

テキスト的表現が異なる 重複バグ報告に対する 検知手法の提案

発表者: 稲葉 真幸

ソフトウェアのバグ報告の重複

- 大規模なソフトウェアシステムでは、毎日何百ものバグが報告されるため
複数の人が同じバグ報告を提出すると重複が発生
- ソフトウェアのバグ報告の約40%程度は重複している[1]
→保守の際には大きな負担となる
- 従来、テキスト的に類似したバグ報告の重複を検出することに重点
→しかし、実際には多くの重複バグレポートが
テキスト的に類似しておらず、従来の手法では不十分な可能性

[1] W. Zou, D. Lo, Z. Chen, X. Xia, Y. Feng and B. Xu. How Practitioners Perceive Automated Bug Report Management Techniques. In IEEE Transactions on Software Engineering, vol. 46, no. 8, pp. 836-862.

“テキスト的表現が異なる”重複バグ報告

簡単な例)

これらは同じ

「Firefoxが起動時にクラッシュ」

というバグを報告しているが
事象の表現方法や語彙が異なる。



Firefoxを開くと何も表示されずに
すぐクラッシュします。



私のPCでFirefoxを起動すると突然終了します。
エラーダイアログは出ません

- 同じ事象でも、報告者の主観や発生状況によって報告内容が変わる
- 実際には発生した事象以外にも
期待値や再現手順など様々な内容が含まれ、例より複雑
- 重複バグのうちこれに該当するものは19%~23%存在[2].

[2] S. Jahan and M. M. Rahman. Towards Understanding the Impacts of Textual Dissimilarity on Duplicate Bug Report Detection. In SANER, Taipa, Macao, 2023, pp. 25-36,

重複バグ報告をなぜ検知したいか

- 開発者の時間は有限である
- 自動で検知できるようになれば
重複バグの分類にかかっていたリソースを
他の開発や、バグ自体の修正に充てることができ、
結果としてソフトウェア自体の改善につながる。

重複バグ報告検知の取り組み

- テキスト的表現が一致する報告の検知に焦点が当てられていた.
- 重複バグを自動検出する試みとして過去10年来行われてきたもの
 - ①自然言語処理（NLP）：テキスト的な非類似の処理 難
 - ②情報検索（IR）：語彙的な不一致の処理 難
 - ③機械学習（ML）：データ不均衡, 一般性の問題→自然言語の複雑さゆえに完全とはいえない.
- これらの手法は,
テキスト的に異なる表現の検知や新しいバグ報告への適応に問題あり.

先行研究で使われていた手法

バグ報告の類似度を計算するために使われていた技術

- BM25

情報検索のためのランキング関数の一種

検索クエリと文書の間の関連性を評価するために用いられる。TF-IDFの拡張とも考えられる。

- LDA (Latent Dirichlet Allocation : 潜在ディリクレ配分)

トピックモデルの一種で、テキストデータのクラスタリングに使用される。

文書が複数のトピックから生成されると仮定し、トピックの分布や関連するキーワードの推定が可能

- GloVe (Global Vectors)

分散表現の一種。

文章に含まれる「単語」をベクトルに変換するWord2Vecに、大域的な分析を加える改良を行ったもの

これらが主に**重複バグ報告**の検知手法として使われていた

→しかし、**表現が異なるもの**に対しては検知精度が不十分である[3]との結果

目的

テキスト的表現の異なる重複バグ報告に対する検知精度を向上させる.

提案手法 (WIP)

- ① 分散表現をより大規模なものに置き換える
- ② 追加情報の活用 (今後)
- ③ 先行研究に用いられているデータセットのクリーンアップ (今後)

提案手法①： 分散表現をより大規模なものに置き換えて性能比較

より大規模な分散表現モデル (例:BERT, XLNet, GPT等のLLM) を採用することで検知精度の向上を目指す。
また、それらのモデル間で重複を検知する性能を比較する。

現在利用しようとしている分散表現モデル

- Sentence Transformers (Sentence BERT) : 複数文章をクラスタリングするのに特化
- BERT_base (Bidirectional Encoder Representations from Transformers)
- BERT_SE: ソフトウェア工学について事前学習されたBERT (使えそうなら)
- XLNet
- GPT

提案手法①：

分散表現をより大規模なものに置き換えて性能比較

これらに変えると何が良いのか：

- 従来のRNN(GloVe等に採用されている)では，文章中の単語が登場した順番通りに重み付けされていた。
→長い入力に弱い，重要な情報が残りづらい
- Transformerを用いた分散表現を利用
→Attention機構によって重要な情報に「注目」し，
従来よりも文脈を考慮しながら判定可能

提案手法②：追加情報の活用（構想）

- バグ報告のテキスト情報だけでなく、プロジェクトのVCS等に含まれる追加情報を活用して、重複の検知精度を向上させる
- 将来的にバグ自体の個体識別や、バグとバグ報告、修正状況の紐付け等が行えればバグ関連の工数をより減らせるのではないか

例)

コミットメッセージ、コードの変更、開発者の情報など

→報告の背景となるコードの変更履歴や開発者の情報などから、テキスト情報だけでは捉えきれない類似性を調査し、検知精度向上や紐付けに役立てる。

進捗・目標

- 「分散表現をより大規模なものに置き換えて性能を比較」
を終わらせる

- Sentence Transformerへの置き換え作業中

- 以下のモデルでも同様に行って性能を比較

BERT_base (Bidirectional Encoder Representations from Transformers)

BERT_SE: ソフトウェア工学について事前学習されたBERT (使えそうなら)

XLNet

GPT

今後の研究課題

“表現は違うが意味は同じ”という状況は、
重複バグだけでなく他の分野でも起こり得る。

例) ソフトウェア工学分野なら・・・

- 要件仕様書やユーザストーリー
同じ要件が異なる表現で記述されることがある。
- コードレビューのコメントや議論
同じ問題に対するフィードバックが異なる表現で複数回提供されることがある。
- Stack Overflowの質問と回答
同じ問題に対する質問や回答が異なる表現で投稿されることがある。
- ドキュメンテーション
規模が大きくなると同じ情報が異なる場所や異なる表現で記述されることがある。

→精度が高められれば、これらにも応用が可能