T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

論文 / 著書情報
Article / Book Information

論題	欠陥箇所検索に有効なバグレポート中のキーワード抽出
Title	Extracting Optimal Query Keywords from Bug Report for IR-based Bug Localization
著者	林 和輝, 小林隆志
Authors	Kazuki Hayashi, Takashi Kobayashi
出典	電子情報通信学会技術研究報告, Vol. 121, No. 416, pp. 84-89
Citation	IEICE technical report, Vol. 121, No. 416, pp. 84-89
発行日 / Pub. date	2022, 3
権利情報 / Copyright	本著作物の著作権は電子情報通信学会に帰属します。 Copyright(c) 2022 IEICE

Powered by T2R2 (Science Tokyo Research Repository)

一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
SS2021-56(2022-03)

欠陥箇所検索に有効なバグレポート中のキーワード抽出

林　　和輝†　小林　隆志†

† 東京工業大学 情報理工学院 情報工学系　〒 152–8552 東京都目黒区大岡山 2–12–1
E-mail: khayashi@sa.cs.titech.ac.jp, tkobaya@cs.titech.ac.jp

あらまし　ソフトウェアの欠陥箇所特定を支援するために 情報検索技術に基づく欠陥箇所検索手法が研究されている．これら研究では，検索クエリにバグレポート中の部分的な文章のみを利用することで精度が改善されることが実験的に示されてきた．近年，欠陥の位置から逆算することで理想的な検索結果を提示する模範クエリを構成し，情報検索技術に基づく手法の限界や実験的に示されてきた知見を検証する試みがなされた．しかし，その模範クエリを構成する方法については明らかとされていない．本研究では模範クエリの特徴を分析し，それを構成する単語が持つ構造的特徴や自然言語的特徴を調べる．調べた結果を元に，模範クエリを再現する手法を提案する．分析の結果，模範クエリの構成単語に構造的特徴や自然言語的特徴が確認できた．また，提案手法は先行研究より模範クエリに近い検索クエリを構成できることがわかった．

キーワード　自然言語処理，クエリ改善，Bug Localization，ソフトウェア保守

1. はじめに

ソフトウェアに対して不具合が発見された際には，その不具合を修正するために原因箇所を特定する必要がある．この特定作業では，本来想定されていた挙動や発生した問題についてまとめられたバグレポートの参照が有効であることが知られている [1]．しかし，依然として原因箇所の特定に多くの労力を要するため，バグレポートをもとに情報検索技術を用いて原因箇所を機械的に特定する手法が提案されてきた [2–5]．本研究ではこれらの手法を欠陥箇所検索と呼ぶ．

最も単純な欠陥箇所検索手法では，検索クエリにバグレポートの内容をそのまま用いる．ただし，この方法では実用性を伴う欠陥箇所検索は厳しいことが指摘されている [6]．これまでに，バグレポート以外の副次的情報を補うことで検索精度を向上させる試みが数多くなされており，類似バグレポート の情報を用いる手法 [3] やスタックトレースを用いるもの [4] などが提案されている．近年ではこれらの副次的情報も併せた埋め込み表現を用いた検索手法 [5] が高い精度を示すことが報告されている．しかしながら，実際の状況においてはこれらの研究が利用する副次的情報が利用できない状況も少なくない．このため，本研究ではバグレポート中の文章のみを利用する欠陥箇所検索手法の精度改善を対象としている．

高い精度で欠陥の位置を特定するにはバグレポートの一部のみを抽出して検索クエリとして用いる「クエリ削減」が有効であることが報告されている．特に，バグレポートの構成要素に着目して部分的に利用することで精度が改善されることが実験的に示されてきた [6–9]．

クエリ削減の戦略と欠陥箇所検索の潜在的有用性については，Mills らが正解から逆算することで構成した模範クエリを構成して議論をしている [10]．この研究では，情報検索技術に基づく手法にはまだ精度改善の余地があること，これまで実験的に示されてきた知見は最適なクエリ削減戦略ではない可能性があることが示唆されている．しかしながら，模範クエリに近い検索クエリを構成する方法については明らかにされておらず，実験的に示されてきたクエリ削減手法をどのように改善するべきかの指針は存在しない．

