

技術的負債に関する 課題票の分類モデルに 単語分散表現が与える影響の分析

和歌山大学オープンソースソフトウェア工学研究室

田口 舞奈

技術的負債(Technical Debt : TD)

ソフトウェア開発にかかる時間（コスト）の制約のために、実装や設計の質を妥協する事象^[1]

- スパゲッティコード
- コーディング規約に反するソースコード など

⇒蓄積すると除去のための保守コストが増加

✓早期除去には技術的負債の管理が重要

[1] : Ward Cunningham. The wycash portfolio management system. SIGPLAN OOPS Mess., Vol. 4, No. 2, pp. 29–30, 1992.

既存研究

技術的負債検出ツールの開発^[2]

- ソースコードの静的解析
- ソースコードの依存関係解析

➤ 重要でない技術的負債を含んだ大量の検出結果
⇒ 積極的なツールの利用に至らず^[3]

➤ 研究で取り上げられる技術的負債と，開発者が注目する技術的負債との間に違いが存在

[2] : Zengyang Li, Paris Avgeriou, and Peng Liang. A systematic mapping study on technical debt and its management. Journal of Systems and Software, Vol. 101, No. C, pp. 193–220, 2015.

[3] : Emerson Murphy-Hill, Chris Parnin, and Andrew P. Black. How we refactor, and how we know it. IEEE Transactions on Software Engineering, Vol. 38, No. 1, pp.5–18, 2012.

Self-Admitted Technical Debt : SATD

開発者がTDを管理する現象を指す^[4]

- 課題管理システムで管理されるSATD : SATD-Issue

Remove the "no-cache, no-store, must-revalidate" for login.html #24490

☒ Closed User1 opened this issue on Dec, 2020 · 2 Comments

User1 commented on Dec 2, 2020 • edited ▾

While using nextcloud with CDN, the no cache header will prevent the login.html page from being cached. Extending the time for the page to show up

```
server/lib/public/AppFramework/Http/Response.php
Line 55 in 7e6f044

55          'Cache-Control' => 'no-cache, no-store, must-revalidate'
```

User2 commented on Jun 18, 2021 Contributor

cc @nextcloud/server-triage is this feasible?

technical debt

TDを表すラベルが付けられる

[4] : Aniket Potdar and Emad Shihab. An exploratory study on self-admitted technical debt. In Proceedings of the International Conference on Software Maintenance and Evolution, ICSME, pp. 91–100, October 2014.

SATD-Issueに関する先行研究^[5]

