日本語非機能要件の自動分類における教師あり学習アルゴリズムの評価

大東 誠弥
和歌山大学 システム工学部
ohigashi.seiya@g.wakayama-u.jp

福井 克法
和歌山大学大学院 システム工学研究科
fukui.katsunori@g.wakayama-u.jp

宮崎 智己
和歌山大学大学院 システム工学研究科
miyazaki.tomoki@g.wakayama-u.jp

大平 雅雄
和歌山大学 システム工学部
masao@sys.wakayama-u.ac.jp

要旨

要件定義の際に作成される要件定義書に機能要件と非機能要件がある．中でも非機能要件は，明確な定義が存在せず文献によって定義が異なることがある *[1]*，といった理由により見落とされやすいことが知られている．また，要件定義書はシステムの規模が大きくなるにつれ要件も多くなる．大量の要件を目視で確認し，非機能要件の見落としを防ぐことは困難である．

非機能要件の見落としを未然に防ぐために非機能要件分類手法が提案されている．先行研究 *[2]* では，*k*-近傍アルゴリズム，*SMO* アルゴリズム，*Naïve Bayes* アルゴリズムを用いた場合の分類精度を比較し，*SMO* アルゴリズムが最も分類精度が高いことを示している．しかしながら，先行研究が分類の対象としている非機能要件はすべて英語で記述された非機能要件であり，英語と文構造が異なる日本語非機能要件の分類においても先行研究の手法が最も適しているとは限らない．

本研究では，日本語非機能要件の分類に適した教師あり学習アルゴリズムの評価を行う．また，品詞を限定した場合についての分類精度を比較するとともに，類似する単語を統一する単語の正規化処理を行った場合の分類精度についても調査する．評価実験を行った結果，日本語非機能要件の分類においても先行研究と同様に *SMO* アルゴリズムを用いた場合が最も精度が高いことが確認された．また，*SMO* アルゴリズムを用いる場合，分類に用いる単語を名詞のみ，類似単語を統一することで，精度が向上することが確認された．

1. はじめに

ソフトウェア開発は，大きく分けて要件定義，設計，実装，テストの工程に分かれている．その中でも要件定義は，発注者が，開発者にどのようなシステムの開発を依頼するか定義する工程である．要件定義での定義が曖昧であったり間違った定義を行ってしまうと，開発の途中で手戻りが発生したりシステム障害の原因となることがある．また，要件定義では，システムのセキュリティやシステムの仕様についての定義も行う．これらのことから，要件定義は重要視されている工程の 1 つとなっている．

要件定義の際に作成されるのが要件定義書であり，要件定義書内に記述される要件には，機能要件と非機能要件がある．中でも非機能要件は，明確な定義が存在せず文献によって定義が異なることがある [1]．また，非機能要件は，機能要件と混在して記述される [3]，要件定義書の記述者によって記述表現が異なる [4]，定義が曖昧になる [5] などの問題があり，機能要件と比較すると見落とされやすいことが知られている．要件定義書はシステムの規模が大きくなるにつれ要件も多くなる．大量の要件を目視で確認し，非機能要件の見落としを防ぐことは困難である．

非機能要件の見落としを未然に防ぐために，非機能要件分類手法が提案されている [6,7]．先行研究では，機械学習を用いて非機能要件の分類を行っている．Slankas らの研究 [2] では，線形アルゴリズムである SMO アルゴリズムが最も高い分類精度になることを示した．また，分類精度向上のために分類に用いる品詞に注目し，冠詞

を除いて分類を行うことで分類精度が向上することを示した．しかしながら，先行研究で分類の対象としている非機能要件は，すべて英語で記述された非機能要件である．英語と日本語では文の構造が異なる．また，文を解釈する際に重要になる品詞も異なる．そのため，日本語非機能要件を分類対象とした場合，先行研究の手法が最も適しているとは限らない．

本研究では，日本語非機能要件の分類に適した教師あり学習アルゴリズムの評価を行う．また，品詞を限定した場合についての分類精度を比較するとともに，類似する単語を統一する単語の正規化処理を行った場合の分類精度についても調査する．

2. 非機能要件分類

2.1. 非機能要件

要件定義はシステム開発の初めに行う工程であり，発注者が開発者にどのようなシステムの開発を依頼するか定義する工程である．要件定義では，システムの概要，システム要件，システムを作成する目的，開発スケジュールなどが決められる．設計や実装，テストを行う開発者は，要件定義で決めた内容をもとに各工程を行う．要件定義が適切に行われていないと，他の工程に影響を及ぼすため，要件定義は重要な工程の 1 つとなっている．