本研究では，Mills らが逆算して求めた模範クエリを分析することでその特徴を明らかにし，その特徴に基づいてバグレポート中の単語を選択するクエリ構成手法を提案する．まず，模範クエリとバグレポートの構成要素との対応関係を Mills らより詳細に解析し，模範クエリを構成するうえでのバグレポート構成要素の重要性を再調査する．次に，模範クエリに登場する単語出現箇所の自然言語の観点での特徴を詳細に調査する．本研究では品詞情報や，文の木構造にとくに注目をして分析を行う．最後に，これらの分析で得た知見をもとに既存研究と比べて優れた検索クエリを構成する手法を提案する．

以上を踏まえ，本研究は次の 3 つの Research Question (RQ) に回答することを目的とする．

RQ1:　模範クエリを構成する単語は，バグレポートのどの構成要素に含まれやすいか

RQ2:　模範クエリの自然言語的な観点での特徴とは何か

RQ3:　提案手法は既存手法より模範クエリに近い検索クエリを構成するか

以下ではまず，関連研究について説明した後，3. 節 で模範クエリに対する調査方法について説明し，4. 節　で調査の結果と RQ1, RQ2 への回答を述べる．次に，得られた知見に基づく検索クエリ削減手法を 5. 節 で説明し，6. 節 でその提案手法の評価を行ない，RQ3 に回答する．

2. 関連研究

バグレポートから作成した検索クエリが有用な結果を提示しない場合，より適切なクエリを作成するための戦略が必要とな

This article is a technical report without peer review, and its polished and/or extended version may be published elsewhere.
Copyright ©2022 by IEICE

る．その戦略は大きく分けて，1) 欠陥箇所と無関係である単語を特定しそれをノイズと見做して削除する**クエリ削減**戦略，2) バグレポート内に必要な単語が欠如していると仮定し観測された問題に関連する単語をクエリに挿入する**クエリ拡張**戦略の 2 つに分類される．2 つの戦略を併用する手法も存在する．

Chaparro と Marcus [11] はクエリ中から無関係の単語を 6 つ削除するだけで，欠陥箇所検索の検索結果の性能が向上することを示した．この研究でノイズ単語の影響の大きさが示されクエリ削減の研究が盛んに行われてきた．

クエリ削減の研究では，バグレポートの構成要素を考慮する手法が採用され，クエリ中から Observed Behaviors (以降 OB) 以外の要素由来の単語を人手で削除することでも，欠陥箇所検索の検索結果の性能が向上することが報告 [7] されている．次に，Chaparro ら [8] はバグレポート中のタイトル・OB・Expected Behaviors(以降 EB)・Steps to Reproduce(以降 S2R) のみを利用してクエリを作成することで，欠陥箇所検索の検索結果の性能が向上することを示した．Chaparro らはこの研究で，バグレポートをタイトル・OB・EB・S2R・コードブロック・Other に分割し，それらの構成要素のどの組み合わせが性能の向上をもたらすか調査した．その結果，コードブロックの利用は検索結果の性能の低下をもたらす傾向にあり，タイトル・OB・EB・S2R の利用が推奨されることを報告している．

クエリ拡張戦略の代表的な手法に BLIZZARD [12] がある．BLIZZARD はバグレポートにスタックトレースやプログラム断片が含まれるかどうかを判定し，それぞれの状況に基づいてクエリの拡張を行う．

双方の戦略を採用するものとして，ソフトウェアの振る舞いの記述を表す「Task 属性」[13] を考慮してクエリ拡張を行うことで，クエリを改善する手法が提案されている [9]．Florez らによるこの手法では，先行研究と同様にバグレポートをタイトル・OB・EB・S2R・コードブロック・Other の構成要素に分割し複数の組み合わせでの効果を比較している．実験の結果，コードブロックと Other 内の Task 属性を持たない単語を削除してから BLIZZARD を適用すると検索性能が上がることを示している．

このクエリ削減戦略の潜在的な有効性を明らかにするために，Mills ら [10] は正解から逆算することで模範クエリを構成し，理想的なクエリ削減がなされた場合の性能を分析した．実験の結果，模範クエリは既存の研究と比べて検索結果の性能を大幅に改善し，IR 技術を使用した欠陥箇所検索の実用性を示した．

