

# 機械学習を用いた非機能要求の分類法の評価

2018SE008 古川 実歩 2018SE042 松尾 美穂 2018SE049 村瀬 摩由加  
指導教員 佐伯 元司

## 1 はじめに

### 1.1 研究背景

システム開発を行う上で、発注側(ステークホルダ)の要求を正しく理解し、獲得する事は重要である。機能要求や非機能要求といったように、要求は多様で複雑である為、要求を正確に把握する為には要求の範囲や性質に基づいて分類する必要がある[1]。

要求の分類を行う場合、人手による分類になる。しかし、人手による要求の分類は、多くの時間を費やし、経験豊富な分析者の暗黙知の部分が多いことや発注側の技術的な要素の理解不足などの理由から、分析者間や発注側と受注側で認識のズレが生じる。

その為、近年では、情報検索手法や機械学習を用いて、要求仕様書に書かれている要求文を自動で分類する研究が注目されている。中でも、機械学習を用いる手法は、有望である[2][3][4]。通常、要求仕様書は機能ごとに非機能要求が記述される為、要求仕様書内に非機能要求が散在する事が多い。その為、要求仕様書内の非機能要求を分類することによって以前は分類されなかった要求の発見や、次のアーキテクチャ設計などがしやすくなるという利点がある[2]。しかし、機械学習を用いた非機能要求の分類の精度は低いものが多い[3]。

### 1.2 研究課題

関連研究では精度の低い理由として学習させる際の分類項目データ数不均衡が報告されている。実際の要求仕様書では互換性など特定の非機能要求の数が少ない。そのため、要求仕様書から収集したデータは分類項目ごとにデータ数が不均衡になり、このまま機械学習に適用するとデータ数が少ない項目の分類が正しく行えない。また、使用しているベクトル化手法は様々である。よって以下の2点を研究課題とする。

- (1) 学習時の非機能要求特性の均衡化
- (2) 種々のベクトル化手法、その学習モデルの比較評価

## 2 準備

本章では本研究で使用する技術について説明する。

### 2.1 CNN(畳み込みニューラルネットワーク)

CNN(Convolutional Neural Networks)[5]とは畳み込み層、プーリング層、全結合層から構成される順伝播型ニューラルネットワークの一種である。CNNは主に画像認識で用いられている。更に、テキスト処理においても高い精度が報告されている[6]。

CNNの特徴は、局所受容野および重み共有と呼ばれる特別な層間結合をもつことである[5]。この特徴により、ある位置で学習に重要な特徴に関して、他の位置で同じような特徴が抽出された場合でも重要な可能性があるとして学習する。入力データの位置も考慮して特徴の抽出を行う為、文章中

に現れる分類を決定づける単語の抽出や、単語の流れを考慮した特徴の抽出に期待できる。その為、本研究では、CNNを用いた非機能要求の学習と分類を行う。

### 2.2 テキストのベクトル化

CNNによる非機能要求の分類の入力データとして、単語の特徴を数値化した固定長ベクトルが必要になる。これを取得する為に本研究では Word2Vec[7]と Doc2Vec[8]を使用する。Word2Vec と Doc2Vec は単語・テキストから教師あり学習を行い、ベクトル分散表現を獲得する。ベクトル分散表現は固定長ベクトルの為、CNNの入力データとして有用である。

#### 2.2.1. Word2Vec

Word2Vec とは、T.Mikolov[7]らによって開発され、文章中の単語からベクトル分散表現を獲得する手法である。

Word2Vec は入力層と出力層、1層の隠れ層で構成されるニューラルネットワークであり、単語や単語間の結びつきや類似度を学習するという特徴を持つ。学習モデルとして CBOW と Skip-gram の2つのモデルを提案している。

##### A) CBOW(Continuous Bag-of-Words)

入力層：対象の単語の周辺語

出力層：対象の単語の予測

##### B) Skip-gram

入力層：対象の単語

出力層：対象の単語の周辺語の予測

#### 2.2.2. Doc2Vec

Doc2Vec とは、Quoc Le[8]らによって開発された、Word2Vec を拡張したベクトル化手法である。Doc2Vec は文章やテキストからベクトル分散表現を獲得する事が可能であり、文章の流れを考慮して学習するという特徴を持つ。学習モデルとして PV-DM と PV-DBOW の2つのモデルを提案している。