要件定義で決めた内容を記した書類は要件定義書と呼ばれる．要件定義書に記述される要件のうち，システム要件には大きく分けて機能要件と非機能要件が存在する．

- 機能要件：システムの機能について定義した要件
- 非機能要件：システムの機能以外に関わるセキュリティやシステムの性能について定義した要件

機能要件と非機能要件は，それぞれ章ごとに分けて記述されていることがほとんどである．しかし，実際には機能要件を記述している章の中に非機能要件が混在している場合がある [3]．非機能要件は明確な定義が存在せず，文献によって定義が全く異なる [1]．また，要件定義書の記述者によって記述表現が異なる [4]，定義が曖昧になる [5] ことがあり，他の開発者が誤った解釈をしてしまうこともある．そのため，非機能要件を目視によって，要件定義書内から特定することは困難である．

2.2. 先行研究

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 [2,8–11]．教師あり学習，半教師あリ学習，教師なし学習それぞれを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 [8]．分類に用いる特徴量に，単語のみ（BoW ベクトル），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，単語を特徴量に用いた場合が精度が最も高かった．

Slankas らは k-近傍アルゴリズムと SMO アルゴリズム，Naïve Bayes アルゴリズムを用いた非機能要件分類手法を提案している [2]．ナイーブベイズと 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．実験の結果，SMO アルゴリズムが最も精度が高く，次いで k-近傍アルゴリズム，Naïve Bayes アルゴリズムの順に精度が高いことを示した．

また，Riaz らはセキュリティ要件の詳細な分類を対象とした分類手法を提案している [10]．先行研究 [2] と同様の方法で分類器を構築しており，評価実験の結果では SMO と k 近傍法の精度が高かった．

これら先行研究で用いられたデータセットは，すべて英語で記述された非機能要件である．[2,10] で最も分類精度が高かった SMO アルゴリズムが，日本語非機能要件の分類に対しても最も精度が良くなるとは限らないため，本研究では [2] で用いられた 3 つの教師あり学習アルゴリズムを日本語非機能要件の分類に適用して評価を行う．

3. 研究の目的

本研究の目的は，日本語非機能要件の分類に適した教師あり学習アルゴリズムを評価することである．本研究では以下のリサーチクエスチョンを設定する．

RQ1：日本語非機能要件の分類において，最も分類精度が良い教師あり学習アルゴリズムは何か？

先行研究 [2] では，英語で記述された非機能要件について，k-近傍アルゴリズム，SMO アルゴリズム，Naïve

表 1. ISO25010 による非機能要件の品質特性

非機能要件の種類	説明
機能適合性	明示された状況下で使用するとき，明示的ニーズ及び暗黙のニーズを満足させる機能を，製品又はシステムが提供する度合い．
性能効率性	明記された状態 (条件) で使用する資源の量に関係する性能の度合い．
互換性	同じハードウェア環境又はソフトウェア環境を共有する間，製品，システム又は構成要素が他の製品，システム又は構成要素の情報を交換することができる度合い，およびその要求された機能を実行できる度合い．
使用性	明示された利用状況において，有効性，効率性及び満足性を持って明示された目標を達成するために，明示された利用者が製品又はシステムを利用することができる度合い．
信頼性	明示された時間帯で，明示された条件下に，システム，製品又は構成要素が明示された機能を実行する度合い．
セキュリティ	人間又は他の製品もしくは，システムが認められた権限の種類及び水準に応じたデータアクセスの度合いをもてるように，製品又はシステムが情報及びデータを保護する度合い．
保守性	意図した保守者によって，製品又はシステムが修正することができる有効性及び効率性の度合い．
移植性	1 つのハードウェア，ソフトウェア又は他の運用環境若しくは利用環境からその他の環境に，システム製品又は構成要素を移すことができる有効性及び効率性の度合い．

Bayes アルゴリズムの 3 種類の教師あり学習アルゴリズムを用いて分類精度の比較を行っている．しかしながら，日本語非機能要件の分類を行う場合には，英語と日本語とで文の構造が異なっている点，日本語には主語を省略することがある点の二点の違いが挙げられる．以上の二点から，必ずしも先行研究で最も精度が高かったアルゴリズムが日本語非機能要件の分類にも適しているとは限らない．RQ1 では，日本語非機能要件に適した教師あり学習アルゴリズムを比較実験により明らかにする．

RQ2 : 分類に使用する品詞の限定や，単語の正規化を行うことで分類精度に違いは生じるか？

先行研究 [2] では，分類に使用する品詞から冠詞を取り除くことで分類精度が向上することを明らかにしている．しかしながら，日本語非機能要件の分類において，同じように品詞を限定した場合に精度が良くなるとは限らない．また，品詞の問題だけでなく，英語とは違い日本語の場合，書き手によって要件文の語尾が異なることや要件の表現が異なることが考えられる．RQ2 では，分類に使用する品詞と，類似する単語の統一（単語の正規化）に着目して分類精度を比較する．なお，分類に使用する品詞は，独立して意味を持つ名詞と動詞に着目して分類を行う．