Mills らは模範クエリを生成した後，興味深い研究を実施しており，構成する単語の由来に特徴がないかを調査している．調査の結果，タイトルと本文で比較した場合，模範クエリを構成する単語の分布に偏りがないことから，先行研究 [8] の主張である，タイトルが重要であるという主張が否定されている．また，Chaparro らの先行研究 [7,8] を評価するため，バグレポートの構成要素と模範クエリの構成の関連性をクエリを構成する単語の種類に着目して調査している．その結果，通常の検索クエリを構成する単語の種類の分布と，模範クエリを構成する単語の種類の分布に変化がなかった．このことは，タイトル・OB・EB・S2R を利用する Chaparro らの手法は，バグレポートをそのまま検索クエリに用いるより精度を向上させているものの，最適な検索性能を示す模範クエリを構成するには不十分なクエリ削減であることを示唆している．

本研究は，この Mills らの調査での次の 2 つの課題を扱っている点が特徴である．まず，Mills らの分析は，模範クエリを構成する単語の探索の実現性を評価するのに不十分である．Mills らは最適化前の検索クエリを構成する単語の種類の集合と，模範クエリを構成する単語の種類の集合のみに着目し比較している．これは模範クエリの作成へ貢献しないと言う結論を導出するのには十分であるが，個別の構成要素の有用性については評価していない．たとえば，個別の構成要素における模範クエリに貢献する単語の出現しやさなどは考慮しない．

また，Mills らは模範クエリへの分析は，単語の由来が，1) バグレポートのタイトルか・本文か, 2) EB, §2R, OB, タイトルか の 2 つの視点でしか行われておらず，例えば，自然言語処理の観点といったクエリの単語の詳細な特徴に基づいた分析はなされていない．筆者らの調べた限り，自然言語処理によって得られる構文上の情報と有用な検索クエリとの関連性を調べる研究はこれまで行われていない．

3. 予備調査：模範クエリの持つ特徴の分析

3.1 データセット

本研究では Mills らの研究 [10] で作成された模範クエリデータセットのサブセットを利用する．このデータセットは 15 のプロジェクトから収集した合計 1,037 のバグレポートから構成されている．その内ソースコード中の欠陥の修正が紐づけられた 815 件のバグレポートに対して模範クエリが生成されている．

Mills らは最適の基準を複数用意しため，バグレポート毎に最大で 30 個，全体合計 20,680 個の模範クエリが存在する．本研究では，最も優れた模範クエリを正解データとするために，検索対象が最も高い順位となった一部のクエリのみを利用する．また，データセットを分析したところ，いずれの構成要素もマークアップされていないバグレポートが多く確認された．そのため，本研究ではデータセットの一部を理想環境として抽出して二種類のデータセットを構築した．

- **理想環境**：バグレポートに OB・EB・S2R・Other のいずれかのマークアップを含むペア
- **一般環境**：全てのバグレポートと模範クエリのペア

表 1 に本研究で用いる 2 種類のデータセット概要を示す．Birt と Tomcat ではマークアップされたバグレポートが存在しなかったため，理想環境には 13 プロジェクトのバグレポートのみが含まれる．

3.2 模範クエリとバグレポートの対応付け

模範クエリとバグレポートの対応の計算は次の 3 段階を経て行う．

（1） バグレポートから一般的な方法で検索クエリを生成
（2） 上記の検索クエリと模範クエリの対応関係を計算
（3） 模範クエリの単語とバグレポートの単語の紐付け

バグレポートに対して，ストップワードの除去や，複合語の分

表 1　本研究で用いた模範クエリデータセット．#BR はバグレポート数，#GAQ は本研究で使用した模範クエリの数

	一般		理想	
プロジェクト名	#BR	#GAQ	#BR	#GAQ
AspectJ	185	1,223	49	265
Birt	100	435	0	–
BookKeeper	24	232	3	30
Derby	50	500	15	150
JodaTime	7	55	4	25
Lucene	32	293	10	91
Mahout	25	234	6	60
OpenJpa	16	134	4	29
Pig	36	266	3	22
SWT	85	772	17	141
Solr	45	450	14	140
Tika	21	201	5	50
Tomcat	98	775	0	–
ZXing	13	117	3	30
ZooKeeper	78	780	17	170
合計	815	6,467	150	1,203

割，語幹化など一般的な検索クエリ構成法を適用したものを媒介にすることで，模範クエリの単語と バグレポートの単語を対応付ける．

通常の検索クエリと模範クエリの対応関係の計算はクエリ単語に関する編集距離 [14] を計算することで行う．編集距離を動的計画法を用いて計算する際に対応関係を記録することで単語間の対応関係を得る．本研究では，単語の削除コストは 1，追加コストは 200，置換コストは 5 で計算を行った．

