

バグ限局に有効なバグレポート及び ソースコードの特徴に関する研究

工学部工学科応用情報工学コース

ヒューマンウェア工学研究室

B4 内山稜太

研究概要

●目的

情報検索に基づくバグ限局に有効なバグレポートやソースコードの**特徴**を分析

●手法

バグレポートとソースコードの特徴量を取得し、ランダムフォレストの重要度分析を行う

●結果

バグレポートでは**3種類**、ソースコードでは**2種類**の特徴量が重要であることを確認

背景

- ソフトウェアで発生したバグはバグレポートにバグの内容を記述して報告される
- 報告されたバグレポートの内容からバグの原因箇所を特定(**バグ限局**)し, バグの修正を行う
- バグの原因と疑われるソースファイルを自動で特定する手法として情報検索に基づいた手法(**IRベースのバグ限局**)がある

Bug 11076 ・ Deleted configs remain in favorite list
Add some launch configuration to your favorites.
Delete this configurations via the launch configuration dialog.
The configurations remain listed in the favorites.

IRベースのバグ局限

- ソースファイルを検索対象
- バグレポートの内容の一部またはすべてから検索クエリを生成
- 検索エンジンを使用し検索クエリからバグの原因と思われるソースファイルを検索
- ランキング形式(1位から10位)でバグの原因と疑わしいソースファイルが出力される
⇒実際にバグの原因であったソースファイルが上位で出力されると良い

先行研究

Rahman, M. M., Khomh, F., Yeasmin, S. and Roy, C. K.: The forgotten role of search queries in IR-based bug localization: an empirical study, Empir. Softw. Eng., Vol. 26, No. 116, pp. 116:1–116:56 (2021).

●**検索クエリ生成手法**の比較実験

バグレポートからどのキーワードを抽出して検索クエリを生成するか

●例

- ベースライン(基本的な前処理のみ行う)
- 頻度ベース(トークンの出現頻度に基づき重要度を求める)
- グラフベース(トークンの依存関係に基づき重要度を求める)

研究課題

- バグレポート内にバグ限局に**必要な情報が十分**
にない場合, キーワード抽出に工夫を凝らしても**バ**
グ限局は難しい
- 先行研究でバグ限局精度の良いクエリの特徴の分析
は行われたが, **バグレポート自体の特徴**は分
析されていない
- 検索対象である**ソースコード**についても同様に特
徴の分析は行われていない



IRベースのバグ限局で成功または失敗するバグレポート
及びソースコードの特徴は何か？

データ分析(1/2)

● 対象

先行研究で使用された6つのオープンソースプロジェクトを使用

- 2,289個のバグレポート
- 2,123個のバグ有りソースファイル

● 特徴量

- バグレポートから95種類
- ソースコードから106種類

データ分析(2/2)

- ランダムフォレストによる**重要度分析**
抽出した特徴量を説明変数, バグ限局精度(RR,AP,Hit@K)を
目的変数としてランダムフォレストを構築
(ただし, 相関の強い特徴量は一方を除外)
- バグ限局の成否による特徴量の分布の違いがある
か検定
2種類の有意差検定(有意水準: 0.05)

結果

- **バグレポートの重要な特徴量**

- 数字の割合
- キャメルケース分割し重複を除いたその他の品詞の割合
- Java予約語の割合

- **ソースコードの重要な特徴量**

- コメント内の名詞の割合
- メソッド名の平均文字数

考察 (バグレポート)

バグ限局の成否で**特徴量の分布**の違いを調査

- 数字の割合(低い方が望ましい)
クエリ作成のための前処理が影響(数字はクエリに使用しない)
数字の割合が高いと**使用されない情報**がバグレポート内を多く占める
- その他の品詞の割合(高い方が望ましい)
自然言語で頻繁に書かれる単語でないものが多い
エラーメッセージや**スタックトレース**などが記載

```
JDIDebugUIPlugin is incorrectly triggering the creation of the  
imagedescriptor registry at shutdown. Current Java thread:  
at org.eclipse.swt.internal.motif.OS.XtOpenDisplay(Native Method)  
at org.eclipse.swt.widgets.Display.createDisplay(Display.java:442)
```

スタックトレースを含むバグレポートの例(一部)

考察 (ソースコード)

バグ限局の成否で**特徴量の分布**の違いを調査

- コメント内での名詞の割合 (低い方が望ましい)
1,2単語のみのコメントでは名詞の割合は高くなりやすい
⇒ **コメントの記述量**が必要
コメントの**トークン数**ではバグ限局成功時に**多い傾向**
- メソッド名の平均文字数 (長い方が望ましい)
処理内容について**適切に情報を提供**するためにはメソッド名は長くなりやすい

メソッドの例(バグ限局成功)

```
extendTimeToLiveForNullCheckedField
```

メソッドの例(バグ限局失敗)

```
run
```

まとめ

● バグレポートについて

“数字の割合”や“その他の品詞の割合”が重要



- 一般的な文だけでなく**エラーメッセージ**や**スタックトレース**等の記述が望まれる

● ソースコードについて

“コメント内の名詞の割合”や“メソッド名の平均文字数”が重要



- **コメントの内容が充実**であることや**識別子を説明的**にすることが望まれる

今後の課題

- 自動チェックシステムの確立

バグレポート及びソースコードの重要な特徴量に望ましくない傾向が見られた場合
⇒警告等を出し改善を求める

- 有効性の確認

実際のバグ報告に対し自動チェックシステムの有効性を検証