

日本語非機能要件の分類 手法の提案

オープンソースソフトウェア工学研究室

学籍番号：60206050

大東 誠弥

目次

- 背景
- 目的
- 既存研究
- リサーチクエスチョン
- データセット
- アプローチ
- RQの結果

背景

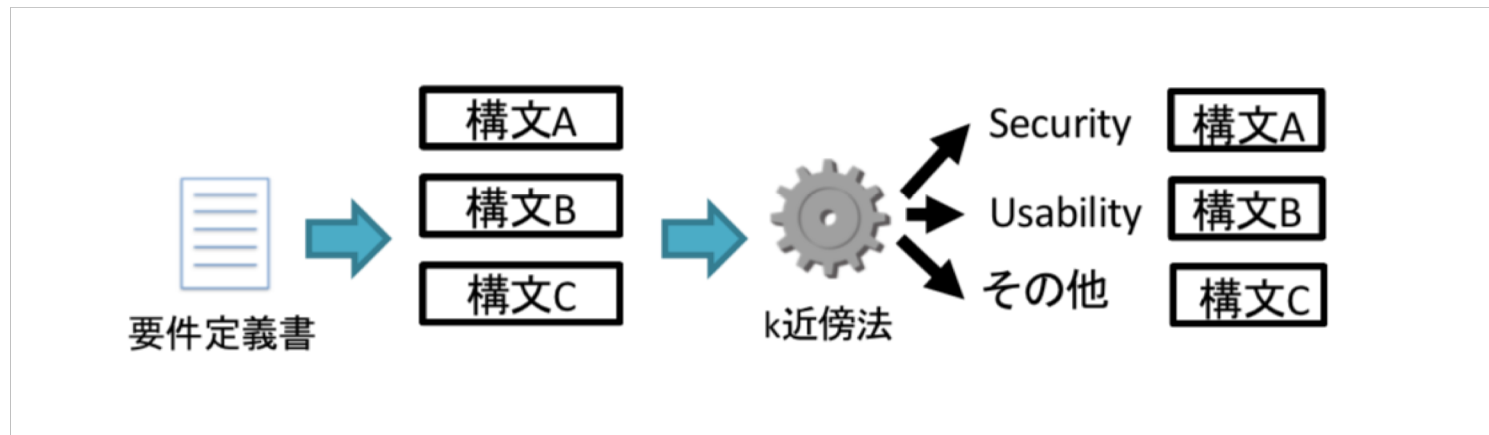
- 非機能要件は，要件定義書内に混同して記述され，目視で探す場合コストがかかる．
- 非機能要件は，記述者によって記述内容が異なり，受取手によって解釈が異なるという問題点がある．

目的

- 日本語で記述された要件定義書から，非機能要件の分類に適した，テキストマイニング手法を提案することを目的とする．

先行研究

- k近傍アルゴリズムを用いて、要件定義書(英語)内の非機能要件を分類している. [1]



[1] John Slankas and Laurie Williams, Automated extraction of non-functional requirements in available documentation, NaturaliSE '13, page 9-16

リサーチクエスション

- **RQ1**：日本語で記述された非機能要件の分類において、最も分類精度が良い分類アルゴリズムは何か.
- **RQ2**：分類に使用する品詞や語幹を統一することで、分類精度に違いは生じるか.
- **RQ3**：システムのドメインごとにごとに学習、分類した場合精度は良くなるか.

データセット

厚生労働省が公開している要件定義書計及び調達仕様書**12**件をデータセットとする。

関連するシステム	件数
ハローワーク関連システム	4件
労働保険関連システム	6件
年金関連システム	2件
合計	12件

データセット

ISO25010の品質特性に基づいて手作業で分類

機能適合性(functionality)
性能効率性(performance)
互換性(compatibility)
使用性(usability)
信頼性(reliability)
セキュリティ(security)
保守性(maintainability)
移植性(portability)