##### A) PV-DM

入力層：対象の単語の周辺語+文書 ID

出力層：対象の単語の予測

##### B) PV-DBOW

入力層：文書 ID

出力層：文章中に登場する単語の予測

### 2.3 Word2Vec と Doc2Vec の比較について

非機能要求の分類に関して、分類項目ごとで特定の単語の出現が分類を決める上で重要であれば Word2Vec の精度が高くなると期待できるが、文章中の単語の出現順が分類を決定づける上で重要であれば Doc2Vec の精度が高くなると期待できる。その為、本研究では Word2Vec, Doc2Vec 両方を採用し、分類結果を比較したいと考えている。

## 3 関連研究

### 3.1 Doc2Vec と CNN による要求分析

関連研究として安齋[3]の研究では要求仕様書から機能

要求と非機能要求の分類を行っている。分類手法は形態素解析 (kuromoji) + Doc2Vec + CNN・MLP (Multilayer Perceptron: 多層パーセプトロン) を組み合わせて学習・分類をしている。使用データは要求仕様書 13 件、要求の種別は、機能要求は 4 つ、非機能要求は ISO25010 に基づき 8 つに分類している。

評価結果は「PV-DBOW+CNN」の組み合わせの分類正解率が高く、86%であった。しかし「PV-DBOW+CNN」での非機能要求の分類正解率が 45%と低い。安齋は理由として学習したデータ数、特に非機能要求に関する文章が足りないことを報告している。

### 3.2 Word2Vec と CNN による要求自動分類

関連研究として北川ら[4]の研究では、提案依頼書(RFP: Request for Proposal)から機能要求と非機能要求、その他要求の分類を行っている。分類手法は形態素解析(MeCab)+ Word2Vec+CNN を組み合わせて学習・分類をしている。使用データは RFP 71 件、要求の種別は、機能要求は 6 つ、非機能要求は非機能要求グレードから 6 つ、どちらでもないものとしてその他 2 つに分類している。

評価結果は全体の正解率として 62%、分類結果の上位 3 つ以内での一致が 83%となった。分類正解率は高いと言えるが、データ数が少ない項目に関しては分類精度が高いとは言えない。北川らは精度の向上の為にデータ数を均衡させる必要があると報告している。

## 4 アプローチ

本研究では、要求仕様書から非機能要求を抽出したのち、Doc2Vec や Word2Vec を用いてベクトル化を行い、CNN で非機能要求の分類モデルを作成し、それを用いて非機能要求の特性ごとに分類し精度値を算出するアプローチをとる。

