

# 単語分散表現による類義語統一と単語 N-gram によるフレーズ抽出に基づくセキュリティ要件分類手法

宮崎 智己<sup>1,a)</sup> 東 裕之輔<sup>1,2,b)</sup> 大平 雅雄<sup>1,c)</sup>

受付日 2021年4月13日, 採録日 2021年10月8日

**概要:** 見落とされやすい非機能要件の特定を支援するために、非機能要件分類手法が提案されている。しかしながら、既存手法には (1) 単語の類似性を考慮しておらず類似した単語で構成される要件文を異なる種類の品質特性を有する要件文と扱うことがある、(2) 単語の順序関係を考慮しておらず同じ単語で構成されるが異なる意味を持つフレーズ (異なる品質特性を有する要件文) を同じものとして扱うことがある、という問題点がある。本研究では、単語の類似性と単語の順序関係を考慮することで、非機能要件分類手法の精度向上を目指す。単語の類似性を考慮するために、(1-A) シソーラスを用いた類義語の統一、(1-B) 単語分散表現を用いた類義語の統一の 2 つのアプローチをとる。単語の順序関係を考慮するために、(2-A) 固定長の単語 N-gram をフレーズとして抽出する場合と (2-B) 可変長の単語 N-gram ( $N \geq 1$ ) をフレーズとして抽出する場合の 2 つのアプローチをとる。非機能要件のうち、セキュリティ要件の分類を対象とした評価実験の結果、提案手法は既存手法よりも精度が高くなることを確認した。また、教師ありアルゴリズムと分類対象のセキュリティ要件によって、精度が最も高い類義語統一処理とフレーズ抽出の組み合わせが異なることが分かった。

キーワード: 非機能要件, 自動分類, 機械学習

## Classification of Security Requirements Based on Unification of Synonyms Using Word Embedding and Extraction of Phrases Using N-gram

TOMOKI MIYAZAKI<sup>1,a)</sup> YUNOSUKE HIGASHI<sup>1,2,b)</sup> MASAO OHIRA<sup>1,c)</sup>

Received: April 13, 2021, Accepted: October 8, 2021

**Abstract:** In order to identify easily-overlooked nonfunctional requirements, many automated classification methods for nonfunctional requirements have been proposed. Existing methods, however, have common shortcomings: (1) they does not consider similarities among words used in requirement documents and sometimes classify requirements described using similar words into different classes, and (2) they also does not consider order relations of words used in requirement documents and they sometimes classify requirements using different phrases with the same words into the same class. The goal of our study is to improve the classification accuracy by considering similarities and order relations among words used in requirement documents. Our method unifies synonyms in requirement documents by using (1-A) a thesaurus (SEThesaurus) and (1-B) word embeddings (Word2vec, fastText, and SO\_Word2vec) to resolve the issue of similarities among words used in requirement documents. It also extracts phrases of (1) fixed-length word N-grams and (2) variable-length word N-gram ( $N \geq 1$ ) to resolve the issue of order relations of words used in requirement documents. As a result of a security classification experiment, our method outperformed the existing method and showed a combination of the synonym unification and the phrase extraction is different from properties of security requirements.

**Keywords:** nonfunctional requirement, automated classification, machine learning

<sup>1</sup> 和歌山大学大学院システム工学研究科  
Graduate School of Systems Engineering, Wakayama University, Wakayama 640-8510, Japan

<sup>2</sup> 株式会社日本総合研究所  
The Japan Research Institute, Limited, Shinagawa, Tokyo 141-0022, Japan

a) miyazaki.tomoki@g.wakayama-u.jp  
b) higashi.yunosuke@g.wakayama-u.jp  
c) masao@wakayama-u.ac.jp

## 1. まえがき

情報システムは、業務内容や利用環境によって重要視される品質特性<sup>\*1</sup>が異なる。たとえば、自動運転システムでは機能適合性が特に重要視されるが、金融系のシステムでは情報漏えいが発生しないようにセキュリティが特に重要視される。開発者は開発するシステムごとに重要視される品質特性を正しく理解しておく必要がある。

開発者は設計・実装を行うために要件定義書から機能・非機能要件を手作業で特定する。しかしながら、非機能要件の多くは機能要件にも該当する [1]、非機能要件は機能要件と比べて曖昧な表現になりやすい [2]、同じ内容の要件文でも開発者によって表現が異なる場合がある、といった理由により、非機能要件は機能要件と比べ見落とされやすいことが知られている。非機能要件の見落としは開発工程の手戻りやシステム障害の原因となる。また、システムが大規模化・複雑化するにつれ求められる要件も多くなり、大量の要件から非機能要件をすべてを把握するのが困難になる。

見落とされやすい非機能要件の特定を支援するために、機械学習を用いた非機能要件分類手法が提案されている [3], [4], [5], [6], [7], [8]。しかしながら既存手法は、(1) 単語の類似性を考慮しておらず類似した単語で構成される要件文を異なる種類の品質特性を有する要件文と扱うことがある、(2) 単語の順序関係を考慮しておらず同じ単語で構成されるが異なる意味を持つフレーズ（異なる品質特性を有する要件文）を同じものとして扱うことがある、という2つの問題点がある。これら2つの問題点を改善することで、非機能要件分類の精度を向上できる可能性がある。

本研究では、単語の類似性と単語の順序関係を考慮することで、非機能要件分類手法の精度向上を目指す。単語の類似性を考慮するために、(1-A) シソーラスを用いた類義語の統一、(1-B) 単語分散表現を用いた類義語の統一の2つのアプローチをとる。単語の順序関係を考慮するために、単語 N-gram によるフレーズ抽出を行う。単語 N-gram を用いたフレーズ抽出には、(2-A) 固定長の単語 N-gram をフレーズとして抽出する場合と (2-B) 可変長の単語 N-gram ( $N \geq 1$ ) をフレーズとして抽出する場合の2つのアプローチをとる。