データセット

目視分類の一例

70	受託者以外の者が市場で調達することが可能であるサービスであること。	使用性	
71	受託者以外の者が市場で調達することが可能である製品であること。	使用性	
72	アプリケーションプログラムの動作、性能等に支障を来さない範囲において、オープンソースソフトウェア(OSS)製品(ソースコードが無償で公開され、改良や再配布を行うことが誰に対しても許可されているソフトウェア製品)の活用を図ること。	不明	
73	受託者は、利用するソフトウェアはサポート期間を考慮して選定し、ソフトウェアベンダーによるサポート又は他の事業者によるサポートサービスを必ず受けること。	不明	
74	本システムの特性及び基本方針書を踏まえ、効率的に設計・開発及び保守可能な手法を採用すること。	保守性	
75	本システムの利用者は、本省職員及び地方厚生局等職員合わせて約 910 名である。	機能適合性	
76	本システムを利用する拠点数は、本省 1 か所、地方厚生局等 47 か所及び地方厚生局 分室 2 か所である。	機能適合性	
77	検索、参照、登録、更新及び削除に係る処理については、業務の繁忙期においても平均処理応答時間(レスポンスタイム):3 秒以内を実現可能とすること。	性能効率性	
78	導入するハードウェアの選定に当たっては、過剰な設備投資により必要以上の性能を確保することなく、適切なハードウェアを導入すること。	性能効率性	
79	平均処理応答時間(レスポンスタイム):3 秒以内は、通常運用時に実現可能とし、縮退運転時はこの限りではない。	性能効率性	
80	可用性に係る指標は「稼働率」として目標値を 99.5%とする。	信頼性	
81	障害発生時等の可用性を確保するため、本システムを構成する各サーバ等を適切な方式で冗長化し、迅速かつ円滑な切り替えによる業務遂行を可能とすること。	信頼性	
82	データの紛失や改ざんからデータを保護し、データの正確性及びデータの一貫性を保証すること。	セキュリティ	
83	データベースについては、定期的に自動バックアップを行うこと。なお、バックアップの方法及び実施サイクルについては、受託者が担当職員と調整の上、決定すること。	不明	

データセット

- 品質特性ごとの各要件数

	ハロー ワーク	労働保険	年金	合計
機能適合性	12	12	8	32
性能効率性	43	94	71	208
互換性	19	17	28	64
使用性	45	14	1	60
信頼性	18	42	28	88
セキュリティ	87	81	88	256
保守性	69	166	116	351
移植性	20	22	15	57
合計	313	448	335	1116

アプローチ

- 分類手法の流れ

脆弱性/の/対策...

①要件定義書から抽出した1文を形態素に分割する.

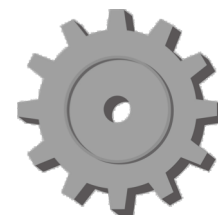
分類に用いる品詞を変える. また, 語幹抽出の有無を変える.



[0,1,1,...]

②分割した形態素をもとに文をベクトル化する.

ベクトル化手法を変える.



③教師あり学習のアルゴリズムを用いて分類する.

アルゴリズムを変える.

アプローチ

①形態素解析

オープンソース形態素解析エンジンである
MeCabを用いて1文を形態素に分割する.

脆弱性の対策をすること



形態素解析

脆弱/性/の/対策/を/する/こと

赤色:名詞, 青色:助詞, 緑色:動詞

アプローチ

Mecabの出力を用いて、分類に使用する品詞や、語幹抽出を行うかどうかを決定する。

脆弱性対策を行った。	
脆弱	名詞,形容動詞語幹,*,*,*,*,脆弱,ゼイジャク,ゼイジャク
性	名詞,接尾,一般,*,*,*,性,セイ,セイ
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
.	記号,句点,*,*,*,*,.,.,.

MeCabの出力例(「脆弱性対策を行った。」を形態素解析した場合)

アプローチ

②形態素をもとにベクトル化

2種類のベクトル化手法を用いる.