4. ケーススタディ

4.1. データセット

本研究では，データセットとして厚生労働省が公開している要件定義書，調達仕様書を使用する．データセットとして使用した，要件定義書及び調達仕様書の内訳及び，各システムの要件数を表 2 に示す．

表 2. 各システムの要件数

	ハローワーク (4 件)	労働保険 (6 件)	年金 (2 件)	合計 (12 件)
機能適合性	12	12	9	33
性能効率性	45	98	72	215
互換性	30	21	22	73
使用性	48	14	1	63
信頼性	21	54	33	108
セキュリティ	97	93	94	284
保守性	89	188	128	405
移植性	25	28	16	69
その他	561	654	521	1,736
合計	901	1,134	881	2,916

厚生労働省の要件定義書と調達仕様書を選んだ理由は以下の通りである．

- インターネット上に公開されている要件定義書及び調達仕様書の数が豊富であること
- 公開されている文書が pdf 形式で用意されており，データを入手しやすいこと

要件文には非機能要件の分類ラベルが付いていないため，要件定義書から 1 文ずつ抜き出し，1 文ごとに ISO25010 の品質特性に基づき手作業でラベル付けを行う．ISO25010 が定めた品質特性の項目は表 1 の通りである．なお，8 つの品質特性のどれにも属さないと判断されたものに対しては,「その他」のラベルを付けて対応した．ただし，表形式で要件が記述される場合があり，この場合は図 1 のように表の 1 列分を 1 文と解釈する．

No	利用者	機能	スループット	備考
1	すべての拠点の利用者	適用徴収機能	382,000件／日 35.370 件 ／ 秒 (※1)	※1 ピーク時アクセス件数（件／日）÷（3時間×3600秒）で算出。
2	インターネット経由でアクセスする利用者	適用事業場公開機能	1,433,000 ページビュー／月	

図 1. 表形式の記述例

職員等の利用するシステムは、使いやすい操作性を持たせること。	使用性
本調達に係る業務で発生し得るセキュリティリスクについて、受注者において対策を実施すること。	セキュリティ
一連の処理又はリクエストを発行してからレスポンス（画面表示）が返るまで、5秒以内とすること	性能効率性

図 2. ラベル付けの一例

また，要件定義書や調達仕様書には，機能要件や非機能要件だけでなく，調達の背景や目的，開発スケジュールなどが記されている．本研究では，これらの記述内容については対象外とする．また，図として記述された要件についても対象外とする．

ラベル付けを行うにあたり，個人でラベル付けを行った場合に主観が混入する恐れがあるため，ソフトウェア工学の知識を有する学生 2 名でラベル付けを行った．この際，他者が付けたラベルを見てラベル付けを行ってしまうとお互いの意見に影響される可能性がある．そのため，それぞれ独立してラベル付けを行い，その後 2 名でラベルの照らし合わせを行った．両者が異なったラベルを付けていた場合は，話し合いにてどちらかのラベルに決定するか，もしくは，1 文に対して 2 つのラベルを付けた．ラベル付けの一例を図 2 に示す．

4.2. 分類手順

本節では，設定した RQ1,2 について分析を行うための手順について説明する．手順は以下の 4 つのステップから構成される．

1. 要件文を単語ごとに分割するために形態素解析を行う
2. 教師あり学習アルゴリズムで扱うために，自然言語で記述された要件文を Bag of Words を用いてベクトル化する

```
脆弱性の対策を行った
脆弱性	名詞,固有名詞,一般,*,*,*,脆弱性,ゼイジャクセイ,ゼイジャクセイ
の	助詞,連体化,*,*,*,*,の,ノ,ノ
対策	名詞,サ変接続,*,*,*,*,対策,タイサク,タイサク
を	助詞,格助詞,一般,*,*,*,を,ヲ,ヲ
行っ	動詞,自立,*,*,五段・ワ行促音便,連用タ接続,行う,オコナッ,オコナッ
た	助動詞,*,*,*,特殊・タ,基本形,た,タ,タ
EOS
```