本提案手法は特定の品質特性の分類に特化するものではないが、情報システムの普及にともないセキュリティ侵害のリスクに大きな関心があること [9] をふまえ、特に本稿ではセキュリティ要件の分類精度を既存手法よりも向上できるかどうかを検証する。以降ではまず、2章において既存手法を説明し本研究の位置付けを示す。3章で提案手法について述べる。4章で提案手法を評価するために行った

評価実験について説明し、5章で評価実験の結果を報告する。6章で実験結果についての考察を行い、7章で本稿をまとめる。

## 2. 先行研究

見落とされやすい非機能要件の特定を支援するための自動分類手法が提案されている。本章ではまず、既存の非機能要件分類手法について紹介する。次に、既存手法の問題点を示し本研究の位置付けを説明する。

### 2.1 非機能要件分類手法

近年、機械学習を用いた非機能要件分類手法が提案されている [4], [5], [6], [7], [8]。教師あり学習、半教師あり学習、教師なし学習それぞれ用いた手法が提案されているが、本研究では教師あり学習を用いた手法に着目する。

Zhang らは SVM を用いた非機能要件分類手法を提案している [4]。分類に用いる特徴量に、単語のみ、文字 N-gram、形容詞・名詞・前置詞から構成される複合語、の3つのパターンを設けて実験を行った結果、単語を特徴量を用いた場合の分類精度が最も高かった。

Slankas らは Naïve Bayes, SMO, k 近傍法を用いた非機能要件分類手法を提案している [5]。Naïve Bayes と SMO を用いる場合、要件文中に出現する単語を特徴量とした BoW ベクトルを用いて分類器を構築している。k 近傍法を用いる場合、要件文間の単語の編集回数（挿入・削除・置換）を用いて分類器を構築している。評価実験の結果、SMO を用いて構築した分類器が最も高い精度を示した。

Riaz らはセキュリティ要件の詳細な分類を目的とした分類手法を提案している [7]。既存手法 [5] と同様に、要件文中に出現する単語を特徴量とした BoW ベクトルを用いて構築した SMO に基づく分類器の分類精度が高い (Precision : 0.8, Recall : 0.76, F 値 : 0.78) ことを示している。

なお、特に近年では、ANN (Artificial Neural Network) や CNN (Convolutional Neural Network) を用いた手法 [10] や BERT (Bidirectional Encoder Representations from Transformers) [11] を用いた手法 [12] なども提案されている。これらの手法は機能・非機能要件の分類が目的でありセキュリティ要件の詳細な分類を行う本研究とは分類の粒度が異なること、また、事前学習に用いられるデータセットが本研究とはまったく異なることから本稿では比較対象外とするが、将来的には検討の余地がある。

### 2.2 既存手法の問題点

BoW ベクトルを用いた既存手法 [5], [7] は、高い分類精度を示したが以下で説明する2つの問題がある。

1つ目の問題は、既存手法は単語の類似性を考慮していないため、同じ内容ではあるが類似した単語で構成される要件

<sup>\*1</sup> ISO/IEC 25010: <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

文を誤分類してしまう場合がある点である。“The system shall allow authenticated users to access to data.”と“The system shall allow authorized users to access to data.”という文を例に説明する。これらの要件文は“authenticated”と“authorized”が異なるが、どちらの要件文も特定のユーザに対してデータへのアクセスを許可することを意味する機密性に該当する要件文である。しかしながら、既存手法では“authenticated”と“authorized”の類似性を考慮しないため、既存手法で前者の要件文のみを機密性として学習した場合、後者の要件文を機密性以外であると誤分類してしまう。

2つ目の問題は、BoW ベクトルは単語の順序関係を考慮していないため、同じ単語で構成されるが異なる意味を持つフレーズを同じものとして扱う点である。たとえば、“log in”と“in a log”というフレーズがあった場合、どちらも“log”と“in”を含んでいるが順序関係が異なるため意味は異なる。しかしながら、BoW ベクトルは単語の出現回数の特徴量に用いるため、頻出する“log”と“in”を含む“log in”と“in a log”といったフレーズを同様のものとして分類してしまう。

本研究では、単語の類似性と単語の順序関係を考慮することでBoW ベクトルを用いた既存手法の問題点を解消し、非機能要件分類手法の精度向上を目指す。

### 3. 提案手法

本章では単語の類似性と単語の順序関係を考慮したセキュリティ要件分類手法について説明する。提案手法は5つのステップで構成される。まず、第1ステップでは、前処理として要求文をクリーニングし単語に分割する。第2ステップでは、シソーラスを用いた類義語の統一を行う。次に、第3ステップで分散表現手法を用いた類義語の統一を行う。第4ステップで、単語 N-gram を用いたフレーズ抽出を行い、抽出した N-gram を特徴量とした特徴ベクトルを作成する。最後に第5ステップで分類器を構築する。それぞれのステップについて以降で詳しく説明する。

#### 3.1 前処理

まず、要件文のクリーニングを行う。要件文中のピリオドやカンマ、箇条書きの記号など、分類のノイズとなる文字を正規表現を用いて除去する。また、URL も記号と同様に分類のノイズとなってしまうため、正規表現を用いてすべて同じ文字列 (“URL”) に統一する。次に、“Author”と“author”のような表記ゆれを解消するために文字をすべて小文字に変換する。

クリーニング後の要件文に対して形態素解析を行う。数字を含む固有名詞 (“sha1” など) が一部あるため、すべての数字ではなく形態素解析により分割された単語が数字のみの場合に限り数字をすべて同じ文字列 (“suji”) に統一

する。次に、レマタイズ処理を行い単語の正規化を行う。最後に、分類に重要ではないストップワードを除去する。本手法では nltk \*2 が定義しているストップワードを除去の対象とする。ただし、本手法ではイディオムなどのフレーズの抽出を前提にしているため、nltk が定義しているストップワードのうち、前置詞と接続詞の単語は除去しない。

