

深層学習による要求仕様自動分類手法の提案と評価

北川 貴之^{1,a)} 長岡 武志^{1,b)}

概要: 情報システムの要求定義においては、ステークホルダの要求を的確に把握することが重要である。このステークホルダの要求を記述した文書として要求仕様書がある。要求仕様の分析者は、要求仕様書に記述された要求仕様の性質を明らかにし、要求仕様が重複、矛盾、漏れがないことを確認するため、要求仕様の分類を行う。要求仕様書から要求仕様を抽出し分類するノウハウは、経験豊富な分析者の暗黙知となっている場合が多いため、経験が少ない分析者では、要求仕様書からの要求仕様の抽出に多大な時間を要したり、分類の結果に一貫性を欠くことがある。これまでに著者らは、非機能要求に着目し、要求仕様書が非機能要求のどの種別に属するのかを深層学習を用いて自動分類する手法を試行した。本研究では、要求仕様の自動分類の範囲を機能要求と非機能要求の両方に拡張した深層学習による要求仕様の自動分類手法を提案する。提案手法の有効性を確認するため、提案手法に基づき作成した分類モデルの精度を評価する実験を行った。

1. はじめに

情報システムの要求定義においては、ステークホルダの要求を的確に把握することが重要である。このステークホルダの要求を記述した文書として要求仕様書がある。要求仕様書には、要求仕様に加え、システムの目的や概要、制約事項が記述されている。要求仕様の分析者は、要求仕様書に記述された要求仕様の性質を明らかにし、要求仕様が重複、矛盾、漏れがないことを確認するため、要求の分類を行う [1]。文献 [1] では、要求仕様を機能要求、非機能要求に分類し、さらに非機能要求を種別毎に分類する方法が示されている。例えば、ある開発現場では、要求仕様の 1 文 1 文が、可用性、性能・拡張性など、非機能要求のどの種別に当てはまるかを人手作業で分類し、要求仕様に重複、矛盾、漏れがないことを確認している。このような要求仕様の分類に関するノウハウは、経験豊富な分析者の暗黙知となっている場合が多く、経験が少ない分析者では、要求仕様書からの要求仕様の抽出に多大な時間を要したり、分類の結果に一貫性を欠くことがある。

要求工学の研究コミュニティでは、要求仕様の自動分類に関する研究が注目を集めており文献 [2][3][4][5][6]、特に文献 [3][4][5][6] では、機械学習による要求仕様の自動分類を試みている。文献 [3] では、非機能要求を自動分類の対象としているが機能要求やそれ以外の文章は対象とし

ていない。文献 [4][5] の手法は、英語で作成された要求仕様を対象としており、日本語で作成された要求仕様に対しては、同様の性能が出るかは定かではない。また文献 [6] は、日本語で作成された要求仕様を対象としているものの、実案件の要求仕様への適用は行われていない。

著者らは文献 [7] において、深層学習を用いた要求仕様の自動分類を試行した。文献 [7] では、提案依頼書 (Request For Proposal:RFP) に含まれる文章を対象としているが、非機能要求やプロジェクト管理に関する文章しか扱うことができず、RFP に含まれるそれ以外の文章は分類することができなかった。本研究では、自動分類の対象に機能要求やそれ以外の文章も加え、RFP に含まれる全ての文章を対象とした要求仕様の自動分類手法を提案する。深層学習のモデルには、文献 [7] と同様に畳み込みニューラルネットワーク (Convolutional Neural Network:CNN) を利用する。モデルの作成および評価のデータには、インターネット経由で取得可能な官公庁の RFP を用いる。RFP は、表 1 に示すように中央省庁や都道府県など複数のドメインから収集した。

一般的に機械学習では、学習時と推論時でのデータの分布が同じであることを仮定しており、この仮定が崩れると (機械学習を利用した) システムの精度が落ちてしまうとされている [8]。文献 [7] では、調達元のドメインが同じ RFP は、含まれる要求仕様の分布が類似していると仮説を立て、この仮説を検証するための評価実験を行った。評価実験より、分類対象が要求仕様の非機能要求やプロジェクト管理に関する文章においては、RFP のドメインの違