図 3. MeCab で形態素解析を行った例

3. RQ1 の実験として各要件文の BoW ベクトルを入力として，各教師あり学習アルゴリズムを適用し要件文の分類を行う
4. RQ2 の実験として分散表現を利用して類似する単語の統一（単語の正規化）を行う

4.2.1. 形態素解析

形態素解析とは，1 文を単語ごとに分割する手法である．本研究では，オープンソース形態素解析エンジンの MeCab [12] を使用して形態素解析を行う．なお，MeCab 標準の辞書では,「厚生労働省」のような固有名詞を単語に分割しようとした場合,「厚生」と「労働省」といった分割をしてしまう．この問題を解決するため，より単語の登録数の多い辞書である mecab-ipadic-NEologd [13] を使用した．MeCab を用いて形態素解析を行った例を図 3 に示す．MeCab の出力から後述する単語の分散表現を作成し，類似する単語の統一（単語の正規化）を行う．

4.2.2. Bag of Words

自然言語で記述された要件文のままでは教師あり学習アルゴリズムを用いて分類することができない．本研究では，一般的なベクトル化手法である Bag of Words を用いて各要件文のベクトル化を行う．Bag of Words とは，出現する単語の各回数を特徴量としてベクトル化する手法である．本研究で扱う Bag of Words は，各要件文を行，各要件文に出現する単語の各回数を列とした行列である．

4.2.3. 教師あり学習アルゴリズム

Bag of Words を用いて生成した行列を入力として，教師あり学習アルゴリズムを用いて分類を行う．分類に使

用する教師あり学習アルゴリズムは [2] において採用されていた以下の 3 種類である．

- k-近傍アルゴリズム：学習データとテストデータそれぞれの距離を計算しておき，未知データ（テストデータ）から最も近い距離となる k 個の学習データを取得し多数決でクラスを決定する．なお，本研究では，[2] と同様に $k = 1$ で距離計算を行っている．
- SMO アルゴリズム [14]：学習データとテストデータそれぞれに対して，マージンと呼ばれる 2 クラスの境界面から各クラスに属するデータの最短距離が最大となるよう境界線を引いて分類を行う．
- Naïve Bayes アルゴリズム [15]：ベイズの定理に基づいてテストデータが属するクラスの確率を算出し，確率の最も高くなったクラスがテストデータの属するクラスと推定する．

4.2.4. 類似単語の統一処理（単語の正規化）

RQ2 についての実験では，1 文ごとに分割した単語を用いてベクトル化を行う前に，分類に用いるすべての単語のうち類似度の高い単語同士の置換を行う．単語の正規化を行う理由は以下の 2 点である．

- 要件定義書ごとに要件文に表記揺れがあり，同じ内容の要件文でも語尾が変化すること，表現の仕方が変わることで，別の要件文に分類される可能性があること
- 日付や時間などに関する要件文を Bag of Words を用いてベクトル化した場合，同じ日付および時間に関する要件であっても，数字が異なるだけで全く別の非機能要件と分類される可能性があること

単語の正規化を行うために本研究では単語の分散表現手法を用いる．単語の分散表現手法は，単語を任意の次元ベクトルで表現する手法である．単語をベクトルで表現することにより，ベクトル間の類似度を計算し単語間の類似度を算出することができる．

本研究では，分散表現手法のうち Word2Vec [16] を用いることで単語のベクトル表現を獲得する．Word2Vec は，2 層のニューラルネットワークを用いた分散表現手法である．Word2Vec では，単語の意味は周辺の単語によって構成されるという仮説に基づいて学習が行われる．

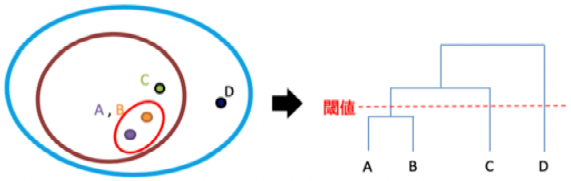

図 4. 樹形図作成の例

つまり，周辺に位置する単語が似ている単語同士は類似した単語であるとみなすアルゴリズムである．

本研究では，鈴木らが作成した日本語 Wikipedia エンティティベクトル [17] を用いて分散表現を取得する．日本語 Wikipedia エンティティベクトルは，Word2Vec [16] を用いて日本語版 Wikipedia を学習させている．

分散表現を取得した後，階層クラスタリングを用いて樹形図を作成する．分散表現の各単語間の距離計算は，コサイン類似度を用いて計算する．また，クラスタ間の距離計算には群平均法を用いる．階層クラスタリングを用いた樹形図作成の例を図 4 に示す．

図 4 のように閾値以下の集合を単語の類似集合とすることで，類似する単語を代表単語で置換することにより統一することができる．閾値は 0 から 1 の値を取り，閾値が大きくなるほどより多くの単語が類似していると判定する．集合に含まれる単語の中から任意の単語を代表単語としてその他の単語を代表単語と置換する．なお，閾値の設定によって抽出されるクラスタが異なるため，置換すべき単語は閾値に依存する．そのため，予備実験を行い最も精度が良かった時の閾値を採用する．