目的

- SATD-Issueの特徴分析
- IssueからSATD-Issueを抽出するための分類モデルの作成

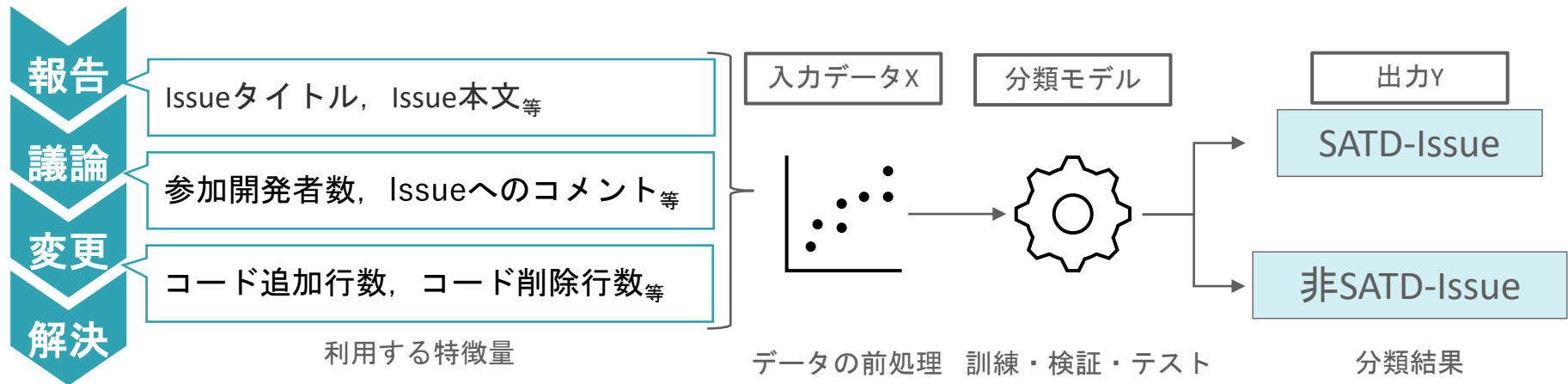


図2：先行研究の分類モデルのアプローチ

[5]：木村祐太, 大平雅雄, 技術的負債に関連する課題票分類手法の構築, 情報処理学会論文誌, Vol. 64, No. 1, 2023

先行研究と本実験の動機の比較

先行研究の動機

- SATD-Issueの絶対数が少ないため、潜在的なSATD-Issueを抽出しデータを増やす
 - 報告～解決までに得られる特徴量を全て用いた分類

本実験の動機

- 技術的負債をできるだけ早期に発見することで、ソフトウェア開発の保守コストの増加を防ぐ
 - 報告時点で得られるテキスト特徴量を用いた分類

先行研究の改善点と本実験の目的

先行研究のIssue分類における改善点

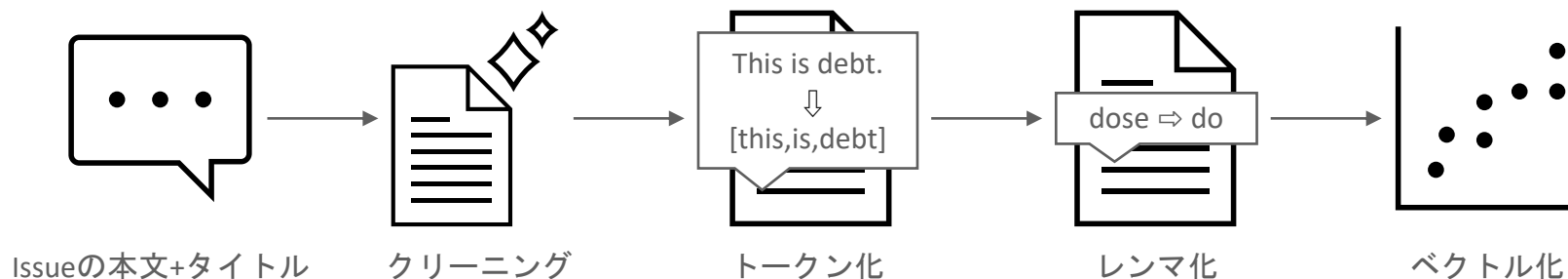
- BoWとTF*IDFによる単純なテキスト処理での分類
⇒ 単語分散表現を用いて精度を改善

本実験の目的

- テキストデータを入力とし、単語分散表現を用いたIssue分類モデルを作成し、性能が向上するかを分析

評価実験(1/2)

テキストデータの前処理



- 利用する単語のベクトル化手法

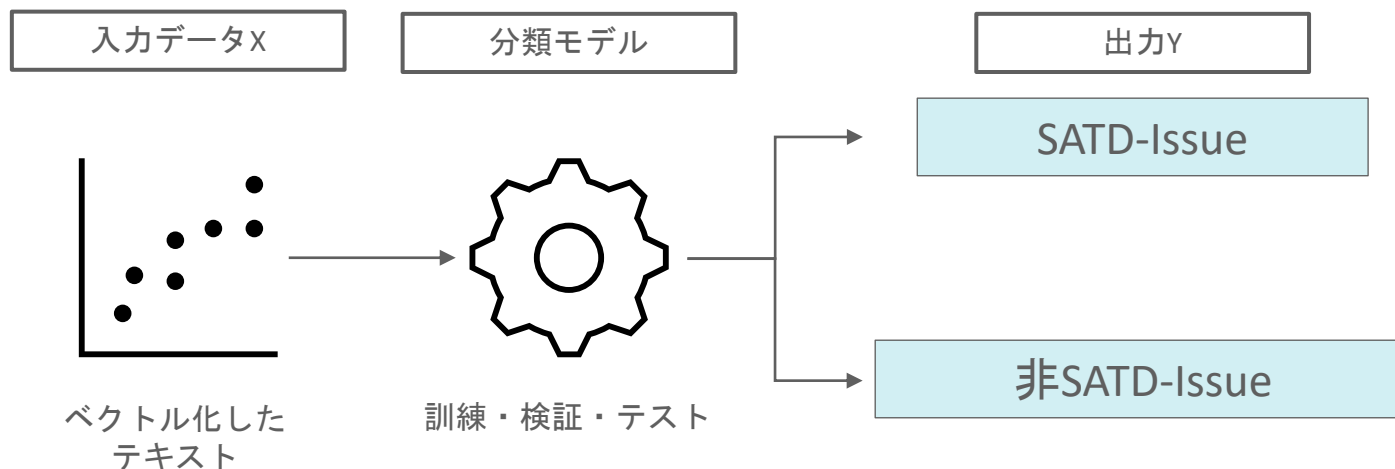
- 先行研究で採用 : BoW, TF*IDF
- 本研究で追加 : Word2Vec, GloVe