¹ 東芝デジタルソリューションズ (株)
TOSHIBA DIGITAL SOLUTIONS CORPORATION

^{a)} Kitagawa.Takayuki@toshiba-sol.co.jp

^{b)} Nagaoka.Takeshi@toshiba-sol.co.jp

表 1 収集した RFP の一覧

ドメイン	ファイル数	文章数
中央省庁	12	10,887
独立行政法人	3	1,145
地方支分部局	12	4,695
市町村	11	4,903
都道府県	9	4,566
公法人	13	2,351
一般財団法人	3	223
準官公庁	4	201
地方公共団体	1	97
総計	71	29,068

いは分類の精度にそれほど大きい影響を与えないという結果を得ている。ただし、適用対象を RFP に含まれるすべての文章に拡大した場合においても、同様の結果となるかは定かではない。また、分類対象の種別を増やしたことにより、分類の精度が低下することが予想される。本研究では、この課題を解決するため、以下のような研究設問を設定した。

RQ1: 分類対象を RFP に含まれるすべての文章に拡大した場合、RFP の調達元のドメインの違いが分類精度にどの程度影響を与えるか。

RQ2: 分類対象を RFP に含まれるすべての文章に拡大した場合、文献 [7] に比べ分類精度に影響があるか。

これらの研究設問に対する解を得るため、提案手法を実装し、評価実験を行った。

以降の本稿の構成を示す。2 章では、要求仕様の分類、および、深層学習を適用した要求仕様の分類の関連研究について述べる。3 章では、提案手法について述べる。4 章では、評価実験について述べる。5 章では、考察について述べる。6 章では、本研究の結論を述べる。

2. 関連研究

2.1 要求仕様の分類手法

山田ら [2] は、非機能要求の中でも信頼性に着目し、自然言語で記述された要求仕様書から信頼性に関する要求仕様(信頼性要求)を自動抽出し、要求の無矛盾性、非冗長性、非曖昧性、完全性を検証する手法を提案している。本手法では、予め信頼性を特徴づける用語を特定しておき、特定した用語をキーワードとして利用することで要求仕様の中から、信頼性要求を自動抽出する。特定した用語によって正しく信頼性要求を抽出できている例はあるものの、本手法では自然言語の文脈を考慮することができないため、誤検出が多いことが課題として報告されている。

2.2 機械学習を適用した要求仕様の分類

機械学習を適用した要求仕様の分類手法として文献 [3][4][5][6] が報告されている。大東ら [3] は、日本語

で作成された非機能要求の分類に適した機械学習のアルゴリズムを明らかにするため、先行研究で用いられた 3 種類のアルゴリズム (k-近傍法, Sequential Minimal Optimization(SMO), Naïve Bayes) の比較評価を実施した。大東らの比較評価では、日本語で作成された非機能要求の分類において、SMO を用いた場合が最も分類精度が高いと報告されている。文献 [3] では、非機能要求の自動分類を扱っているのに対し、本稿で提案する手法では、非機能要求以外にも、機能要求やプロジェクト管理も扱っており、自動分類の対象が異なっている。また、文献 [3] で比較対象としているアルゴリズムには、深層学習を用いたアルゴリズムが含まれていない。

Dekhtyar ら [4] は、Word2Vec と CNN を用いた要求仕様の分類手法を提案した。Dekhtyar らの手法は、要求仕様を 2 種類の種別(「機能要求か非機能要求か」または「セキュリティかセキュリティではないか」)に分類する。Kurtanovic ら [5] は、サポートベクタマシン (Support Vector Machine) を用いた要求仕様の自動分類手法を提案している。Kurtanovic らの手法は、要求仕様の 2 種類の種別への分類(「機能要求か非機能要求か」と、非機能要求の 4 種類の種別 (Usability, Security, Operational, Performance) への分類を自動化する。

文献 [4][5] は、英語で作成された要求仕様の分類を対象としており、日本語で作成された要求仕様に対しては、同様の性能が出るかは定かではない。また、Dekhtyar らの手法は「機能要求か非機能要求か」、「セキュリティかセキュリティではないか」といった 2 値の分類を扱っているのに対し、本稿では 3 種類以上の要求仕様を分類する多クラス分類を扱っており、扱う問題の対象が異なっている。