1. レーベンシュタイン距離を応用したベクトル化手法(先行研究)
2. Bag of Words (BOW) を用いたベクトル化手法

アプローチ

1.レーベンシュタイン距離を応用したベクトル化

- レーベンシュタイン距離

文字を置換，挿入，削除した回数を距離とする．

(例) light $\Rightarrow$ right (距離:1)

アプローチ

1.レーベンシュタイン距離を応用したベクトル化

- レーベンシュタイン距離の応用版

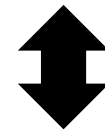
文字ではなく，単語単位の置換等で距離を算出．

(例) I have an apple. $\Rightarrow$ You have an apple. (距離:1)

アプローチ

文A 北海道は寒い

北海道/は/ 3/度/であっ/た



文B 沖縄は30度だ
(距離計算のルール)

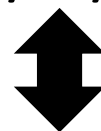
沖縄/は/ 4/度/暖かい/

条件	距離
両単語が同じ属性(固有名詞)の場合	0
単語が同じ場合	0
両単語が数字の場合	0
片方の単語が存在しない場合	1
上記以外はWordNetで単語の類似度を算出	0～1
それ以外	1

アプローチ

文A 北海道は寒い

北海道/は/3/度/であっ/た



文B 沖縄は30度だ

沖縄/は/4/度/暖かい/

距離：

0/0/0/0/ 0.6 /0

文Aと文Bの距離の合計は 0.6 となる

	文A	文B	文C
文A	0	0.6	4.2
文B	0.6	0	2.8
文C	4.2	2.8	0

アプローチ

2. Bag of Words (BOW)を用いたベクトル化手法
文ごとの単語の有無を行列で表す.

文A 不正/ログイン/対策

文B 脆弱/性/対策

	不正	ログイン	脆弱	性	対策
文A	1	1	0	0	1
文B	0	0	1	1	1

アプローチ

③教師あり学習アルゴリズムでの分類

使用する分類器

- K Nearest Neighbor (K近傍法)
- Support Vector Machine (SVM)
- Naïve Bayes (ナীবベイズ)

K近傍(レーベン)の分類精度

	Precision	Recall
機能適合性	0.000	0.000
性能効率性	0.661	0.597
互換性	0.429	0.188
使用性	0.545	0.333
信頼性	0.400	0.153
セキュリティ	0.421	0.558
保守性	0.407	0.457
移植性	0.111	0.117

K近傍(BOW)の分類精度

	Precision	Recall
機能適合性	0.444	0.400
性能効率性	0.632	0.693
互換性	0.333	0.250
使用性	0.818	0.500
信頼性	0.280	0.269
セキュリティ	0.590	0.597
保守性	0.546	0.562
移植性	0.364	0.471

SVM(BOW)の分類精度

	Precision	Recall
機能適合性	0.300	0.300
性能効率性	0.794	0.806
互換性	0.364	0.250
使用性	0.842	0.889
信頼性	0.385	0.385
セキュリティ	0.763	0.753
保守性	0.581	0.581
移植性	0.308	0.235

Naïve Bayes(BOW)の分類精度

	Precision	Recall
機能適合性	0.200	0.100
性能効率性	0.649	0.597
互換性	0.188	0.188
使用性	0.500	0.222
信頼性	0.370	0.385
セキュリティ	0.514	0.714
保守性	0.368	0.610
移植性	0.267	0.235

RQ1の結果

RQ1：日本語で記述された非機能要件の分類において，最も分類精度が良い分類アルゴリズムは何か．

	Precision	Recall
K近傍(レーベン)		
K近傍(BOW)	0.742	0.726
SVM(BOW)	0.732	0.750
Naïve Bayes(BOW)	0.768	0.488

RQ2の結果

RQ2：分類に使用する品詞や語幹を統一することで，分類精度に違いは生じるか．

(RQ1で最も精度が良かった分類器を採用予定)

	語尾統一	統一しない
名詞のみ		
動詞のみ		
名詞 + 動詞		
全部		

※Precision, Recallそれぞれ出す．

RQ3の結果

RQ3：システムのドメインごとにごとに学習，分類した場合精度は良くなるか．

(RQ1,RQ2それぞれで最も精度が良かった分類器，前処理を採用予定)

	Precision	Recall
ハローワーク		
労働保険		
年金		