得られた対応付けを元に，Mills らが手動で精査したバグレポートの構成要素に関するマークアップ情報を付与する．

本研究ではバグレポート中の文章に対して，クエリ作成の手順に合わせて空白区切りによる文分割のルールを用い文章を分割する．自然言語的特徴を分析するために，自然言語解析ツール Stanza [15] の Depparse モジュール，POS モジュールなどを利用し文単位で処理を行う．Stanza には既存の機械学習モデルを利用して文章を文に分割する仕組みがあるが，プログラム中の識別子などが頻繁に登場するバグレポートにはうまく適用できなかった．

形態素解析により文は木構造で表され，各単語には品詞情報に加え，親の単語の情報・親との関係が付与される．この関係から，各単語が主格・目的格・副詞節・形容詞節などの構造に属することが解析できる．

最後に，バグレポートの単語が Task 属性 [13] を持つかどうかを判定するするために，全ての文を Task Phrase Extraction(注1) に適用する．当該サービスは入力された文中に含まれる Task 属性を含む箇所を提示する機能を持つ．本研究では，この出力結果を解析することで各単語に Task 属性の有無を付与する．

3.3　調査 1: バグレポート構成要素と模範クエリの関係

前節で述べたように，Mills らの分析では単語集合を形成して分析をしているため，複数の構成要素に単語が出現している場合，構成要素の有用性を正確に分析できない．本研究では，前述のように前処理を行ったデータセットを用いて単語の出現個所を考慮した対応関係を計算し，「RQ1:模範クエリを構成する単語は，バグレポートのどの構成要素に含まれやすいか」を明らかにする．

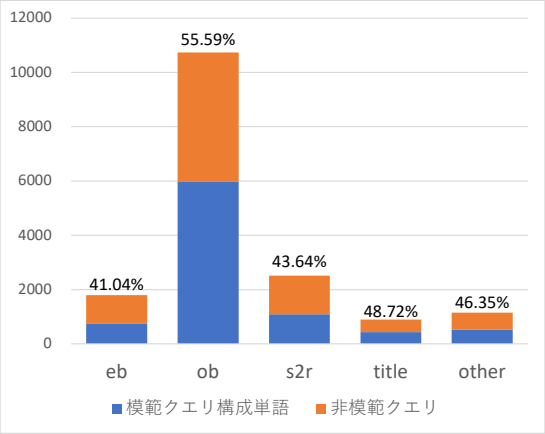

図 1　構成要素毎の単語出現数と模範クエリ単語であった割合

この調査のためには，自然言語で記述されたバグレポートから，EB, OB, S2R など等の構成要素に相当する個所を特定したうえで，模範クエリのそれぞれの単語が対応する箇所を特定する必要がある．

この分析には，理想環境のデータセットに含まれる 150 のバグレポートに対応する最も単語数が多い模範クエリを抽出し，合計 150 ペアのデータを対象として調査を行う．

3.4　調査 2: 自然言語としての特徴と模範クエリの関係

自然言語としての特徴には多くの候補があるが，本研究では，品詞情報・構文情報・Task 属性に注目し，それぞれ対応する RQ2-1 〜 RQ2-3 の 3 つサブ RQ を設定する．それらの答えをもって「RQ2：模範クエリの自然言語的な観点での特徴とは何か」に回答する．この分析には，一般環境のデータセットに含まれるバグレポートに対応する最も単語数が多い模範クエリを抽出し，合計 815 ペアのデータを対象として調査を行う．

まず，「RQ2-1：品詞情報はクエリの最適化に有用か」へ回答するため，検索クエリと模範クエリに出現した各単語の品詞情報を集計する．バグレポート・模範クエリの単語の対応を計算し，単語の品詞情報を付与して集計を行う．次に「RQ2-2：文の構造はクエリの最適化に有用か」へ回答するため，基本的な英文構造に基づく解析を行う．模範クエリの単語が出現しやすい文構造を調べると同時に，模範クエリに貢献する単語の種類を多く含む文構造がないか調査する．最後に「RQ2-3：Task 属性はクエリの最適化に有用か」へ回答するため，Task 属性の分析を行う．バグレポートに Task 属性を付与した後，計算されたバグレポート・模範クエリの単語の対応を利用し，集計をする．