松田ら [6] は、要求仕様の変更要因の一つである環境の変化に着目し、自然言語で記述された環境の変化に関する文章を、機械学習 (深層学習) のアルゴリズムの一種であるリカレントニューラルネットワーク (Recurrent Neural Network:RNN) を拡張した LSTM(Long Short Term Memory) により非機能要求の 8 種類の種別 (機能性, 信頼性, 使用性, 効率性, 保守性, 移植性, セキュリティ, コスト) に分類する手法を提案した。ここでの環境とは、市場動向や法令などの外部環境と、組織・事業内におけるポリシーやガイドラインなどの内部環境を指す。松田らの手法では、訓練データの数が数百件程度と少なく、分類に必要な単語が不足していたため、期待通りの分類ができなかったという課題が報告されている。

3. 提案手法

著者らは文献 [7] において、深層学習を用いた要求仕様の自動分類を試行した。文献 [7] では分類対象の要求仕様が非機能要求やプロジェクト管理に関する文章のみであり、RFP に含まれるそれ以外の文章は分類することができな

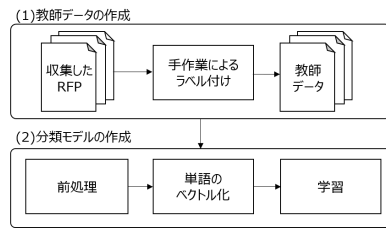


図 1 提案手法の全体像

かった。本稿では、適用対象を RFP に含まれる全ての文章に拡大し、深層学習を利用した要求仕様の自動分類手法を提案する。深層学習で利用するニューラルネットワークには、文献 [7] と同様に深層学習のアルゴリズムの一種である CNN を利用した。

深層学習を利用して要求仕様を自動分類するためには、予め要求仕様がどの種別であるのかラベルを付与したデータ（本稿では要求仕様にラベルを付与したデータを、機械学習の学習時に使用する「訓練データ」という名称と区別して「教師データ」と呼ぶこととする）を用意する必要がある。文献 [7] ではインターネット経由で取得可能な官公庁の RFP を用いて教師データを作成しており、本稿でも同様に教師データを作成した。

図 1 に提案手法の全体像を示す。提案手法は、収集した RFP からの教師データの作成と、深層学習を用いた分類モデルの作成からなる。提案手法の処理の流れを以下に示す。

- (1) 教師データの作成：インターネット経由で取得した RFP のテキストを 1 文単位で分割し、それぞれの文に対しラベル付けを行う。
- (2) 分類モデルの作成：教師データを入力とし、ニューラルネットワークで特徴を学習する。

3.1 教師データの作成

要求仕様に関する教師データは、インターネット経由で取得可能な官公庁の RFP を用いて作成した。収集した RFP の一覧を表 1 に示す。教師データは、RFP に含まれる要求仕様の文に対して、要求仕様の種別を手作業でラベル付けすることで作成する。以下に教師データの作成の流れを示す。

- (1) インターネット経由で官公庁の RFP を収集し、図以外の部分をテキスト化する。
- (2) テキストを句点（。）の単位で 1 文に分割する。箇条書きと表は 1 行を 1 文とする。
- (3) 1 文毎に下記に示す要求仕様の種別をラベルとして付与する。

表 1 の「文章数」は教師データ作成の作業によって分割された文章の総数を示している。全文章の文字数の平均値は 37.5、中央値は 29.0、最大値は 4,753、最小値は 1 であった。

表 2 定義した機能要求の種別

種別	内容
機能	システムの振舞いに関する要求
画面・帳票	システムの画面・帳票に関する要求
データ	システムのデータモデルに関する要求
外部インタフェース	他システムとの間で発生するデータのやりとりに関する要求
環境	システムのハードウェア、ミドルウェア、OS、ネットワーク等に関する要求
業務	新しい業務の在り方や運用をまとめた上で、業務上実現すべき要求

表 3 定義した非機能要求の種別

種別	内容
可用性	システムを継続的に利用可能とするための要求
性能・拡張性	システムの性能、および将来のシステム拡張に関する要求
運用・保守性	システムの運用と保守に関する要求
移行性	現行システム資産の移行に関する要求
セキュリティ	システムの安全性の確保に関する要求
システム環境・エコロジー	システムの設置環境やエコロジーに関する要求