Stack Overflow_[6]またはWikipediaから
取得したコーパスで学習したモデル

Wikipediaから
取得したコーパスで学習したモデル

[6] : プログラミング技術に関する知識共有コミュニティ. <https://stackoverflow.com/>

評価実験(2/2)



- 機械学習アルゴリズム
 - ランダムフォレスト
 - ロジスティック回帰
 - サポートベクターマシン

ベクトル化手法：6種類
×
機械学習アルゴリズム：3種類
＝
分類モデル：18種類
どの分類モデルが性能が良いかを調査する

データセット

GitHub 上のスター数上位5000件のうち,

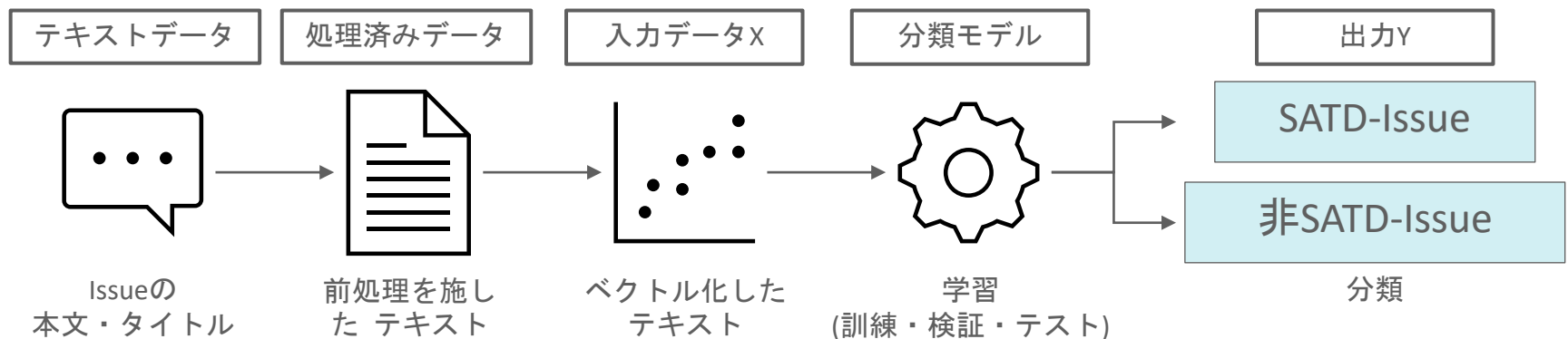
- SATD-Issueをよく扱う4つのプロジェクト_[7]内の,
- 2回以上報告経験のある開発者の解決済みIssue

オーナー / リポジトリ名	SATD の label	非 SATD-Issue 数	SATD-Issue 数
influxdata / influxdb	kind/tech-debt	8,514	120
saleor / saleor	technical debt	2,230	108
nextcloud / server	technical debts	7,255	83
microsoft / vscode	debt	79,829	2,044
合計	-	97,828	2,355

