N-gram IDF によるバグレポートの分類の試行

神吉裕次[†1]　門田暁人[†1]　畑 秀明[†2]

概要：本稿では，Inverse Document Frequency（IDF）の理論的拡張である N-gram IDF を用いて，バグレポートの自動分類を試みる．

An Attempt to Bug Report Classification using N-Gram IDF

YUJI KANKI[†1]　AKITO MONDEN[†1]　HIDEAKI HATA[†2]

1. はじめに

大規模化･複雑化している近年のソフトウェア開発では，非常に多くのバグが報告される．そのため，タイプミスなどの「軽微なバグ」からセキュリティに関する「緊急度の高いバグ」までが同等に扱われてしまい，緊急度の高いバグが見逃されることが問題となっている．また，バグの分類には多大な労力が必要となることが問題である．例えば，Herzig らは，7401 件のバグレポートを手作業により分類するのに 90 日を費やした[1]．緊急度の高いバグを見逃さないためには，バグレポートを自動分類する技術の開発が重要となる．

本稿では，バグレポートを分類するためのテキストマイニング技術として，Inverse Document Frequency（IDF）の理論的拡張である N-gram IDF を採用する．IDF とは，単語の珍しさの尺度であり，単語の重要度を示す．また，IDF は情報検索の重み係数としてよく利用される．しかし，IDF は N≧2 の N-gram，すなわち，2 つ以上の単語から構成される句を扱うことができない．この問題を解決するために，白川らは IDF と Multiword Expression Distance（MED）を組み合わせた N-gram IDF を提案している[2]．N-gram IDF では，任意の長さの N-gram をすべて抽出し，単語と句の重みを比較することによって，最も有力な N-gram を選択できる．得られた N-gram は，分類タスクの特徴として利用できる．

本稿では，4 つのオープンソースソフトウェアプロジェクト（Apache Ambari，Camel，Derby and Wicket）について，各 1000 件のバグを 10 名の研究者が手動で分類したデータセット[3]を用いて評価実験を行う．このデータセットでは high impact bug の観点でバグレポートを手動で分類している．high impact bug とは開発者やユーザに大きな影響を与えるバグであり，その分類を表 1 に示す．

本稿では，N-gram IDF から得られる特徴を用いて分類モデルを構築し，high impact bug の観点で分類を試行して，その性能を評価する．

表 1　High Impact Bug の分類

大分類	小分類	説明
High Impact Process Bug	Surprising	予期せぬ時期に予期せぬ箇所で発生するバグ
	Dormant	次のバージョンまで発見されなかったバグ
	Blocking	他のバグの修正を妨げるバグ
High Impact Product Bug	Security	セキュリティ上の問題に関わるバグ
	Performance	パフォーマンスの低下を招くバグ
	Breakage	機能追加やバグ修正時に混入したバグ
Low Impact Bug	—	その他のバグ

2. 提案方法

バグ分類モデルを構築するための提案方法は，テキスト処理フェーズと分類フェーズの 2 つに分かれる．

2.1 テキスト処理フェーズ

このフェーズは 3 つのステップで構成される．

2.1.1 テキスト前処理

このステップでは，まず，バグレポートファイルから，プログラミング構文に関連する文字（例えば，“==”，“+”，“-”）を取り除く．次に，一般的な英語のストップワード（例えば，“a”，“the”）をバグレポートから取り除く．

記号やストップワードはバグ分類に有用といえないため，このステップが必要となる．

2.1.2 N-gram IDF の適用

N-gram IDF を適用することにより，重複する N-gram の中から最も有力な N-gram を得ることができる．そのため，ドキュメントのテキストから任意の長さの特徴語を抽出することができる．本研究では，N-gram Weighting Scheme ツールを使用する．ツールを前処理されたテキストデータに適用した後の出力は，すべての効果的な N-gram のリストである N-gram 辞書となる．

†1 岡山大学
Okayama University
†2 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

表 2　High Impact Bug の分布

Project	Surprising	Dormant	Blocking	Security	Performance	Breakage
Ambari	317	2	10	29	41	8
Camel	388	153	18	34	95	55
Derby	147	137	23	88	101	197
Wicket	359	98	5	10	83	66