(注1)：http://task-phrases.herokuapp.com

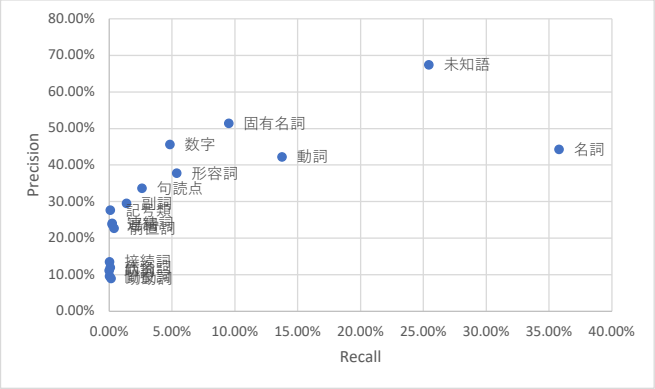

図 2　模範クエリ構成単語に対する品詞毎の Precision/Recall

4.　分析結果と RQ への回答

4.1　RQ1:模範クエリを構成する単語は，バグレポートのどの構成要素に含まれやすいか

図 1 に理想環境の 150 ペアでの構成要素毎の単語出現数の分析結果を示す．分析の結果，OB は他の構成要素と比べて顕著に多い約 68 % の 模範クエリの単語を含むことが分かった．また，OB 内の単語は，56% が模範クエリに含まれる結果となり，これは，何もしない時よりやや高い事も分かった．一方で，その他の構成要素に含まれる模範クエリの単語は全体の分布と同程度であり，クエリ削減の基準には適さないことが分かった．以上のことから RQ1 には次のように回答する．

RQ1 への回答

構成要素としては OB に出現する単語が模範クエリを構成しやすく，過半数以上の模範クエリの単語が発見できる．残りの構成要素は模範クエリの構成に強く貢献しない．

4.2　RQ2:模範クエリの自然言語的な観点での特徴とは何か

4.2.1　RQ2-1:品詞情報はクエリの最適化に有用か

模範クエリ構成単語かどうか識別する問題と捉え，各品詞毎の有用性を Precision, Recall で評価結果を図 2 に示す．

図より固有名詞と未知語は特に Precision が高く，模範クエリを構成する際に選択基準となることが分かる．分析の結果，アルゴリズム名・クラス名・メソッド名がこれらの品詞に分類されていることを確認した．これらの単語は Kocchar ら [16] が指摘する検索のヒントである可能性が高く，バグレポートに含まれる重要な情報が品詞の観点で発見できることを示していると考える．

名詞と動詞の Precision は 40%程度であるが，それぞれ 35%，約 14%の Recall があり，模範クエリ構成単語の約半分がこれらに分類される．一方で，上記の品詞を除く，助動詞・助詞・前置詞などは Precision, Recall 共に低いことが分かる．文法上形式的に出現するこれらの品詞は，検索にはあまり貢献せずノイズとなる．このことから，これらの品詞を持つ単語は模範クエリの構成に寄与しない単語として削除するべきことが示唆される．以上のことから，RQ2-1 へは以下のように回答する．

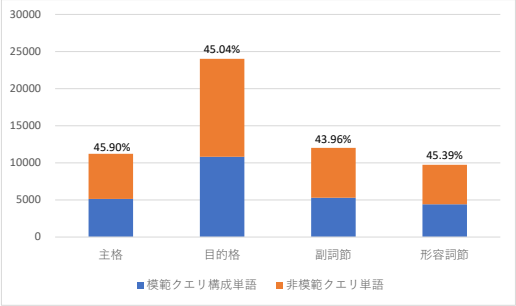

図 3　文の構造毎の単語出現数と模範クエリ単語である割合

RQ2-1 への回答

品詞情報はクエリ最適化に有用である．文法上形式的に使われやすい単語の品詞は，模範クエリの構成に顕著に貢献しない．一方，出現する模範クエリの単語の半分は名詞や動詞を由来とする．未知語や固有名詞は特に重要である．

4.2.2　RQ2-2: 文の構造はクエリの最適化に有用か

まずは，単語の出現数に基づく分析を行う．図 3 より，どの形態素解析の属性を使用しても，模範クエリを構成する単語である割合に差がないことがわかる．しかし，模範クエリの構成単語の割合において，目的格は他の構造の 2 倍あることがわかる．目的格には模範クエリへ貢献する単語の約 1/3 が含まれており，大きな割合を占めている．