4.3. 分類精度の比較方法

教師あり学習アルゴリズムの分類精度を比較するために，式 1 で表せる適合率 (Precision)，式 2 で表せる再現率 (Recall)，式 3 で表せる適合率と再現率の調和平均である F 値 (F measure) を用いる．教師あり学習アルゴリズムが予測した非機能要件分類を集合 A，実際の要件文に割り当てられている非機能要件分類を集合 B（正解集合）とした場合，適合率は予測した分類の中に正解がどのくらい含まれていたかを表し，再現率は実際の正解集合の中に予測した分類がどのくらい含まれていたかを表す．例えば，人手によって機能適合性のラベルが付けら

表 3. 教師ありアルゴリズムごとの分類精度

	k-近傍			SMO			Naïve Bayes		
	P	R	F 値	P	R	F 値	P	R	F 値
機能適合性	**0.359**	**0.191**	**0.240**	0.247	0.164	0.186	0.102	0.053	0.068
性能効率性	0.711	0.668	0.687	**0.789**	**0.755**	**0.770**	0.594	0.686	0.635
互換性	0.273	0.198	0.224	**0.394**	**0.295**	**0.331**	0.128	0.138	0.131
使用性	0.726	0.402	0.508	**0.777**	**0.528**	**0.620**	0.422	0.251	0.307
信頼性	0.354	0.343	0.344	**0.475**	**0.420**	**0.442**	0.305	0.293	0.295
セキュリティ	0.578	0.648	0.610	**0.759**	**0.724**	**0.740**	0.456	0.686	0.546
保守性	0.474	0.458	0.465	**0.513**	**0.516**	**0.513**	0.263	0.512	0.347
移植性	0.335	0.354	0.339	**0.444**	**0.392**	**0.409**	0.237	0.303	0.263

表 4. 品詞の限定や正規化を行った際の分類精度（平均値）

		k-近傍			SMO			Naïve Bayes		
		P	R	F 値	P	R	F 値	P	R	F 値
正規化なし	品詞すべて	0.476	0.408	0.427	0.550	0.474	0.501	0.313	0.365	0.324
	名詞	0.490	0.432	0.450	**0.572**	**0.494**	**0.523**	0.294	0.395	0.322
	名詞，動詞	0.495	0.412	0.438	0.556	0.486	0.511	0.308	0.382	0.326
正規化あり	品詞すべて	0.471	0.417	0.431	0.555	0.483	0.510	0.317	0.393	0.338
	名詞	0.489	0.432	0.450	**0.571**	**0.501**	**0.526**	0.285	0.417	0.326
	名詞，動詞	0.491	0.412	0.436	0.566	0.493	0.519	0.303	0.400	0.333

れた要件文が 33 件，教師あり学習アルゴリズムによって機能適合性と予測し分類した要件文が 40 件，予測した 40 件のうち機能適合性と正しく分類できた要件文が 20 件であったとする．その時，適合率は $20 \div 40 = 0.500$ と算出でき，再現率は $20 \div 33 = 0.606$ と算出することができる．

$$Precision = \frac{|A \cap B|}{|A|} \tag{1}$$

$$Recall = \frac{|A \cap B|}{|B|} \tag{2}$$

$$F - measure = \frac{2 \times P \times R}{P + R} \tag{3}$$

5. 結果

5.1. RQ1：日本語非機能要件の分類において，最も分類精度が良い教師あり学習アルゴリズムは何か？

RQ1 では，それぞれの教師あり学習アルゴリズムについての分類精度を評価するために，適合率，再現率および適合率と再現率の調和平均である F 値を用いて分類精度の評価を行った．表 3 に 3 種類の教師あり学習アルゴリズム，*k*-近傍アルゴリズム，SMO アルゴリズム，Naïve Bayes アルゴリズムを用いて非機能要件を分類した際の分類精度を示す．

表 3 より，3 種類の教師あり学習アルゴリズムのうち機能適合性以外の非機能要件分類では，SMO アルゴリズムが Precision，Recall，F 値のすべてにおいて最も精度が高くなることが分かる．

5.2. RQ2：分類に使用する品詞の限定や，単語の正規化を行うことで分類精度に違いは生じるか？

RQ2 では，分類に使用する品詞を限定した場合の分類精度，単語を正規化した場合の分類精度の評価を行った．表 4 に分類に用いる品詞を限定した場合，単語の正規化を行った場合についてのそれぞれの分類精度を示す．なお，Precision，Recall，F 値のそれぞれは，8 つの非機能要件分類の結果の平均値を示している．