関連研究の問題点解決や、ベクトル化する際にどのツールを使用すれば精度が向上するのかを比較する為に、本研究では以下の 2 つを行う。

- (1) 非機能要求分類項目ごとのデータ数の均衡性の保持
- (2) Word2Vec と Doc2Vec の精度比較

## 5 非機能要求の分類と評価プロセス

### 5.1 非機能要求の分類と精度算出プロセス

本研究での非機能要求の分類と精度算出プロセスを図 1 に示す。

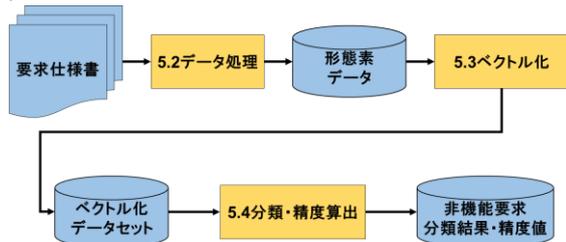


図 1 非機能要求の分類と精度算出プロセス

### 5.2 データ処理

そのままの要求仕様書ではデータセットに使用できないので要求仕様書に対して以下の 5 つの処理を行う。

- (1) 非機能要求抽出  
要求仕様書から、非機能要求を手作業で抽出する。
- (2) ラベル付け  
(1) で抽出した非機能要求を、ISO25010 [9]を基に「機能適合性」「性能効率性」「互換性」「使用性」「信頼性」「セキュリティ」「保守性」「移行性」の 8 つに分類し、文章の先頭に 0~7 のラベル番号を付与する。
- (3) データの再確認  
分類の属人性が低くなるよう 3 人で役割を決め、それぞれデータのラベル付けの再確認を行う。
- (4) 文章の正規化  
文章の正規化では、空白の削除、句読点や記号の削除、全角英数字を半角に変換、活用形を原形に変換を行う。
- (5) 形態素解析  
形態素解析器を用いて、非機能要求データを単語ごとに分ける。

### 5.3 ベクトル化

前処理で作成した形態素データから、配列化を行う。また、配列化を行った後でラベル分離によって、ラベルデータと文章データに分割する。作成した文章データから、Doc2Vec や Word2Vec を用いてベクトル化を次元数 300 次元で行い、文章ベクトル化データを作成する。最後にラベルデータと文章ベクトル化データを合わせて、ベクトル化データセットが作成する。ベクトル化プロセスを図 2 に示す。

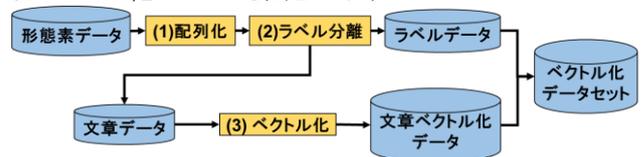


図 2 ベクトル化プロセス

以下に配列化、ラベル分離、ベクトル化処理の例を示す。

- (1) 配列化  
[[2][約][500][名][に対して][利用][可能][で][ある][こと]]
- (2) ラベル分離  
ラベルデータ  
[[2]]  
文章データ  
[[約][500][名][に対して][利用][可能][で][ある][こと]]
- (3) ベクトル化  
[約] : [[-6.14, 2.57, ..., 1.67]]  
[500] : [[1.14, 5.25, ..., -2.53]]  
:  
:  
[こと] : [[7.14, 4.57, ..., 2.61]]

### 5.4 分類・精度算出

ベクトル化で作成したベクトル化データセットを、学習データとテストデータの 2 つに分割する。次に、学習データを入力とし、CNN で教師あり学習を行い、非機能要求の分類モデルを作成する。そこに、ラベル付けされたテストデータを入力し、非機能要求分類を行い、出力された分類項目とラベルを照合し、精度値算出や分類モデルが予測した分類項目を 1 位から 3 位までを表示する。

## 6 プロトタイプの実装

### 6.1 実装環境

本研究の分類法を実装するプロトタイプの実装環境を以下の表1に示す。

表1 実装環境

コンポーネント	コンポーネント名	バージョン
OS	Ubuntu	18.04
実装言語	Python	3.8.8
形態素解析器	Mecab	0.996.3
配列化ツール	numpy	1.21.1
ベクトル化ツール	genism	3.8.3
機械学習フレームワーク	Chainer	7.8.0

### 6.2 プロトタイプの構成

本研究でのプロトタイプの構成を図3に示す。

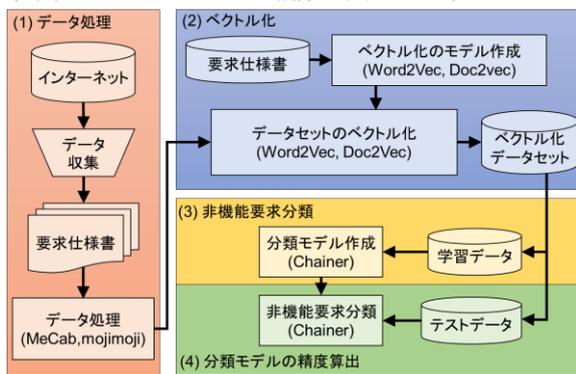


図3 プロトタイプの構成

## 7 実験と評価

### 7.1 適用するデータセット

データ数の均衡を保つことで学習モデルの精度が向上するかを検証するため、(1) データ数が不均衡なデータセットと(2) データ数が均衡しているデータセットの2つを用意した。2つのデータセットの内訳を表2,表3に示す。