RQ2-2 への回答

単語の出現数で分析した場合，各形態素解析の要素から得られる模範クエリに貢献する単語の割合は変わらない．しかし，目的格に含まれる模範クエリに貢献する単語の割合は多い．単語の種類からも同様な傾向が得られる．

4.2.3　RQ2-3: Task 属性はクエリの最適化に有用か

表 2 に Task 属性を持つ単語のバグレポート全体での出現数と模範クエリ中での出現数，Task 属性を持つことを条件とした場合のクエリ削減の精度を示す．比較のために何も削減しない場合の精度も掲載している．表から，Task 属性のみで模範クエリを構成すると Precision が 59.2% となり，全体での Precision 46.68% よりも高い結果となった．これは，Task 属性を持つ単語を選択することは，何も条件を考慮しない時よりも模範クエリを構成する単語を選択しやすいことを示している．一方で， Recall は約 15% であり，Task 属性のみでは，模範クエリを再現することができないことを示している．以上のことから本研究では RQ2-3 に次のように回答する．

RQ2-3 への回答

Task 属性は模範クエリを構成する単語の発見に貢献するが，模範クエリの一部しか再現できない．

5.　提 案 手 法

本研究では, RQ1, RQ2 で得られた知見をもとに，Mills らが

表 2　Task 属性を持つ単語の出現数と単語の模範クエリへの貢献

	模範クエリ構成数	Precision	Recall	F_1	出現数	全体における分布
Task 属性	4202	59.20%	14.27%	0.2300	7098	11.26%
全体	29440	46.68%	100.00%	0.6365	63063	100.00%

正解から逆算して求めた 模範クエリに近い検索クエリを構成するクエリ削減手法を提案する．

前節の結果より，まず，RQ1 で明らかにしたように重要な単語の大半は OB に含まれるため，バグレポートの構成要素の情報が利用できる場合は OB の記述をクエリに採用する．

本研究では，クエリ削減の戦略として，できるだけ不要な単語を含まずに模範クエリを構成する単語を選択する方法を採用する．これは，情報検索技術の特性上，検索に有用な単語をクエリに含んでいたとしても，不要な単語を多く含むことで検索順位が著しく下がることが予想できるためである．このことは，既存研究での実験結果から，バグレポートから特定の部分を抽出することが検索精度向上に寄与することでも示されている．

OB ではない構成要素から単語を抽出する他の指標として，RQ2 より得られた傾向を利用する．まず，RQ2-1 より，動詞・名詞・固有名詞といった特定の品詞は，模範クエリを構成する単語となる傾向が高いことを考慮しこれらの品詞を選択する．RQ2-2 より目的格は他の言語構造よりも 模範クエリを構成しやすいため，目的格に含まれる単語も選択する．このほかに，RQ2-3 で示したように，Task 属性を持って出現する単語は，再現度合いは低いが模範クエリに貢献しやすいため，これらも検索クエリに含める．最後に，RQ2-1 の結果に基づき，模範クエリの構成要素になりづらい品詞の単語を除去する．

最終的な提案手法による検索クエリは，バグレポート中の単語集合の ({OB}∪{ 目的格 }∪{ 有要品詞 }∪{Task 属性 })\{ 不要品詞 } となる部分集合となる．なおここでの**有要品詞**は，動詞・名詞・固有名詞を，**不要品詞**は 助動詞・記号類・間投詞・前置詞・冠詞・句読点・副詞・接続詞・代名詞・連結詞・助詞を指す．

それぞれの特徴は完全に直交するものではないため各基準の適用方法についてはいくつかの組み合わせが存在する．本研究ではその中でも一番多くの単語を選択するようこれらの和集合を検索クエリとすることを採用している．

6. 実験と評価

6.1 実験方法

提案手法は既存手法より模範クエリに近い検索クエリを構成するかを評価するために，Florez の手法と提案手法を理想環境の 150 バグレポート (1,203 模範クエリ) と一般環境の 815 バグレポート (6,467 模範クエリ) に適用し，模範クエリとの一致度合いを計測する．3. 節 で示したように，バグレポートによっては最も精度が高い模範クエリが複数存在するためバグレポートの数と模範クエリの数は一致しないことに注意されたい．本実験では，正解として利用する模範クエリは各バグレポートごとに最も精度が高い模範クエリの全て利用しそれぞれの一致度を計算する．一致の度合いは，Precision と Recall を計測し，Precision と Recall の調和平均である F_1 の指標で評価する．