表 4 より，8 つの非機能要件分類の結果の平均値として SMO アルゴリズムが Precision，Recall，F 値のすべてにおいて最も精度が高くなることが分かる．また，品詞を名詞に限定することが分類精度に最も寄与することが分かる．正規化処理の効果については，SMO アルゴリズムにおいて正規化を行わなかった場合に僅差ながら適合率が最も高く，正規化を行った場合には再現率および F 値が最も高い結果となった．

次章では，RQ1 および RQ2 の結果を踏まえて分析を追加し考察を行う．

6. 考察

6.1. 日本語非機能要件の分類に適したアルゴリズム

RQ1 では，3 つの教師あり学習アルゴリズムを用いて分類を行った．先行研究 [2] と同様に，SMO アルゴリズムの分類精度が最も高い結果となり，Naïve Bayes アルゴリズムが最も低い分類精度となった．ただし,「機能適合性」では SMO アルゴリズムが k-近傍アルゴリズムよりも分類精度が低い結果となった．[2] では，非機能要件の分類ごとに分類精度は算出しておらず（表 4 のように）平均値のみで議論されているため単純な比較はできないが，RQ1 において「機能適合性」のみ SMO アルゴリズムが k-近傍アルゴリズムに分類精度が劣っていた原因について考察するため，要件文に頻出する単語を分析する．

表 5 に,「機能適合性」に分類される要件文に頻出する単語の上位 10 件とその出現割合を示す．また，表 6 に，機能適合性以外の非機能要件分類の 1 例として,「使用性」に分類される要件文に頻出する単語の上位 10 件とその出現割合を示す．

表 5 より,「機能適合性」では，出現割合が最も高い単語が「機能」（7.9%）である.「機能」を含む全要件文 532 件中 42 件が「機能適合性」に該当することを意味する．RQ1 では品詞の限定（および単語の正規化処理）は行っていないため,「機能」に続いて動詞の「する」が出現回数の 2 番目として出現している．さらに,「こと」が続き，一般的に広く使われる単語が上位を占めている．実際,「する」「こと」は，全要件文 2,916 件中，それぞれ 2,831 件，2,082 件に出現しており,「機能適合性」を分類する上で重要な単語（特徴語）として機能していない可能性が高い．4 番目以降の単語についても一般的な単語と言えるものが多く，実際，出現割合も低い．

一方，表 6 より,「使用性」では，半数は出現割合が低い単語が含まれるものの,「ユーザビリティ」（94.7%）や「アクセシビリティ」（81.8%）などの単語は出現割合が高く，要件文の中でも「使用性」に関する要件を記述す

表 5.「機能適合性」における出現上位 10 単語

出現単語	出現回数	出現割合
する	24/2035	1.2%
機能	18/346	5.2%
こと	12/1857	0.6%
提供	11/244	4.5%
及び	10/645	1.6%
者	8/749	1.1%
情報	8/307	2.6%
業務	5/408	1.2%
システム	5/636	0.8%
資料	5/138	3.6%

表 6.「使用性」における出現上位 10 単語

出現単語	出現回数	出現割合
こと	49/1857	2.6%
する	41/2035	2.0%
等	32/830	3.9%
システム	16/636	2.5%
し	15/995	1.5%
利用者	15/72	20.8%
利用	14/133	10.5%
アクセシビリティ	13/16	81.3%
要件	13/178	7.3%
ユーザビリティ	12/13	92.3%

表 7. 全文での動詞の出現回数上位 5 件

出現単語	出現回数	出現割合
する	2831	34.0%
し	1302	15.6%
行う	366	4.4%
さ	295	3.5%
示す	280	3.4%

表 8. 最も分類精度が高かった組み合わせ

	アルゴリズム	分類に用いる品詞	単語の正規化	F 値
機能適合性	k 近傍	名詞と動詞	あり	0.251
性能効率性	SMO	品詞すべて	あり	0.779
互換性	SMO	名詞	あり	0.369
使用性	SMO	名詞	なし	0.666
信頼性	SMO	名詞	あり	0.504
セキュリティ	SMO	名詞と動詞	なし	0.755
保守性	SMO	名詞と動詞	あり	0.525
移植性	SMO	名詞と動詞	あり	0.430

る際に特徴的に使用される単語である可能性が高い．また,「機能適合性」は全要件文 2,916 件中 33 件のみが分類されている．したがって，学習データそのものの少なさが分類精度に影響を与えている可能性がある．実際，表 3 の結果から,「機能適合性」の分類精度はいずれのアルゴリズムにおいても実用性の観点で不十分であると言える．

以上より，本研究で比較したアルゴリズムの中では SMO アルゴリズムが日本語非機能要件分類に最も適していると考えられる．しかしながら，非機能要件の分類ごとに分類精度のばらつきが大きく，また，F 値が 0.70 を超えた「性能効率性」および「セキュリティ」についても実用の観点からはさらなる精度向上が必要である．今後は学習データを増やすなどともに，他の教師あり学習アルゴリズムの適用を検討する予定である．