収集期間 : 2022 年4 月26 日まで

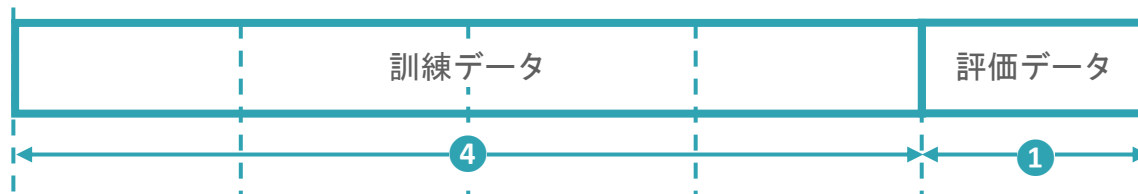
[7] : Laerte Xavier, Fabio Ferreira, Rodrigo Brito, and Marco Tulio Valente. Beyond the Code: Mining Self-Admitted Technical Debt in Issue Tracker Systems. In Proceedings of the 17th International Conference on Mining Software Repositories, MSR, pp. 137–146, June 2020.

分類モデルの作成



- データの正規化：
$$X' = \frac{X - x_{min}}{x_{max} - x_{min}}$$

- 層化5分割交差検証



- 評価データ 交替5回 × 5分割20回 = 合計 100回 学習

結果

表 3. 性能評価実験の結果

機械学習アルゴリズム	ベクトル化手法	Accuracy	Precision	Recall	F1
ランダムフォレスト	BoW	0.978	0.880	0.059	0.111
	TF*IDF	0.978	0.896	0.059	0.110
	Word2Vec-SO	0.977	0.768	0.018	0.035
	Word2Vec-Wiki	0.977	0.663	0.020	0.038
	GloVe-Wiki	0.977	0.639	0.002	0.038

- AccuracyやPrecisionが高く， Recallが低い
⇒ 正例を高い精度で分類， 正例の取りこぼし多数
- 分類モデルの精度を改善するためベクトル化手法を追加したが， 精度は向上しなかった。

追加調査

1. なぜこの結果になったかを知るため

- ランダムフォレストを用いたIssue分類モデルにおいて、重要な特徴量となった単語を調査

2. Issueの分類モデルの精度を向上させるため

- データ数の少ない正例に重み付けし、再実験

追加調査1：結果の原因の調査

ランダムフォレストモデルで分類に寄与した単語

ベクトル化手法	重要度の高い単語				
BoW	oov	unique	inlinecode	link	debt
TF*IDF	oov	link	unique	code	debt

- レンマ化できなかった辞書外単語や固有名詞が、「oov」や「unique」に置換された単語が上位
- ✓ 単語分散表現では置換語を表現していない可能性

追加調査2：分類モデルの精度を向上

表 5. 正例データに重み付けした場合の結果

機 種					
	Recall	Precision	F1	Accuracy	Average
ロジスティック回帰	- Recallが高く， Precisionが低い ⇒ 正例の取りこぼし減少， 正例の分類精度が低下				
	BoW	0.827	0.094	0.738	0.166
	TF*IDF	0.916	0.017	0.672	0.272
	Word2Vec-SO	0.834	0.101	0.776	0.179
	Word2Vec-Wiki	0.821	0.094	0.768	0.167
	GloVe-Wiki	0.831	0.099	0.766	0.175
サポートベクターマシン	BoW	0.872	0.112	0.647	0.190
	TF*IDF	0.928	0.172	0.546	0.261
	Word2Vec-SO	0.906	0.155	0.669	0.252
	Word2Vec-Wiki	0.834	0.100	0.770	0.178
	GloVe-Wiki	0.837	0.101	0.759	0.179

まとめ

目的

- テキストデータに単語分散表現を用いたIssue分類モデルを作成し，性能が向上するか分析

結果

- 実験のPrecisionが高いモデルは，SATD-Issueを高い精度で分類するために利用できる
- 再実験のRecallが上昇した分類モデルは，SATD-Issueをもれなく分類するために利用できる
- データセットの前処理などが原因で性能は向上せず