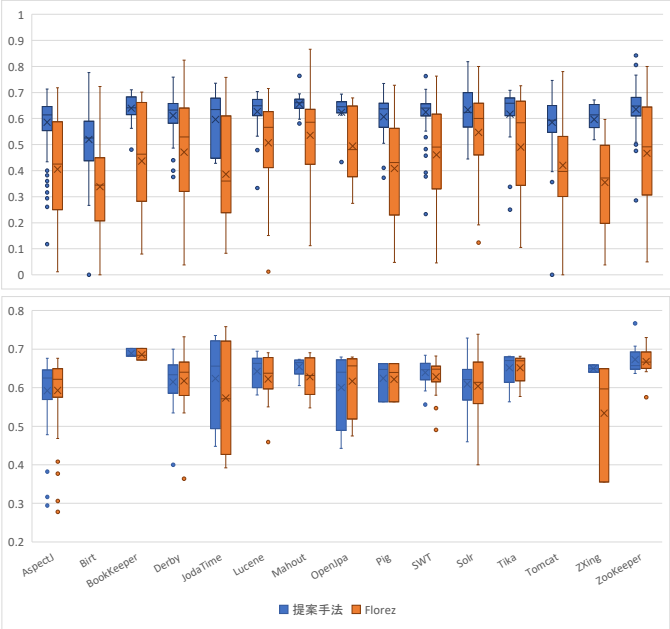

図 4　提案手法と既存手法 [9] のプロジェクト毎の F_1 の分布．上は一般環境，下は理想環境の結果

6.2 実験結果

実験の結果を図 4 に示す．理想環境においては提案手法のクエリの方の F_1 の平均値が 13 プロジェクト中 9 プロジェクトで高かった．Wilcoxon の符号順位和検定の片側検定をした結果，このうちの 6 プロジェクト (Bookkeepr, Lucene, Mahout, SWT, ZXing, ZooKeeper) に，Derby, OpenJpa を加えた 8 プロジェクトで，$p < 0.05$ で統計的に有意に優れることが分かった．このことは，Florez らの手法では S2R や EB といった構成要素中の単語に含まれる不要な単語の影響が大きいこと，Task 属性を持つ単語列だけでは十分に正解が選択出来ていないことを示すものと考える．提案手法は，広く目的格を構成する単語を選択しており，さらに品詞種別による不要な単語の除去を行うことで正解を多く選択しつつ不要な単語を除去しており，このことが模範クエリの再現の成功に寄与したものと考える．

また，一般環境においては，全てのプロジェクトで提案手法のクエリの方の F_1 の平均値が高かった．図 4 より提案手法のクエリの F_1 の分布が Florez らより高いことがわかる．Florez らの手法は必要なバグレポートの構成要素が存在することに依存する．しかし，提案手法は S2R や EB といった構成要素を持たなくても，自然言語の特徴を利用できるため，より性能を発揮することができる．

以上のことから，本研究では RQ3 に対して次のように回答

する．

RQ3 への回答

提案手法は品質の悪いバグレポートが散逸する一般的な環境でも適用可能であり，先行研究の手法よりも模範クエリの再現性が高い．品質の良いバグレポートが得られる理想的環境において，提案手法は模範クエリの再現性能をさらに向上させる．

7. 妥当性への脅威

7.1 内的妥当性への脅威

7.1.1 バグレポートと模範クエリの対応の前処理

バグレポートと模範クエリを編集距離に基づいて対応を計算するとき，単語同士が不適切に対応を取る可能性がある．本研究ではテストケースを数種類用意し，妥当な対応を構築できるか確認した．また，対応された単語を 100 件抽出し，妥当な対応を構築していることを確認した．しかし，不適切な対応が発生した可能性は残る．

7.1.2 バグレポートへの既存の自然言語処理ツールの適応

自然言語処理ツールは，通常の自然言語を前提にしており，バグレポートのようにクラス名やソースコードを含むことを想定していない．そこで本研究では，クラス名やソースコードなど，既存の言語モデルに嵌らない単語を「未知」として検出できる UPOS の品詞情報を利用した．それを前提に形態素解析を行える Stanza をツールとして利用した．しかし，未知語を利用した形態素解析はまだ難しく，文の木構造の中で未知語を含む枝の精度が低くなる可能性は否定できない．