#### 3.2 類義語の統一

次に、要件文に含まれる単語のうち、類義語の関係にある単語を統一する。本手法では、まず、(1-A) シソーラスを用いた類義語の統一を行う。次に、(1-B) 分散表現手法を用いた類義語の統一を行う。それぞれの統一方法について詳しく説明する。

##### 3.2.1 (1-A) シソーラスを用いた類義語の統一

本手法ではソフトウェア用語を対象にしたシソーラスである SEThesaurus \*3 [13] を用いて表記ゆれの関係にある単語を同じ表記に統一する。SEThesaurus はソフトウェア用語における略語やタイプミスといった表記ゆれや文字列が類似している単語を類義語として定義している。具体例には、“authorize”と“authorized”、“authenticated”などがあげられる。

##### 3.2.2 (1-B) 分散表現手法を用いた類義語の統一

SEThesaurus は表記ゆれや文字列が類似単語の解消を目的とするが、シソーラスに存在しない単語を類義語として統一することができない。そのため、本手法では単語の分散表現手法を用いて類義語の統一を行う。単語の分散表現は単語を任意の次元のベクトルで表現する手法であり、ベクトル間の類似性を扱うことで単語間の類似性を扱うことが可能である。本手法では Word2vec [14] と fastText [15] の2つの分散表現手法を用いる。

Word2vec はニューラルネットワークを用いた分散表現手法であり、周辺に位置する単語が似ている場合、その単語どうしは類義語の関係であるという考えに基づいたアルゴリズムである。fastText は、Word2vec に文字列が類似している単語は類義語の関係であるという考えを加えたアルゴリズムである。本手法では skip-gram モデルを用いて Word2vec, fastText の学習を行う。学習の際の計算コストの削減のために、出現頻度の低い単語の除去とネガティブサンプリング [14] を行う。また、分散表現の学習には大量のテキストデータが必要であり、本研究で用いる要件文データセットのみでは分散表現手法が十分に学習できない可能性がある。そのため、Stack Overflow の投稿を学習した Word2vec モデルである SO\_Word2vec \*4 [16] も適用の対象とする。

図 1 に分散表現手法を用いた類義語の統一処理の手順を

\*2 <https://www.nltk.org/>

\*3 <https://se-thesaurus.appspot.com/manualcheck>

\*4 <https://zenodo.org/record/1199620#.W-AYcnozaL4>

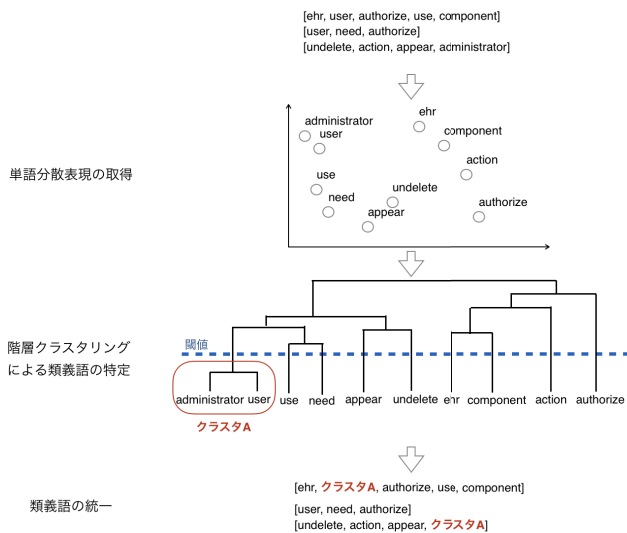


図 1 分散表現手法を用いた類義語の統一

Fig. 1 Unification of synonyms using word embedding.

示す。まず、分散表現手法を要件文に対して適用し、単語分散表現を取得する。次に、取得した単語分散表現に対して階層的クラスタリングを適用し、樹形図を作成する。分散表現間の距離計算にはコサイン距離を用いる。クラスター間の距離計算には外れ値に強い群平均法を用いる。次に、作成したクラスターのうち、閾値以下の距離で構成されるクラスターを類義語と見なして抽出する。抽出したクラスターには図 1 のようにラベル付け (クラスター A) を行う。クラスター抽出後、要件文中に含まれるクラスターの構成単語に該当する単語をクラスターのラベルに置換し単語を統一する。

### 3.3 単語 N-gram を用いたフレーズ抽出による特徴ベクトルの作成

次に、単語の順序関係を考慮するために単語 N-gram を用いたフレーズ抽出を行う。本手法では、(2-A) 固定長の単語 N-gram をフレーズとして抽出する場合と (2-B) 可変長の単語 N-gram ( $N \geq 1$ ) をフレーズとして抽出する場合、の 2 つのアプローチをとる。それぞれのアプローチについて説明する。

#### 3.3.1 (2-A) 固定長の単語 N-gram の抽出

固定長の単語 N-gram を用いてフレーズを抽出する場合、要件文をあらかじめ設定した N-gram に分割する。N-gram では 1-gram の場合では考慮されなかった単語の順序関係を考慮することが可能である。ただし、N の値が大きくなるにつれ抽出する N-gram の種類が膨大になる。本手法では単語 2-gram の場合と単語 3-gram の場合でフレーズ抽出を行い、抽出したフレーズを特徴量に用いて要件文を特徴ベクトルに変換する。