2.1.3 特徴抽出

N-gram辞書を得た後, バグレポートのコーパスとN-gram辞書から特徴ベクトルを作成する．各バグレポートについて，各 N-gram の出現頻度をカウントし，その値に基づいてベクトル要素を作成する．これらのベクトルは，次の分類フェーズのための特徴として利用される．

2.2 分類フェーズ

このフェーズでは，分類モデルの構築を可能とするために, 2 つのデータセットを処理する．1 つはテキスト処理フェーズの出力であるベクトル要素，もう 1 つはバグレポート ID とその正しいバグ分類を含む正解データセットである．これらのデータセットを用いて，次の 2 ステップを実施する．

2.2.1 特徴ベクトル前処理

前のステップから特徴ベクトルを取得した後，正解データセットに基づいて特徴ベクトルをラベル付けする．これにより，特徴ベクトルがバグ分類と結び付けられる．

2.2.2 分類モデルの構築

前処理された特徴ベクトルを基に分類モデルを構築する．本稿では，分類モデルとして，集団学習モデルの一つであるランダムフォレストを採用し，マルチラベル分類モデルを構築する．

3. 評価実験

本実験では, JIRA をバグトラッキングシステムとして使用する 4 つのオープンソースソフトウェアプロジェクト（Apache Ambari，Camel，Derby and Wicket）から収集したバグレポートのデータセットを使用する．各プロジェクトに含まれる high impact bug の分布を表 2 に示す．

本実験では，分類モデルを評価するための指標として，適合率，再現率を採用する．評価にあたっては，3 分割交差検証を行う．3 分割交差検証では，まず，データセットをランダムに 3 個の等しいサイズのサブセットに分割する．それから，これらの 3 個のサブセットのうち，2 個のサブセットを学習データとして使用し，残りのサブセットは評価用データとして使用する. このプロセスを 3 回繰り返し，すべてのサブセットを評価用データとして 1 回ずつ使用する．そして，3 回実行後の適合率，再現率の平均値をそれぞれ求める．実験にあたっては，low impact bug のデータも含めてモデル構築と評価を行う．

表 3　3 分割交差検証の結果

Project	適合率（%）	再現率（%）
Ambari	32.1	10.0
Camel	39.7	6.3
Derby	68.3	17.7
Wicket	36.5	6.4

4. 結果

表 3 に 3 分割交差検証の評価結果を示す．適合率，再現率は，（low impact bug を除いた）6 つの high impact bug の分類の平均値を示す．表 3 より，再現率は適合率と比べて低くなることが見て取れる．

5. おわりに

本稿では，各バグレポートに含まれるテキスト情報に基づいて，バグレポートを自動分類することを目的として，N-gram IDF を用いたバグ分類の試行を行った．4 つのオープンソースソフトウェアプロジェクトを用いて，3 分割交差検証を行った結果，再現率は適合率と比べて低い値となった．この結果は，緊急度の高いバグを見逃さないようにするという本研究の動機からすると，改善の必要がある．

今後の課題として，分類方法を改良し，性能向上を図っていくことが挙げられる．また，N-gram IDF と他の方法を比較する予定である．さらに，各バグ種別の分類に寄与する N-gram についての分析を行っていきたいと考えている．

6. 参考文献

[1] K. Herzig, S. Just, and A. Zeller, “It’s not a bug, it’s a feature: how misclassification impacts bug prediction,” in Proceedings of the 2013 International Conference on Software Engineering, pp. 392–401, 2013.

[2] M. Shirakawa, T. Hara, and S. Nishio, “N-gram idf: A global term weighting scheme based on information distance,” in Proceedings of the 24th International Conference on World Wide Web, pp. 960–970, 2015.

[3] M. Ohira, Y. Kashiwa, Y. Yamatani, H. Yoshiyuki, Y. Maeda, N. Limsettho, K. Fujino, H. Hata, A. Ihara and K. Matsumoto, “A dataset of high impact bugs: Manually-classified issue reports”, In Proceedings of the 12th Working Conference on Mining Software Repositories, pp. 518–521, 2015.