7.2 外的妥当性への脅威

データセットの範囲が狭いことによる一般化への脅威が存在する．本研究で取り扱ったデータセットは 15 の OSS から収集したものであるため，一つの OSS プロジェクトの影響を大きく受けることは考えづらい．しかし，Java 言語の OSS プロジェクトに限定したため，他言語を扱ったプロジェクトや商用プロジェクトに結論が当てはまらない可能性が残る．

7.3 構造的妥当性への脅威

本研究では模範クエリの再現に研究の目的を据え，提案手法の評価を模範クエリへの近さで行った．しかしながら，模範クエリの再現度合いはクエリ削減の性能を保証しない．模範クエリを正解として Precision と Recall を計算し，その調和平均である F_1 を評価指標として用いることで，過不足なく模範クエリを構成することを評価したが，実際に欠陥箇所検索を行う場合，模範クエリにより近い検索クエリが必ずしも検索性能が高くない可能性がある．

8. おわりに

本研究では，情報検索技術に基づく欠陥箇所検索手法の根幹となるクエリ削減手法の改善を行った．Mills らの模範クエリとバグレポートの構成要素との対応関係を詳細に分析することで，模範クエリを構成する単語の自然言語的特徴を明らかにし，得られた知見に基づき新しい戦略でクエリ削減手法を提案した．提案手法では，OB に相当する記述と目的格や Task 属性をもつ箇所に含まれる単語およびいくつかの品詞を選択した後，模範クエリに登場しづらい品詞を除去する戦略を採用した．15 の OSS プロジェクトから集めた 6,467 個の模範クエリを正解とするクエリ削減実験により，提案手法は既存手法よりも模範クエリに近いクエリ削減が可能であることを示した．

今後の課題としては，まず提案手法が構成する検索クエリの検索精度について評価することが挙げられる．提案手法はより模範クエリを再現するが，それはクエリの検索性能を必ずしも保証しない．生成される各種クエリが実環境でどのような性能を示すか調査する必要がある．また，本研究では得られた知見・法則を全て組み合わせて，提案手法を構築した．しかし，組み合わせの仕方により，より模範クエリの性能が向上する可能性はある．利用する法則の取捨選択した場合の影響に関する調査が今後必要である．

謝辞： 本研究の一部は科研費（#18H03221）の助成を受けた．

文　献

[1] M. Soltani, et al. The significance of bug report elements. *Empir. Softw. Eng.*, Vol. 25, No. 6, 2020.
[2] B. Dit, et al. Feature location in source code: A taxonomy and survey. *J. Softw. Evol. and Proc.*,, Vol. 25, No. 1, 2013.
[3] J. Zhou, et al. Where should the bugs be fixed? more accurate information retrieval-based bug localization based on bug reports. In *Proc. ICSE*, 2012.
[4] L. Moreno, et al. On the use of stack traces to improve text retrieval-based bug localization. In *Proc. ICSME*, 2014.
[5] Shayan A. Akbar and Avinash C. Kak. A large-scale comparative evaluation of ir-based tools for bug localization. In *Proc. MSR*, 2020.
[6] S. Haiduc, et al. Automatic query reformulations for text retrieval in software engineering. In *Proc. ICSE*, 2013.
[7] O. Chaparro, et al. Using observed behavior to reformulate queries during text retrieval-based bug localization. In *Proc. ICSME*, 2017.
[8] O. Chaparro, et al. Using bug descriptions to reformulate queries during text-retrieval-based bug localization. *Empir. Softw. Eng.*, Vol. 24, No. 5, 2019.
[9] J. M. Florez, et al. Combining query reduction and expansion for text-retrieval-based bug localization. In *Proc. SANER*, 2021.
[10] C. Mills, et al. On the relationship between bug reports and queries for text retrieval-based bug localization. *Empir. Softw. Eng.*, Vol. 25, No. 5, 2020.
[11] O. Chaparro and A. Marcus. On the reduction of verbose queries in text retrieval based software maintenance. In *Proc. ICSE*, 2016.
[12] M. M. Rahman and C. K. Roy. Improving ir-based bug localization with context-aware query reformulation. In *Proc. FSE*, 2018.
[13] C. Treude, et al. Extracting development tasks to navigate software documentation. *IEEE TSE*, Vol. 41, No. 6, 2015.
[14] D Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, 1997.
[15] P. Qi, et al. Stanza: A Python natural language processing toolkit for many human languages. In *Proc. ACL*, 2020.
[16] P. Kochhar, et al. Potential biases in bug localization: Do they matter? In *Proc. ASE*, 2014.