#### 3.3.2 (2-B) 可変長の単語 N-gram ( $N \geq 1$ ) の抽出

固定長の単語 N-gram をフレーズとして抽出する場合、あらかじめ N の値を決定する必要がある。また、単語 N-gram

は単語 1-gram と比べて出現する文が少ないため、特定の要件文にしか該当しない限定的な特徴量となってしまう場合がある。そのため、可変長の単語 N-gram ( $N \geq 1$ ) をフレーズとして抽出する場合を考える。本手法では Multiword Expression Distance (MED) [17] を用いて N-gram を抽出する。MED はコルモゴロフ複雑性に基づいた単語間結合度を計算するアルゴリズムである。MED は式 (1) を用いて計算される。

$$MED(g) = \log \frac{D(w_1, \dots, w_n)}{D(g)} \quad (1)$$

ここで、 $D(g)$  は単語 N-gram  $g$  を含む文書数、 $D(w_1, \dots, w_n)$  は単語 N-gram  $g$  を構成する単語すべてを含む文書数を表す。MED の値が大きいくほど単語間結合度が低く、MED の値が小さいほど単語間結合度が高い N-gram であることを表す。

MED を用いたフレーズ抽出では、まず、要件文に対して MED を適用し N-gram ( $N \geq 2$ ) を抽出する。抽出した N-gram をすべて用いた場合、ベクトルの次元数が増え計算コストが増加してしまうため、四分位範囲を利用した外れ値除去法を用いて、MED 値が外れ値である N-gram は除去する。外れ値除去後に残った N-gram ( $N \geq 2$ ) と 1-gram を特徴量として用いた特徴ベクトルを作成する。

### 3.4 分類器の構築

作成した特徴ベクトルを教師ありアルゴリズムに学習させ分類器を構築する。本手法では既存手法 [7] で用いられている教師あり学習アルゴリズムである Naive Bayes, SMO, SVM を用いる。

## 4. 評価実験

本章では、単語の類似性と順序関係を考慮することがセキュリティ要件分類手法の精度向上に寄与するかどうかを確かめるための評価実験について述べる。

### 4.1 データセット

本評価実験では既存手法の著者らが公開している表 1 のデータセット [7] を用いる。本データセットは 6 つの医療系システムの要件定義書から抽出した要件文であり、6 種類のセキュリティ要件とセキュリティ要件以外 (その他) の合計 7 種類の分類ラベルが割り当てられている。本評価実験では、表 1 の太字で示した要件文を分類の対象とし、それ以外の要件文は除外する。可用性・真正性・プライバシーはそれぞれ数百件程度しかなく、分類器を構築するうえでのデータ量が十分ではないためである。なお、表 2 に示したように、全 10,871 件の要件文のうち、4,640 件がセキュリティ要件に関するものである。また、4,640 件中 3,928 件には 2 つ以上のラベルが付与されている。

表 1 セキュリティ要件データセット [7]  
Table 1 Security requirement dataset [7].

分類ラベル	説明	要件文 (件)
機密性	データが意図した場合のみ開示されるための度合い	3,120
完全性	データやプログラムに対する不適切な修正や破壊を防止するための度合い	3,386
可用性	ユーザがいつでもシステムを利用できるための度合い	174
責任追跡性	ユーザやデータなどの行動を後から追跡できる度合い	3,882
真正性	ユーザやプロセスなどを認識するための度合い	311
プライバシー	ユーザがデータの使用方法を理解し制御するための度合い	245
その他	機能要件およびセキュリティ要件以外の要件文	6,231

太字の要件文：評価実験における分類対象

表 2 マルチラベルの内訳  
Table 2 Detail of multi-labels.

ラベル数	件数 (データセットすべて)	件数 (セキュリティ要件のみ)
1	6,943	712
2	1,542	1,542
3	2,222	2,222
4	164	164
合計	10,871	4,640

## 4.2 実験方法

前述のとおり、評価実験で用いるデータセットには複数のセキュリティ要件に該当する要件文が含まれている。そのため、評価実験で行う分類タスクはマルチラベル分類となる。そこで評価実験では、マルチラベル分類手法の1つである binary relevance [18] を用いて分類精度を確認する。binary relevance とは各ラベルに対してそのラベルか否かの2値分類器を構築し分類を行う手法である。評価実験において、対象とする3種類のセキュリティ要件（機密性・完全性・責任追跡性）に対してそれぞれ分類器を構築し、各セキュリティ要件か否かを判別する。また、分類結果の妥当性を確保するために10分割交差検証を10回行い、合計100回分の分類精度を用いて評価する。実験手順については4.3節で詳しく説明する。分類精度を評価するための指標に、適合率と再現率、適合率と再現率の調和平均であるF値を用いる。本評価実験における適合率とは、セキュリティ要件と分類した要件文のうち、正しく分類できた要件文の割合を示す。再現率とは、セキュリティ要件の要件文のうち、正しく分類できた要件文の割合を示す。

## 4.3 実験手順

評価実験の手順について説明する。

**手順 1：データセットの整形** binary relevance を行うために分類を行うセキュリティ要件を3種類の中から1つ選択し、選択したセキュリティ要件に該当する要件文のラベルを1に、選択したセキュリティ要件に該当しない要件文のラベルを0にする。

**手順 2：データの分割** 10分割交差検証を行うために、データセットの9割を学習データ、残り1割をテストデー

SEThesaurus	分散表現手法	フレーズ抽出
あり	Word2Vec	1-gram
あり	Fasttext	2-gram
あり	SO_Word2vec	3-gram
なし	SO_Word2vec	N-gram

図 2 実験で実施する類義語統一とフレーズ抽出の組合せ

Fig. 2 Combination of unification of synonyms and extraction of phrases performed by our experiment.

