

バグレポートの検索性向上のための機械学習による 文章単位の自動ラベリング

野寄祐樹† 鷲崎弘宜† 深澤良彰† 鹿糠秀行‡ 大島敬志‡ 土屋良介†

早稲田大学基幹理工学部情報理工学科† (株)日立製作所 研究開発グループ‡

1. はじめに

ソフトウェアの開発者やユーザは不具合に直面した際、原因や修正方法を調べるために、過去のバグレポートを検索する。しかし、バグレポートは構造化が為されていない文章の羅列であることが多く、検索性に課題がある。

例えば、ユーザは遭遇した不具合の修正方法を調べる際、過去のバグレポートから類似の現象を見つけるため、現象に関係するキーワードで全文検索を行う。しかし、バグレポートは不具合の現象に限らず、原因や修正方法、それらに辿り着くまでの議論が記載されているため、単なる全文検索では検索対象に冗長な記述が含まれ、検索精度が低い値となってしまう。

本課題の解決を目的に、我々は各バグレポートの文章毎に意味的ラベルを付与する手法を提案する。既存研究においては、バグレポートの理解性や検索性を向上する目的で、バグレポート中の各文章に対して重要文判定を行いサマライズする研究[1][2]や、バグレポートをバグ種別等でカテゴリ化する研究[3][4]が存在する。しかし、多様な検索目的に応じて検索対象を適切に絞り込む目的においては、これらの既存研究は効果的でない。

本研究では、機械学習によりバグレポート中の文章に5種類の意味的ラベルを付与する手法を提案する。本手法では、学習時の説明変数として、文章に含まれる単語の情報に加えて、文書構造的特徴を用いることによりラベル付け精度の向上を狙う。本手法により、検索者の目的に合致するラベルを選択し検索対象を絞り込むことができ、検索精度の向上が見込める。

本研究の研究課題を以下に示す。

研究課題1: バグレポートの文書単位ラベル付けにおいて文書構造的特徴を説明変数に加えることによりラベル付け精度はどの程度向上するか

研究課題2: 文章単位の意味的ラベル付けによりバグレポートの検索精度はどの程度向上するか

OSS(Open Source Software)のバグレポートを対象とした実験の結果、文書構造的特徴を説明

変数に加えることで、ラベル付け精度は、4つの学習アルゴリズムにおいて6~24%向上した。また、本手法の適用により、検索精度の評価指標であるF値が約0.168改善したことを確認した。

2. 文章単位の自動ラベリング手法

2.1 提案手法の概要

本手法は図1に示す3つのステップからなる。S1: 予め文章毎に意味的ラベルを付与したバグレポートを教師データとして準備し、教師あり学習を行うことで分類器を作成する。S2: 作成した分類器に新たなバグレポートを入力すると、分類器の予測結果に則り、バグレポート中の各文章に対して意味的ラベルが自動的に付与される。S3: 検索者は、自身の目的に合致したラベルを選択することで、該ラベルが付与された文章のみを対象としたキーワード検索が可能となる。

2.2 意味的ラベル

実際のバグレポートに基づく調査、及び、企業有識者との議論の結果、バグレポートの記載内容は、表1に示す5種類のカテゴリに分類できることを確認した。本カテゴリをラベルとして各文章へ付与することで、目的に応じた検索対象の絞り込みを可能とする。

2.3 分類器における学習のための説明変数

S1で使用する学習時の説明変数として、下記3つの文章の特徴を用いる。

- **Bags-of-Words**: 文章に含まれている単語の出現回数をベクトル表記したもの。
- **文章の位置**: バグレポートの中でその文章がどの位置にあるかを示す情報。
- **前文ラベル**: 一つ前の文章に付与したラベル。文章の位置と前文ラベルは、バグレポートにおいて、現象が最初に報告された後に、議論を経て原因や解決策に至るといった文書構造的特徴を学習させるための説明変数である。

3. 評価

3.1 実験概要

研究課題に答えるため、ラベル付け精度と検

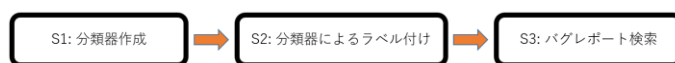


図1. 提案手法の処理ステップ

Automatic Labeling of Sentences with Machine Learning to Search for Bug Reports

† Waseda University, School of Fundamental Science and Engineering

‡ Hitachi, Ltd. Research & Development Group

表 1. 意味的ラベル一覧

ラベル	概要
現象	バグの振る舞いに関する言及
原因	バグがなぜ生じたのかに関する言及
解決策	バグの治し方やパッチに関する言及
議論	バグを修正するにあたって行われた議論
その他	上記以外