6.2. 品詞を限定した場合の分類精度への影響

RQ2 では，品詞をすべて用いて分類を行った場合，名詞のみを用いた場合，名詞と動詞を用いた場合に着目して分類精度を比較した．表 4 より，すべての教師あり学習アルゴリズムにおいて F 値が高くなったのは，正規化を行った場合であった．また，分類に使用する品詞の違いと合わせた結果では，Naïve Bayes アルゴリズムの場合は正規化を行い，かつ品詞をすべて分類に用いた場合が最も F 値が高くなった．また，k-近傍アルゴリズムと SMO アルゴリズムは正規化を行い，かつ名詞のみで分類を行った場合が最も F 値が高い結果となった．

3 つの教師あり学習アルゴリズムのうち，最も分類精度が高い結果となった SMO アルゴリズムで，正規化を行い名詞のみを分類に使用した場合と，正規化を行い名詞と動詞を分類に使用した場合に着目すると動詞を取り除くことで分類精度が向上していることがわかる．名詞のみに限定した場合に最も分類精度が高くなった理由について考察するため，要件文内に出現する動詞の出現回数を分析する．表 7 にすべての要件文内に頻出する単語の上位 5 件とその出現割合を示す．

表 7 より,「する」および「する」の活用形である「し」と「さ」が多く出現していることが分かる．前節でも議論したように，動詞の「する」や「し」「さ」などは，どのような非機能要件においても出現すると考えられる．例えば,「制御する」を形態素解析し品詞ごとに分割した場合，名詞の「制御」と動詞の「する」の 2 つに分割される．一方で，英語の「制御する (control)」を形態素解析すると,「control」という 1 単語の動詞で表される．このように，動詞だけで特徴となりえる英語とは異なり，日本語ではすべての文で多く出現する「する」などの動詞を含むこととなり，動詞を加えた場合に分類精度が下がったと考えられる．日本語にはサ行変格活用やナ行変格活用があり，形態素解析すると名詞と動詞に分割される場合が多いため，今後は名詞のみに限定して分類精度の向上に取り組む予定である．

6.3. 分類精度が最も高くなった組み合わせ

本研究では，先行研究 [2] を参考に，日本語非機能要件の分類に適したアルゴリズムを調べるための評価実験を行った．[2] のデータセットは英語の要件文であるため，本研究では厚生労働省が公開している要件定義書，調達仕様書計 12 件から抽出した合計 2,916 件の要件文を人手により分類（ラベル付け）した．そのため，非機能要件の分類ごとにデータの数のばらつき大きく,「機能適合性」などデータの少ないものは分類精度が著しく低いものになっていると考えられる．今後は，[2] と同程度の規模のデータセットを用いて精度向上を図る必要があるが，現時点での知見をまとめるために，各非機能要件分類の分類精度（F 値）が最も高くなった場合の組み

合わせを表 8 に掲載する．

まず，アルゴリズムについては,「機能適合性」を除く 7 つの分類で SMO の分類精度が最も高い結果となった．先行研究 [2] および [10] においても同様の傾向が示されていることから，日本語非機能要件の分類においても SMO が有効であることがわかった．ただし，３つの教師あり学習アルゴリズムを試行した段階であり，別のアルゴリズムの適用についても今後検討する必要がある．

分類に用いる品詞については,「性能効率性」が品詞すべてを用いた場合,「機能適合性」「セキュリティ」「保守性」「移植性」は名詞と動詞を用いた場合,「互換性」「使用性」「信頼性」は名詞のみを分類に用いた場合が最も高い精度となった.「機能適合性」「セキュリティ」「保守性」「移植性」では名詞と動詞を用いた場合に最も精度が高くなるが，前節で考察したように今後データを増やした場合にサ行変格活用やナ行変格活用の問題が生じ，分類精度が低下することが予想されるため，実験により検証する必要がある．

単語の正規化の有無に関しては,「使用性」と「セキュリティ」を除いた 6 つの分類で単語の正規化を行うことが好ましい結果となった．ただし，表 4 より，単語の正規化の効果は限定的であることも見て取れる．今後は正規化の効果をより高めるための工夫が必要となる．