タに分割する。

**手順 3：提案手法の適用** 本研究で提案している類義語統一処理とフレーズ抽出の方法はそれぞれ複数あるため、評価実験では類義語統一処理とフレーズ抽出の組合せを行い、セキュリティ要件の分類に最適な組合せを明らかにする。評価実験で実施する類義語統一処理とフレーズ抽出の組合せを図2に示す。

**手順 3.1：シソーラスを用いた類義語の統一** 一般的なソフトウェア開発用語の類義語を統一し、要件文にのみ表れる単語の類義語を分散表現を用いて統一するために、データセット全体にSEThesaurusを適用する。ただし、学習データにSO\_Word2vecを適用する場合には、SEThesaurusを適用する場合と適用しない場合の2通りの条件を設定し、SEThesaurus適用の効果を調べる。SO\_Word2vecはスタックオーバーフローの膨大な投稿から構築した単語分散表現であるため、SEThesaurusを適用しても効果が得られない可能性があるためである。

**手順 3.2：単語分散表現を用いた類義語の統一**

次に、学習データにWord2vecやfastTextを適用する。Word2vecとfastTextのハイパーパラメータは、最も精度が良かった、次元数200、最小出現回数15、周辺単語数10、ネガティブサンプリング5としている。また、クラスタ距離の閾値を0.02~0.18まで0.02ずつ変更し、分類精度への影響を調査する。SO\_Word2vecを用いる場合もWord2vec, fastTextと同様に、クラスタ距離の閾値を0.02~0.18まで0.02ずつ変更し、分類精度への影響を調査する。以上より、SEThesaurusを適用する場合の3種類の単語分散表現の分類精度

表 3 提案手法 (最も F 値が高い組合せ) と既存手法の比較結果

Table 3 Comparison result between proposed method (combination with the highest F-value) and existing method.

セキュリティ要件	教師あり アルゴリズム	提案手法				既存手法			
		SEThesaurus	分散表現 (クラスタ距離)	フレーズ抽出	適合率 再現率 F 値	適合率 再現率 F 値			
機密性	Naïve Bayes	あり	SO_Word2Vec (0.14)	2-gram	0.751 <b>0.807</b> 0.778	0.709 0.783 0.744			
	SVM	あり	SO_Word2Vec (0.02)	N-gram	0.809 0.770 <b>0.789</b>	0.778 0.748 0.763			
	SMO	なし	SO_Word2Vec (0.02)	N-gram	<b>0.810</b> 0.766 0.787	0.789 0.739 0.763			
完全性	Naïve Bayes	あり	fastText (0.02)	2-gram	0.742 <b>0.829</b> 0.783	0.709 0.806 0.754			
	SVM	なし	SO_Word2Vec (0.12)	N-gram	<b>0.806</b> 0.781 <b>0.793</b>	0.782 0.767 0.774			
	SMO	なし	SO_Word2Vec (0.14)	N-gram	0.805 0.776 0.790	0.790 0.762 0.775			
責任追跡性	Naïve Bayes	あり	SO_Word2Vec (0.14)	2-gram	0.835 0.834 0.834	0.769 <b>0.855</b> 0.810			
	SVM	なし	SO_Word2Vec (0.18)	N-gram	<b>0.856</b> 0.832 <b>0.844</b>	0.829 0.817 0.823			
	SMO	なし	SO_Word2Vec (0.18)	N-gram	0.854 0.826 0.840	0.840 0.810 0.825			

を比較する。

手順 3.3: フレーズ抽出による特徴ベクトルの作成

次に, 単語の順序関係を考慮するために単語 N-gram により抽出したフレーズを特徴量とした特徴ベクトルを作成する. 評価実験では, 1-gram, 2-gram, 3-gram をそれぞれ特徴量にした場合と 1-gram と N-gram を特徴量にする場合の 4 パターンを比較の対象とする. 最後に, 構築した特徴ベクトルを Naïve Bayes, SVM, SMO に適用して分類器を構築する.

手順 4: 分類精度の評価 テストデータに対して分類を行い, 分類器の分類精度の評価を行う.

手順 5: 10 分割交差検証 学習データとテストデータの組合せを変えて手順 1~4 までを合計 10 回実施する.

手順 6: 10 分割交差検証の繰返し 手順 2~5 をデータセットの分割方法を変えて合計 10 回実施する. 手順 5~6 で得られた合計 100 回の平均値を分類精度の評価に用いる.

手順 7: 残りのセキュリティ要件を対象にした実験 手順 1~6 を残り 2 種類のセキュリティ要件に対して同様に行う.

手順 8: 分類精度の比較 既存手法についても 10 分割交差検証を 10 回行い得られた合計 100 回の結果の平均値を算出し, 手順 1~7 で得られた提案手法の分類精度と比較する. ただし, 提案手法については 144 通りの組合せ (類義語統一処理 4 種類 × フレーズ抽出 4 種類 × クラスタ距離の閾値 9 通り) のうち, 適合率と再現率の調和平均である F 値が最も高くなる組合せを用いる.

5. 実験結果

表 3 に提案手法と既存手法の実験結果を示す. 表 3 において, カッコ内の数値はクラスタ距離の閾値を表している. 太字の数値はセキュリティ要件ごとに各評価指標の値が最も高かった組合せを示している. たとえば, 機密性では, 適合

表 4 対応のある t 検定の結果 (p 値)

Table 4 Result of paired t-test (p-value).

セキュリティ要件	教師あり アルゴリズム	適合率	再現率	F 値
機密性	Naïve Bayes	0.00	0.00	0.00
	SVM	0.00	0.00	0.00
	SMO	0.00	0.00	0.00
完全性	Naïve Bayes	0.00	0.00	0.00
	SVM	0.00	0.00	0.00
	SMO	0.00	0.00	0.00
責任追跡性	Naïve Bayes	0.00	0.00	0.00
	SVM	0.00	0.00	0.00
	SMO	0.00	0.00	0.00

率は「SMO+SEThesaurus なし+SO\_Word2Vec (0.02)+2-gram」の組合せが, 再現率は「Naïve Bayes+SEThesaurus あり+SO\_Word2Vec (0.14)+N-gram」の組合せが, F 値は「SVM+SEThesaurus あり+SO\_Word2Vec (0.02)+N-gram」の組合せが最も高い値を示している.