表2から明らかのように、(1)は互換性のデータ数が最も少なく、最もデータ数が多い項目と約300個以上データ数の差があり、項目ごとにデータ数が不均衡である。(2)はデータ数が少ない分類項目に学習データと同じ文章を追加した。

表2 データ数が不均衡なデータセット

分類項目	内訳
機能適合性	352
性能効率性	523
互換性	248
使用性	452
信頼性	492
セキュリティ	581
保守性	588
移植性	375
総計	3,611

表3 データ数が均衡しているデータセット

分類項目	内訳
機能適合性	567
性能効率性	523
互換性	481
使用性	533
信頼性	533
セキュリティ	581
保守性	588
移植性	545
総計	4,351

### 7.2 k-分割交差検証(k-fold cross validation)

k-分割交差検証は、分類モデルの精度を正確に検証するための評価方法である。検証方法はデータをk個に分割してそのうち1つをテストデータ、残りのk-1個を学習データとして、学習を行う。分割されたデータすべてがテストデータになるようk回学習を行い、出力された精度の平均を分類モデルの正確な精度とする。本研究では、分割回数k=5で行った。

### 7.3 比較実験の方法

本研究での分類法による精度向上を図るため、使用データと使用ツールを変更して比較実験を行う。比較実験は以下の2つを行う。比較実験の方法を図4に示す。

比較実験①:

2つのデータセットをプロトタイプに適用し、交差検証で得られた分類モデルの精度比較を行い、データ数を均衡させることで精度を上げることができるか検証する。

比較実験②:

データセットのベクトル化を行う際、Word2VecのCBOWモデル、Skip-gramモデルとDoc2VecのPV-DMモデル、PV-DBOWモデルの4種類をそれぞれ使用し、交差検証で得られた分類モデルの精度比較を行い、データ数を均衡させることで精度を上げることができるか検証する。

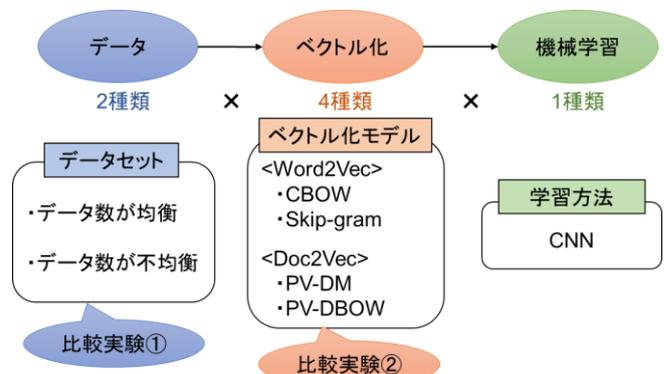


図4 比較実験の方法

## 7.4 比較実験

k-分割交差検証を用いて分類モデルの精度を評価し、比較実験のために8パターンの適用結果を出力した。8パターンの適用結果のまとめを表4に示す。

表4 8パターンの適用結果

ベクトル化モデル	データセット	学習データでの精度	テストデータでの精度
CBOW	不均衡	63.6%	47.9%
	均衡	64.2%	48.4%
Skip-gram	不均衡	76.9%	62.0%
	均衡	77.5%	64.0%
PV-DM	不均衡	74.3%	53.2%
	均衡	76.4%	57.5%
PV-DBOW	不均衡	72.4%	58.5%
	均衡	75.5%	60.5%

## 7.5 結論

比較実験より、データ数が均衡しているデータセットを使用データとし、Word2VecのSkip-gramモデルをベクトル化モデルとする組み合わせが、最も高い精度を得ることができた。本研究では分類モデルの精度は最大64%となり、分類項目や使用データは異なるものの関連研究[3][4]よりも高い精度を得ることができた。

