitNum	課題票解決のためにおこなわれたコミットの数
	ResolutionTime	課題票解決にかかった時間
ソースコード (6次元)	CountDeclFunctionAdd	変更によって増加した関数の数
	CountDeclFunctionDel	変更によって減少した関数の数
	CodeChurnDel	変更時に削除したコードの総行数
	CountLineCommentDel	変更によって減少したコメントの行数規模
	EssentialAdd	変更によって増加したEssential複雑度
	EssentialDel	変更によって減少したEssential複雑度

テキストの前処理

- 数値（ベクトル）に変換するため以下の処理を行う
 - 1-gram（単語）に分割
 - ストップワードなどノイズとなる単語の除去
 - 品詞の限定（名詞，動詞，形容詞，副詞）
 - 固有名詞は除く
 - 単語の正規化（小文字統一，ステミング）
 - BoW（Bag of Words）の用いたベクトル化
 - ~~ただし，出現文書回数が20回以上の単語~~
 - ~~※ 20回をしきい値にするとおおよそ出現文書回数上位20%の単語が取得できる~~
 - ~~※ 出現文書回数でフィルタリングする理由は過学習防止のため~~

評価実験：概要

- データセット

- 外れ値を含むデータは除外

TD-Issue（正例）	通常のIssue（負例）
980 件	54,176 件

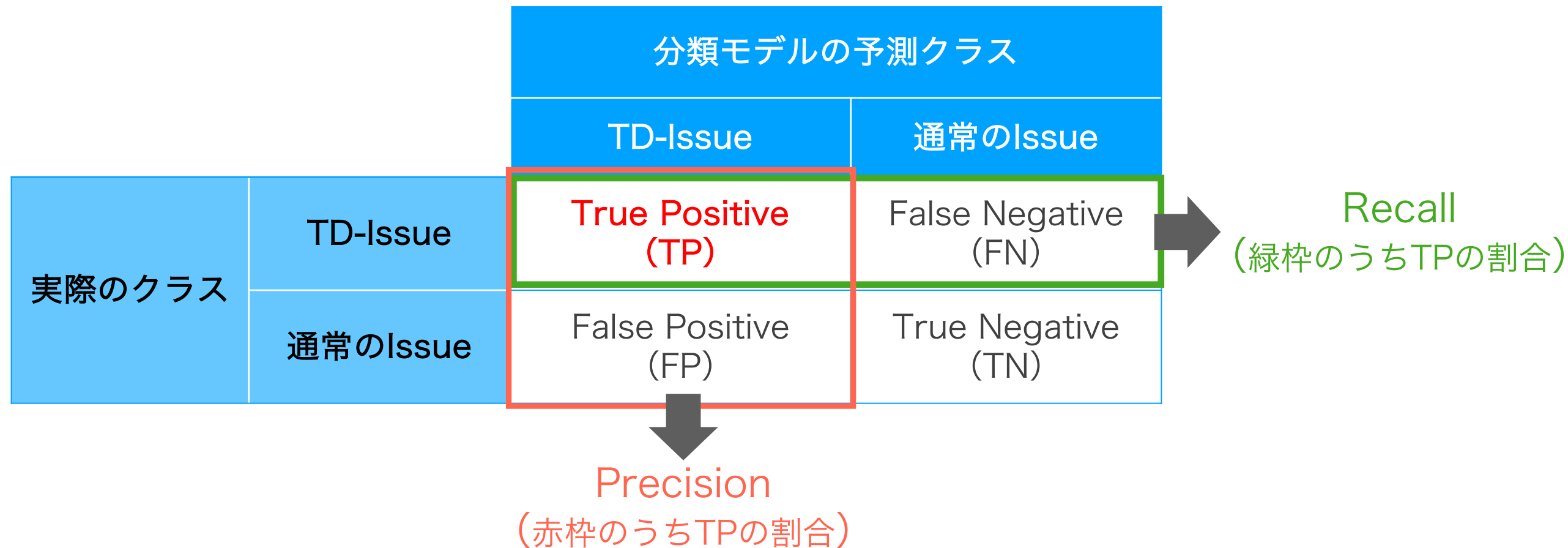
- モデル

- テキスト + （報告者・プロセス・ソースコード）から1つ
- テキスト + （報告者・プロセス・ソースコード）から2つ
- テキスト + 報告者 + プロセス + ソースコード

- 評価方法

- 10分割交差検証を10回試行
- 平均Precision, Recallで精度評価

評価実験：評価指標



$Precision = \frac{TP}{TP+FP}$... TD-Issueと分類して、実際にTD-Issueであった割合

$Recall = \frac{TP}{TP+FN}$... 実際のTD-Issueのうち、分類でTD-Issueを特定した割合

評価実験：結果

- ランダムフォレスト， ロジスティック回帰， SVM
のアルゴリズムを利用

モデル	ランダムフォレスト		ロジスティック回帰		SVM	
	Precision	Recall	Precision	Recall	Precision	Recall
T+R	0.988	0.683	分類結果待ち			
T+P	0.972	0.513				
T+C	0.985	0.448				
T+R+P	0.996	0.728				
T+R+C	0.994	0.732				
T+P+C	0.979	0.582				
T+R+P+C	0.998	0.748	0.993	0.934	0.963	0.936

T：テキスト， R：報告者， P：プロセス， C：ソースコード

今後の予定

- 分類結果待ち
 - ロジスティック, SVM
- 他プロジェクトのデータ収集
- 他プロジェクトのデータを使って, 構築したモデルの評価