表 4 は, 10 分割交差検証の 10 回分 (合計 100 回) の結果を用いて既存手法と提案手法の各評価指標 (適合率, 再現率, F 値) に対して行った対応のある t 検定の結果 (p 値) を示している. 表 4 より, すべての評価指標で提案手法と既存手法の分類精度に統計的有意差 ( $p < 0.01$ ) が確認された.

表 3 より, 各教師ありアルゴリズムによって類義語統一処理とフレーズ抽出処理の組合せは異なるものの, Naïve Bayes を用いて責任追跡性を分類する場合の再現率を除いて, 提案手法の方が既存手法よりもすべてにおいて適合率, 再現率, F 値が高くなるのが分かる. 特に, 3 種類のセキュリティ要件で共通して F 値が高くなるのは, 提案手法を用いて教師ありアルゴリズムに SVM を適用した分類器であることが分かる. 以上より, 本研究で提案する類義語統一処理とフレーズ抽出処理がセキュリティ要件分類の精度向上に寄与するといえる.

## 6. 考察

### 6.1 類義語統一処理の効果

#### 6.1.1 SEThesaurus の効果

実験の結果, Naïve Bayes を用いて分類する場合に SEThesaurus を適用した方が F 値が高かった. 一方で, SVM, または, SMO を用いて完全性と責任追跡性を分類する場合には SEThesaurus を適用しない方が F 値が高かった. このことから, SEThesaurus を用いた類義語の統一が必ずしも分類精度の向上に寄与するわけではないことが分かった.

Naïve Bayes を用いて分類する場合に SEThesaurus を適用した方が分類精度が高い理由として, Naïve Bayes の特性があげられる. Naïve Bayes は, 特徴量は独立していると仮定し, 特徴量があるクラスに属する事前確率の積和を用いて分類を行う. Naïve Bayes の提案手法では 2-gram のフレーズを抽出しているが, 2-gram は 1-gram と比べて含まれる要件文は少ない. SEThesaurus は略語やタイプミスなどの表記ゆれ, 派生語や活用語といった関係にある単語を類義語として統一するため, SEThesaurus 適用前と比べて学習で用いた 2-gram が該当する要件文が増え, 正しく分類できたのではないかと考えられる. たとえば, Naïve Bayes の提案手法で, SEThesaurus を適用することで正しく分類できた “therefore query by physician name allows via identify exam result completed by given provider” という責任追跡性の要件文がある. この要件文に含まれる “query by” というフレーズを含む要件文はデータセット中 2 件 (責任追跡性 1 件) だけだった. そのため, 学習した文のラベルに偏った重みづけとなり, SEThesaurus なしでは正しく分類できていなかった. SEThesaurus では “query” と “querying” を統一するため, “query by” を含む要件文は 4 件 (責任追跡性 3 件) となり, SEThesaurus を適用した場合には正しく分類できたと考えられる.

一方, SVM, または, SMO を用いて完全性と責任追跡性を分類する場合には SEThesaurus を適用しない方が F 値が高かった. SEThesaurus を適用しない方が F 値が高かった理由として, SEThesaurus を適用して類義語を統一することで抽出される N-gram ( $N \geq 2$ ) が増加することが関係していると考えられる.

SVM を用いて完全性を分類したときに [SVM+SEThesaurus なし+SO\_Word2Vec (0.12)] と [SVM+SEThesaurus あり+SO\_Word2Vec (0.12)] でそれぞれ抽出された N-gram の件数を調査した. 1-gram の件数は SEThesaurus なしと比べて SEThesaurus ありの方が平均して 269 件少なく, N-gram ( $N \geq 2$ ) の件数は SEThesaurus なしと比べて SEThesaurus ありの方が平均して 176 件多かった. N-gram ( $N \geq 2$ ) は 1-gram と比べて含まれる要件文が少なくなるため, SEThesaurus を適用して類義語を統一することで増加した N-gram ( $N \geq 2$ ) の重みが高くなり

誤った分類になったと考えられる.

#### 6.1.2 分散表現の効果

評価実験の結果, Naïve Bayes を用いて完全性を分類する場合を除いて, 3 つの分散手法のうち SO\_Word2Vec を用いた方が F 値が高かった. 精度の高い分散表現を取得するためには, 分散表現手法の学習に Stack Overflow のような大規模なテキストデータが必要である. 本実験で Word2Vec と fastText の学習に用いたデータセットは Stack Overflow のような大規模データではないため, 多くの場合には SO\_Word2Vec を用いた方が分類精度が高かったと考えられる.

Naïve Bayes を用いて完全性を分類する場合では 3 つの分散表現手法のうち fastText を用いた方が F 値が高かった. 分散表現手法は同じ学習データであっても学習するたびに取得される分散表現が異なるため, 小規模なデータセットを学習する場合には同じように精度の向上に寄与するとは限らない. そのため, 本研究で扱ったような規模のデータセットでは学習済みモデルである SO\_Word2Vec を適用する方が良いことが分かった.

SO\_Word2Vec は本研究で対象としたような個別のシステムに固有の単語は学習していないため, システム固有の単語に対する適切な分散表現を取得することができない. そのため, あらかじめ大規模なテキストデータを学習した重みを用いて少量のテキストデータを学習する転移学習を用いることで, 将来的にはドメイン固有の単語の分散表現も取得できると考えられる.

### 6.2 フレーズ抽出の効果

評価実験の結果, 教師ありアルゴリズムに SVM, SMO を用いた場合には MED を用いたフレーズ抽出を行った場合が最も F 値が高かった. このことから, MED を用いたフレーズ抽出は分類精度の向上に効果があると考えられる. MED を用いて抽出したフレーズと分類精度の関係を調査するため, F 値が最も高かった SVM を用いた場合の提案手法と既存手法の分類結果が異なった要件文を対象に分析を行った. MED を用いて抽出したフレーズと提案手法の分類結果の例を表 5 に示す. 太字で示しているフレーズは表中の要件文から共通して抽出されたフレーズである.