## 8 考察

### 8.1 研究課題に対する考察

#### 8.1.1. 提案した分類法の比較実験

実験結果より、データ数が少ない分類項目にデータを追加することで、分類に必要な情報が増え、より正確に分類されるようになったと考えた。また、ベクトル化モデルを用いた機械学習の分類においては、文章の特徴よりも文章中の出現単語が重視されると考えた。

#### 8.1.2. 分類モデルの正確性

分類モデルの正確性を調査するため、本研究で作成した分類モデルにテストデータを入力し、予測される分類項目を1-3位までを分類結果として出力した。分類結果の抜粋を表5に示す。

表5 分類結果の抜粋

テストデータ	正解ラベル	分類結果
本システムへの不正ログインを防止する機能を有していること。	セキュリティ	1:セキュリティ 2:保守性 3:使用性

テストデータ全60個を使用したところ、37個が分類結果の1位と正解ラベルが完全一致した。よって、分類項目より得られた精度は61.6%となった。この精度は分類モデル作成時に得られた精度と近似しているため、本研究で得られた分類モデルは正しく学習できていたことが分かる。

### 8.2 結論の妥当性

(1) 内的妥当性への脅威：比較実験①で得られた結果は、「データ数の均衡」か「データ数の増加」のどちらの要因が精度を向上させたのか必ずしも明確でない。

また、手作業による分類は、全員でラベル付け・話し合いを行った。そのため分類結果が偏ることなく、属人性の低い分類結果を得ることができたと思われ、ラベル付けの質による内的妥当性の脅威は少ないと考えられる。

(2) 結論妥当性への脅威：データ数の均衡化によって向上した精度は0.5~4.3%と比較的少ないため、7.5節の結論を主張するには、さらなる分析が必要と思われる。

## 9 今後の課題

(1) データセットの改善

本研究で抽出された非機能要求数はデータ数が少なく、分類項目ごとの偏りも見られた。今後、よりデータ数を増加させて評価を行う必要がある。また8.2節(1)の脅威に対処するため、本手法以外のデータ数の均衡化方法を用いて評価を行う必要がある。

(2) 実際のデータ適用による評価

本研究の分類法が実際のシステム開発において、どの程度有用であるか分からない。そのため、実務者に実際のシステムの非機能要求を分類してもらい、本研究の分類法の有用性を評価する必要がある。

## 10 まとめ

本研究では機械学習を用いた非機能要求の分類法の評価を行った。提案方法のプロトタイプを実装し、データ数の均衡とベクトル化モデルによる比較実験を行った。これより、Skip-gramモデルとデータ数が均衡しているデータセットの組み合わせが、分類モデルを作成する上で最も高い精度を得た。本研究の分類法を用いることで、システム開発における非機能要求の分析が容易になることが期待できる。

## 11 参考文献

- [1] 一般社団法人情報サービス産業協会 REBOK 企画 WG, 要求工学知識体系, 近代科学社, 2011
- [2] Jane Cleland-Huang, Raffaella Settini, Xuchang Zou and Peter Solc: Automated classification of non-functional requirements, Requirements Eng 12, 103–120, (2007)
- [3] 安齋 太地, 深層学習を用いた要求分析に関する研究, 法政大学学術機関リポジトリ, (Mar. 2017)
- [4] 北川貴之, 長岡 武志, 深層学習による要求仕様自動分類手法の提案と評価, 情報処理学会論文誌, Vol.61, No.4, p.842–852, (Apr. 2020)
- [5] 岡谷 貴之, 深層学習, 講談社, 2015.
- [6] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu: Neural machine translation in linear time. The Computing Research Repository (CoRR), arXiv:1610.10099, (2016)
- [7] T. Mikolov, K. Chen, G. Corrado and J. Dean: Efficient Estimation of Word Representations in Vector Space, Proc. International Conference on Learning Representations 2013 (ICLR) Workshop Track, arXiv: 1301.3781, (2013)
- [8] Quoc Le and Tomas Mikolov: Distributed representations of sentences and documents, arXiv preprint arXiv:1405.4053v2, (2014)
- [9] ISO/IEC 25010:2011, <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>