技術的負債に関する課題票の 単語分散表現を用いたテキスト分類

今後の展望

Issue分類モデルの精度を向上させるために,

1. 前処理の単語の置換に工夫を行う
2. 最新の単語分散表現も分析対象とする
3. 技術的負債を種類ごとに分類

前処理の単語の置換に工夫を行う

1. どのような単語が辞書外とされるか調査

- 辞書外となった単語とその重要度を求める

2. 必要に応じてレンマ化の辞書に単語を登録

- 1.で得られた結果をもとに，重要度の高かった単語を含めてモデルの性能の向上を目指す

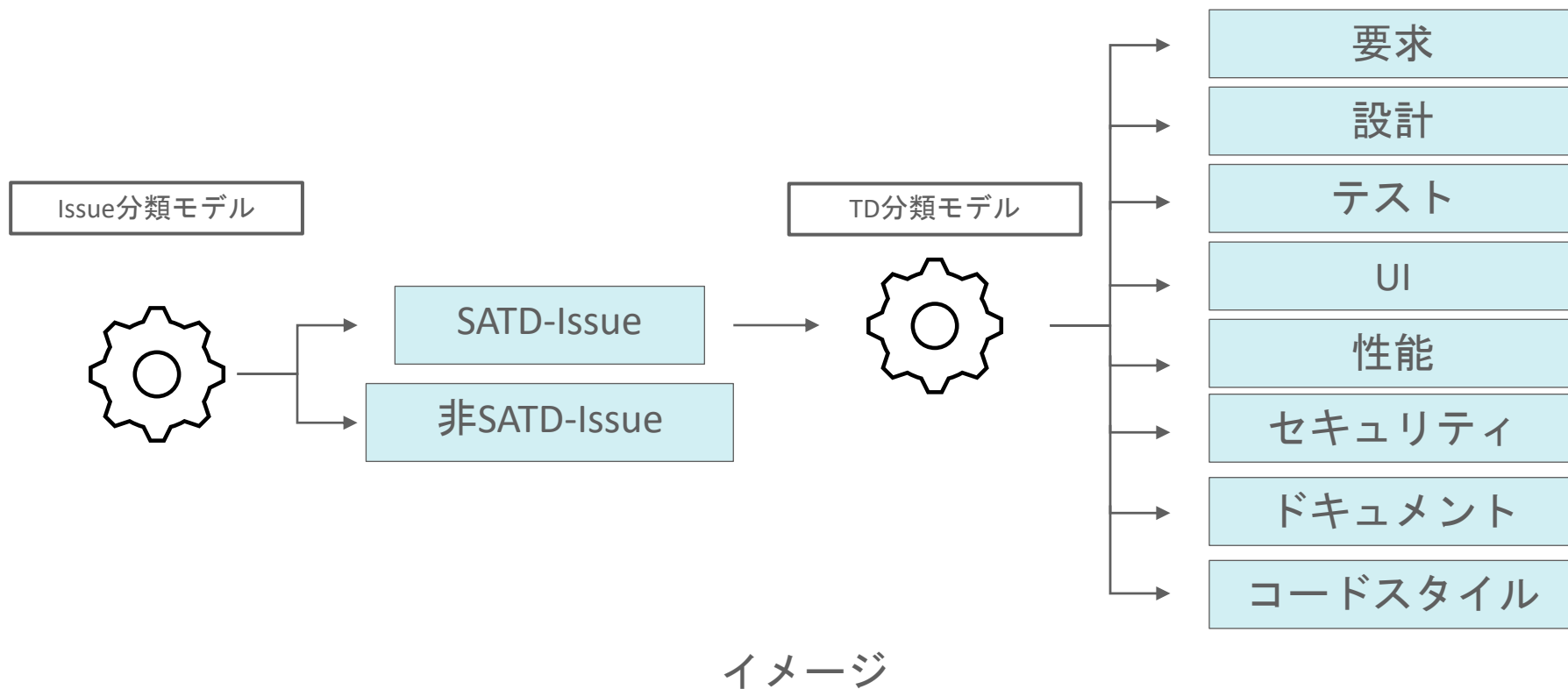
最新の単語分散表現も分析対象とする

現在検討している候補

- Transformer
- BERT
- GPT

技術的負債を種類ごとに分類

- 詳細な分類により正確なSATD-Issueの検出



今後の展望

Issue分類モデルの精度を向上させるために,

- 前処理の単語の置換に工夫を行う
 - どのような単語が辞書外とされるか調査
 - 必要に応じてレンマ化の辞書に単語を登録
- 技術的負債を種類ごとに分類
- 最新の単語分散表現も分析対象とする
 - Transformerや関連する最新モデルを取り入れる