表 5 の 1 つ目の要件文は既存手法では分類できなかったが提案手法が責任追跡性と正しく分類できた要件文である. 提案手法が 1 つ目の要件文から抽出したフレーズが出現する要件文の多くは責任追跡性の要件文であった. 1 つ目の要件文から抽出したフレーズにはユーザ認証を行うことを示すフレーズである “log in” がある. ユーザ認証を行うことでシステムの責任追跡性を保証できるため, “log in” は責任追跡性の要件文に多く出現したと考えられる.

表 5 の 2 つ目の要件文は既存手法では分類できなかったが提案手法が完全性と正しく分類できた要件文である. 提

表 5 MED を用いて抽出したフレーズと提案手法の分類結果  
Table 5 Classification results and phrases extracted using MED.

前処理済みの要件文	提案手法で抽出したフレーズとフレーズの出現割合	分類結果
log in as oscar	log in : 14 件中 13 件が責任追跡性の要件文 in as : 16 件中 15 件が責任追跡性の要件文 as oscar : 44 件中 43 件が責任追跡性の要件文 in as oscar : 11 件中 11 件が責任追跡性の要件文	責任追跡性と正しく分類
system shall provide ability for concurrent user view clinical documentation template	ability for : 27 件中 23 件が完全性の要件文 clinical documentation : 18 件中 11 件が完全性の要件文 documentation template : 3 件中 3 件が完全性の要件文 for concurrent : 3 件中 2 件が完全性の要件文 <b>provide ability : 606 件中 514 件が完全性の要件文</b> <b>shall provide : 469 件中 384 件が完全性の要件文</b> <b>system shall : 698 件中 535 件が完全性の要件文</b> provide ability for : 21 件中 19 件が完全性の要件文 <b>shall provide ability : 377 件中 311 件が完全性の要件文</b> <b>system shall provide : 469 件中 384 件が完全性の要件文</b> <b>system shall provide ability : 377 件中 311 件が完全性の要件文</b>	完全性と正しく分類
system shall provide ability filter sort result by type of test test date	<b>provide ability : 606 件中 514 件が完全性の要件文</b> <b>shall provide : 469 件中 384 件が完全性の要件文</b> <b>system shall : 698 件中 535 件が完全性の要件文</b> <b>shall provide ability : 377 件中 311 件が完全性の要件文</b> <b>system shall provide : 469 件中 384 件が完全性の要件文</b> <b>system shall provide ability : 377 件中 311 件が完全性の要件文</b> ability filter : 9 件中 4 件が完全性の要件文 by type : 3 件中 1 件が完全性の要件文 of test : 3 件中 1 件が完全性の要件文 result by : 14 件中 5 件が完全性の要件文 type of : 31 件中 13 件が完全性の要件文 provide ability filter : 8 件中 4 件が完全性の要件文	完全性と誤って分類

案手法が2つ目の要件文から抽出したフレーズには“system shall provide ability”というフレーズがあるが、これはセキュリティ要件に出現する頻度が高いフレーズである。そのため、“system shall provide ability”やこのフレーズの部分集合となるフレーズが出現する要件文の多くは完全性の要件文であった。

これらのことから、MEDを用いたフレーズ抽出によってセキュリティ要件の特徴を示すフレーズを抽出できたために、既存手法では分類できなかった要件文を提案手法は正しく分類することができたと考えられる。

しかしながら、既存手法が正しく分類できた要件文を提案手法が誤って分類した場合もある。表5の3つ目の要件文は既存手法では正しく分類できたが提案手法で誤って完全性と分類した要件文である。3つ目の要件文は機密性と責任追跡性に該当するが完全性には該当しない要件文である。3つ目の要件文から抽出したフレーズには、2つ目の要件文と共通するフレーズが6つあった。これら6つのフレーズは完全性にのみ出現するフレーズではなかったため、提案手法は3つ目の要件文を誤って完全性と分類したと考えられる。本手法ではN-gramのMED値に対して外れ値除去を行い抽出するフレーズを選択したが、セキュリティ

要件にのみ出現するようなフレーズを抽出するようにフィルタリングすることでさらなる精度向上を期待できる。

### 6.3 制約

本研究では6つの医療系システムの要件文を対象に評価実験を行った。そのため、医療系以外のドメインのシステムの要件文を対象にした場合、本稿で示している類義語統一処理とフレーズ抽出の組合せとは異なる組合せとなる可能性がある。

本研究では、144通りの組合せ（類義語統一処理4種類×フレーズ抽出4種類×クラスタ距離の閾値9通り）を評価した。さらに、手順3.2で述べたようにハイパーパラメータの探索を多数行っている。そのため、表3で示した比較結果は、評価データに過剰に適合した組合せを選択している可能性がある。今後の課題として、今回の評価実験で得た組合せとパラメータが別のデータセットでも同様の分類性能を示すかどうかを検証する必要がある。

また、今回の評価実験では、アルゴリズムごとに異なる最適な組合せが存在することが確認された。結果的に、単語の類似性と順序関係のどちらがより重要であったのかを明確に示すことができなかった。今後はablation studyを



通じて SEThesaurus, 分散表現, フレーズ抽出がそれぞれ具体的にどの程度効果があるのかを明らかにする必要がある。

## 7. むすび

本研究ではセキュリティ要件分類手法の精度向上のために, 類義語統一処理とフレーズ抽出を用いたセキュリティ要件分類手法を構築した。本手法では類義語を統一するために (1-A) シソーラスを用いた類義語の統一, (1-B) 単語分散表現を用いた類義語の統一の 2 つのアプローチをとった。また, 単語の順序関係を考慮するために, (2-A) 固定長の単語 N-gram をフレーズとして抽出する場合と (2-B) 可変長の単語 N-gram をフレーズとして抽出する場合の 2 つのアプローチをとった。