7. まとめと今後の課題

本研究では，日本語非機能要件の分類に適した教師あり学習アルゴリズムを明らかにするために，先行研究で用いられた教師あり学習アルゴリズムによる分類精度の比較を行った．厚生労働省の 3 種類のシステムの要件定義書をデータセットに用いて，k-近傍アルゴリズム，SMO アルゴリズム，Naïve Bayes アルゴリズムの分類精度の比較を行った結果，先行研究 [2] と同様に SMO アルゴリズムの分類精度が最も高い結果となった．ただし，データ数が少ない「機能適合性」は k-近傍アルゴリズムを用いて分類する方が精度が高かった．また，最も精度が高かった SMO アルゴリズムで分類する場合，分類に用いる単語を名詞のみにすることで精度が向上することが確認された．英語とは異なり，日本語では名詞と動詞を組み合わせることで動作を表現することがあるため，日本語非機能要件の分類においては名詞のみを用いた方が精度が高くなったと考えられる．単語の正規化に関しては，8 つのうち 6 つの分類で好ましい結果となった．

本研究で用いたデータセットは非機能要件ごとにデータ数にばらつきがあり，少数しかない非機能要件がある．また，学習に用いたシステムは 3 種類と少なく，異なる種類のシステムの要件定義書を分類した場合に，本実験の結果と異なる可能性がある．そのため，先行研究 [2] で用いられたデータセットと同程度の規模のデータセットで実験を行い，分類結果の一般性を確かめる必要がある．

また，先述の通り，決定木などの本実験で用いたアルゴリズム以外の教師ありアルゴリズムも用いて評価実験を行い，日本語非機能要件の分類に適した教師ありアルゴリズムを明らかにする予定である．

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] M. Glinz, “On non-functional requirements,” Proceedings of 15th IEEE International Requirements Engineering Conference (RE ’07), pp.21–26, 2007.

[2] J. Slankas and L. Williams, “Automated extraction of non-functional requirements in available documentation,” Proceedings of 1st International Workshop on Natural Language Analysis in Software Engineering (NaturaLiSE ’13), pp.9–16, 2013.

[3] L.M. Cysneiros, “Evaluating the effectiveness of using catalogues to elicit non-functional requirements,” Proceedings of Workshop em Engenharia de Requisitos (WER ’07), pp.107–115, 2007.

[4] Z.S.H. Abad, A. Shymka, S. Pant, A. Currie, and G. Ruhe, “What are practitioners asking about requirements engineering? an exploratory analysis of social q a sites,” Proceedings of 24th International Requirements Engineering Conference Workshops (REW ’16), pp.334–343, 2016.

[5] A. Borg, A. Yong, P. Carlshamre, and K. Sandahl, “The bad conscience of requirements engineering: An investigation in real-world treatment of

non-functional requirements," Proceeding of 3rd Conference on Software Engineering Research and Practice in Sweden (SERPS '03), pp.1–8, 2003.

[6] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc, "Automated classification of non-functional requirements," Requirements Engineering, vol.12, no.2, pp.103–120, 2007.

[7] V.S. Sharma, R.R. Ramnani, and S. Sengupta, "A framework for identifying and analyzing non-functional requirements from text," Proceedings of the 4th International Workshop on Twin Peaks of Requirements and Architecture (TwinPeaks '14), pp.1–8, 2014.

[8] W. Zhang, Y. Yang, Q. Wang, and F. Shu, "An empirical study on classification of non-functional requirements," Proceedings of 23rd International Conference on Software Engineering & Knowledge Engineering (SEKE '11), pp.444–449, 2011.

[9] A. Casamayor, D. Godoy, and M. Campo, "Identification of non-functional requirements in textual specifications: A semi-supervised learning approach," Inf. Softw. Technol., vol.52, no.4, pp.436–445, 2010.

[10] M. Riaz, J. King, J. Slankas, and L. Williams, "Hidden in plain sight: Automatically identifying security requirements from natural language artifacts," Proceedings of 22nd International Requirements Engineering Conference (RE '14), pp.183–192, 2014.

[11] A. Mahmoud, "An information theoretic approach for extracting and tracing non-functional requirements," Proceedings of 23rd International Requirements Engineering Conference (RE '15), pp.36–45, 2015.

[12] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to japanese morphological analysis," Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP '04), pp.230–237, 2004.

[13] 佐藤敏紀，橋本泰一，奥村　学，"単語分かち書き辞書 mecab-ipadic-neologd の実装と情報検索における効果的な使用方法の検討," 言語処理学会第 23 回年次大会発表論文集，pp.875–878，2017.

[14] J.C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Technical report, Microsoft Research, 1998.

[15] G.H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," Proceedings of of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI '95), pp.338–345, 1995.

[16] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," Proceedings of 26th International Conference on Neural Information Processing Systems (NIPS '13), pp.3111–3119, 2013.

[17] 鈴木正敏，松田耕史，関根　聡，岡崎直観，乾健太郎，"Wikipedia 記事に対する拡張固有表現ラベルの多重付与," 言語処理学会第 22 回年次大会発表論文集，pp.797–800，2016.