表 2. ラベル付け精度の実験結果

	a	a+b
SVM	0.385477	0.497746
NN	0.485116	0.55809
NB	0.477356	0.536455
RF	0.402767	0.63725

表 3. 検索精度の実験結果

	A1	A2
ヒット件数	49	25
Precision	0.3877551	0.64
Recall	1	0.8421053
F値	0.55882353	0.7272727

検索精度について評価した。データセットとして[3]の研究で使用されているバグレポートを使用した。本データセットには Eclipse Platform, Gnome, Mozilla, KDE の 4 つの OSS のバグレポート 36 個が含まれている。本研究では、本データセットの内、17 個のバグレポート、984 個の文章に手作業でラベルを付与し、交差検定によりラベル付け精度を評価した。そして、残りの 19 個のバグレポートを用いて検索精度を評価した。

機械学習には Python の「Scikit-learn」ライブラリを使用した。Bag-Of-Words 作成には Python の「gensim」ライブラリを使用した。

3.2 分類器によるラベル付け精度の評価

表 2 に、S2 の実験結果を示す。本実験においては、分類器の作成に際して、サポートベクターマシン(SVM)、ニューラルネットワーク(NN)、ナイーブベイズ(NB)、ランダムフォレスト(RF)の 4 種類の学習アルゴリズムを用いて交差検定をした。また、Bag-Of-Words のみを学習した場合(a)、文書構造的な特徴を説明変数に加えて学習した場合(a+b)、それぞれに対してラベル付け精度を比較評価した。

表 2 より、文書構造的な特徴を説明変数に加えることで、各学習アルゴリズムにおいてラベル付け精度が 6~24%向上したことを確認した。本結果が研究課題 1 の回答となる。

3.3 バグレポート検索精度の評価

S3 のバグレポートの検索結果を以下に示す。前節でラベル付け精度が最も高かったランダム

フォレストを用いて学習した分類器により、学習データとしては使用していない 19 個のバグレポートを対象にラベルを付与した。本実験では、19 個のバグレポートの中から各バグレポートを検出する為の検索キーワードをそれぞれ 1 つずつ設定し、計 19 回検索を行った。なお、キーワードは、バグレポートのタイトルに含まれる単語に基づいて、不具合の現象に係る単語を設定した。

表 3 に、検索精度の実験結果を示す。バグレポート内のタイトルを除く全文章を対象にキーワード検索を行った結果が A1 である。一方で、意味的ラベル「現象」が付与された文章を対象にキーワード検索を行った結果が A2 である。

A1 では、各キーワードに対して、正解を含む複数のバグレポートがヒットした為、Recall は高いが、Precision が低い値となっている。A2 では、検索対象を絞り込んだ為、Recall が 0.158 低下したものの、Precision は約 0.252 向上した。これにより、検索精度の評価指標である F 値が約 0.168 改善したことを確認した。本結果が研究課題 2 の回答となる。

4. おわりに

本研究ではバグレポートの各文章に意味的ラベルを付与することで検索性の向上を図った。学習データ数が課題である為、データ数を増やし、ラベリング精度の向上、検索性の更なる向上をめざしていきたい。

6. 参考文献

- [1] S. Rastkar, G. C. Murphy, and G. Murray. "Automatic summarization of bug reports." IEEE Trans. Software Eng., 40(4):366-380, 2014.
- [2] Ferreira, E. Cirilo, V. Vieira, and F. Mourao, "Bug report summarization: An evaluation of ranking techniques," in 2016 X Brazilian Symposium on Components, Architectures and Reuse Software, 2016, pp. 101-110.
- [3] Y. Kashiwa, H. Yoshiyuki, Y. Kukita, M. Ohira, "A Pilot Study of Diversity in High Impact Bugs", Proceedings of 30th The International Conference on Software Maintenance and Evolution, pp. 536-540, 2014.
- [4] Nachai Limsettho, Hideaki Hata, Akito Monden and Kenichi Matsumoto, "Automatic unsupervised bug report categorization." in Proceedings of the 6th IEEE International Workshop on Empirical Software Engineering in Practice, 2014, pp. 7-12.