評価実験の結果, 分類に用いる教師ありアルゴリズムと分類対象のセキュリティ要件によって, F 値が最も良い類義語統一処理とフレーズ抽出の組合せが異なることが分かった。類義語統一処理に関しては Naïve Bayes を用いて完全性を分類する場合を除いて SO\_Word2vec を用いる場合が最も F 値が高かった。フレーズ抽出に関しては, Naïve Bayes では 2-gram を用いた場合, SVM, SMO では MED を用いた場合に最も F 値が高かった。

今後の課題としては, 転移学習を用いた分散表現の改善や特徴量に用いる N-gram の選択方法の確立があげられる。

謝辞 本研究の一部は, 文部科学省科学研究補助金 (基盤 (A) : 17H00731, 基盤 (C) : 18K11243) による助成を受けた。

## 参考文献

- [1] Eckhardt, J., Vogelsang, A. and Fernández, D.M.: Are “Non-functional” Requirements Really Non-functional?: An Investigation of Non-functional Requirements in Practice, *Proc. 38th International Conference on Software Engineering (ICSE '16)*, pp.832–842 (2016).
- [2] Borg, A., Yong, A., Carlshamre, P. and Sandahl, K.: The Bad Conscience of Requirements Engineering: An Investigation in Real-World Treatment of Non-Functional Requirements, *Proc. 3rd Conference on Software Engineering Research and Practice in Sweden (SERPS '03)*, pp.1–8 (2003).
- [3] Cleland-Huang, J., Settimi, R., Zou, X. and Solc, P.: Automated Classification of Non-functional Requirements, *Requir. Eng.*, Vol.12, No.2, pp.103–120 (2007).
- [4] Zhang, W., Yang, Y., Wang, Q. and Shu, F.: An Empirical Study on Classification of Non-Functional Requirements, *Proc. 23rd International Conference on Software Engineering & Knowledge Engineering (SEKE '11)*, pp.444–449 (2011).
- [5] Slankas, J. and Williams, L.: Automated extraction of non-functional requirements in available documentation, *Proc. 1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE '13)*, pp.9–16 (2013).
- [6] Casamayor, A., Godoy, D. and Campo, M.: Identification of Non-functional Requirements in Textual Specifications: A Semi-supervised Learning Approach, *Inf. Softw. Technol.*, Vol.52, No.4, pp.436–445 (2010).
- [7] Riaz, M., King, J., Slankas, J. and Williams, L.: Hidden in plain sight: Automatically identifying security requirements from natural language artifacts, *Proc. 22nd International Requirements Engineering Conference (RE '14)*, pp.183–192 (2014).
- [8] Mahmoud, A.: An information theoretic approach for extracting and tracing non-functional requirements, *Proc. 23rd International Requirements Engineering Conference (RE '15)*, pp.36–45 (2015).
- [9] Walton, J.P.: Developing an Enterprise Information Security Policy, *Proc. 30th Annual ACM SIGUCCS Conference on User Services (SIGUCCS '02)*, pp.153–156 (2002).
- [10] Baker, C., Deng, L., Chakraborty, S. and Dehlinger, J.: Automatic Multi-class Non-Functional Software Requirements Classification Using Neural Networks, *Proc. 43rd Annual Computer Software and Applications Conference (COMPSAC '19)*, pp.610–615 (2019).
- [11] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (2019).
- [12] Hey, T., Keim, J., Koziolok, A. and Tichy, W.F.: NoR-BERT: Transfer Learning for Requirements Classification, *Proc. 28th International Requirements Engineering Conference (RE '20)*, pp.169–179 (2020).
- [13] Chen, C., Xing, Z. and Wang, X.: Unsupervised Software-specific Morphological Forms Inference from Informal Discussions, *Proc. 39th International Conference on Software Engineering (ICSE '17)*, pp.450–461 (2017).
- [14] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. and Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality, *Proc. 26th International Conference on Neural Information Processing Systems (NIPS '13)*, pp.3111–3119 (2013).
- [15] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T.: Enriching Word Vectors with Subword Information, *Trans. Association for Computational Linguistics*, Vol.5, pp.135–146 (2017).
- [16] Efstathiou, V., Chatzilenas, C. and Spinellis, D.: Word Embeddings for the Software Engineering Domain, *Proc. 15th International Conference on Mining Software Repositories (MSR '18)*, pp.38–41 (2018).
- [17] Bu, F., Zhu, X. and Li, M.: Measuring the Non-compositionality of Multiword Expressions, *Proc. 23rd International Conference on Computational Linguistics (COLING '10)*, pp.116–124 (2010).
- [18] Tsoumakas, G. and Katakis, I.: Multi-label classification: An overview, *Int. J. Data Warehousing and Mining*, Vol.2007, pp.1–13 (2007).



宮崎 智己

平成 29 年和歌山大学システム工学部  
情報通信システム学科卒業。平成 31  
年同大学大学院システム工学研究科博  
士前期課程修了。ソフトウェア工学、  
特にマイニングソフトウェアリポジト  
リの研究に従事。



東 裕之輔

平成 27 年和歌山大学システム工学部  
情報通信システム学科卒業。平成 29  
年同大学大学院システム工学研究科博  
士前期課程修了。同年株式会社日本総  
合研究所。令和 2 年より和歌山大学大  
学院システム工学研究科博士後期課程  
(在学中)。ソフトウェア工学、特にマイニングソフトウェ  
アリポジトリの研究に従事。



大平 雅雄 (正会員)

平成 10 年京都工芸繊維大学工芸学部  
電子情報工学科卒業。平成 15 年奈良  
先端科学技術大学院大学情報科学研究  
科博士後期課程修了。同大学情報科学  
研究科助教を経て、平成 24 年和歌山  
大学システム工学部准教授。博士 (工  
学)。ソフトウェア工学、特にマイニングソフトウェアリ  
ポジトリの研究に従事。電子情報通信学会、IEEE、ACM  
各会員。