定義した機能要求の種別の一覧を表 2 に、非機能要求の種別の一覧を表 3 に示す。「機能要求」は、機能要件の合意形成ガイド [9] の技術領域を参考に「機能」、「画面・帳票」、「データ」、「外部インタフェース」の 4 種類を定義した。なお、機能要件の合意形成ガイド [9] の「システム振舞い」と「バッチ」をまとめて「機能」、「画面」と「帳票」をまとめて「画面・帳票」とした。これに、RFP の記述内容から「環境」、「業務」の 2 種類を追加した。「環境」とは、システムのハードウェア、ミドルウェア、OS、ネットワーク等に関する要求であり、非機能要求のシステム環境・エコロジーとは異なる点に注意されたい。機能要求の例として、例えば「QA システムを利用するためにユーザ情報の登録を行う。」*1 は機能、「購入者が、所望のペットを選択し、ショッピングカートの内容を表示する画面」*2 は画面・帳票とラベル付けする。

非機能要求には、非機能要求グレード [10] からシステム基盤をスコープとした「可用性」、「性能・拡張性」、「運用・保守性」、「移行性」、「セキュリティ」、「システム環境・エコロジー」の 6 種類を定義した。非機能要求の例として、例えば「オンラインデータを時間当たり、xxxxx 件を処理できること」*3 は性能・拡張性、「本システムの運用時間は、原則として勤務時間内とするが、繁忙期などにおいては、可能な限り 24 時間運用が行えること。」*3 は運用・保守性とラベル付けする。

RFP には、機能要求または非機能要求のどの種別にも

*1 IPA:機能要件の合意形成ガイド～システム振舞い編～,
<https://www.ipa.go.jp/files/000004525.pdf>

*2 IPA:機能要件の合意形成ガイド～画面編～,
<https://www.ipa.go.jp/files/000004521.pdf>

*3 IPA:情報システム調達のための技術参照モデル(TRM)調達仕様書ひな形編, <https://www.ipa.go.jp/osc/trm/trmtemplate.html>

該当しないものがある。これらには「プロジェクト管理」, 「対象外」のどちらかのラベルを付けた。「プロジェクト管理」とは, 開発ドキュメントや開発体制, テスト方針など, 開発プロジェクトを成功裏に完了させるために発注者が開発プロジェクトに求める要求である。「対象外」は, 機能要求, 非機能要求, プロジェクト管理のどれにも該当しないものであり, 要求仕様の分析を行う技術者にとって不要な情報である。「対象外」とラベルを付けたものの例としては, 入札の条件, 日付情報, 発注者の業務拠点の住所や担当者情報などが挙げられる。

文献 [7] では, 主に非機能要求, プロジェクト管理に関する要求仕様のみを学習対象としていたため, 推論時にそれ以外の文章を入力とすることができなかった。本稿ではそれ以外の文章も学習対象としており, RFP に含まれる全ての文章を推論時の入力とすることができる。また, 分析の対象外であるものを「対象外」とラベル付けしたことにより, 要求仕様の分析を行う技術者が不要な情報を, 推論時に「対象外」としてフィルタリングすることもできるようになる。

3.2 分類モデルの作成

3.2.1 前処理

要求仕様の文に対し, 形態素解析器を用いて単語に分割した。形態素解析器には, MeCab[11] を利用した。単語への分割は, 後述するニューラルネットワークが入力を単語レベルで扱うためである。次に単語に分割した文に対し以下の正規化を行った。

- アルファベットを小文字化する。
- 数字を全て 0 に置き換える。
- 単語の活用形を原形に戻す。

アルファベットの小文字化は, 単語を文字種の区別なく同一の単語として扱うためである。数字については, 分類で利用する可能性がないことから 0 に置き換えた。単語の活用形を原形に戻す処理は, 語彙数を減らすために行った。

上記の正規化を行った文に対し, ストップワードの除去を行った。ストップワードとは, それ自体が意味を持たない, 一般的で役に立たない, というような理由で自然言語処理の対象から除外する単語のことである。提案手法では, 以下をストップワードとし除去した。

- MeCab によって助詞, 助動詞, 接続詞, 接頭詞, 連体詞, 記号と判断された単語。
- 文頭の数字。例えば箇条書きの 1), 2) など。

例えば「オンラインデータを時間当たり, xxxxx 件を処理できること。」*3(「xxxxx」は適当な数値に置き換えたとする) という要求は前処理によって「オンライン」, 「データ」, 「時間」, 「当たる」, 「0」, 「件」, 「処理」, 「できる」, 「こと」という単語の列に変換される。

表 4 Word2Vec モデルのハイパーパラメータ

ハイパーパラメータ	値
次元数	100
単語の出現頻度の閾値	2
ウィンドウサイズ	10

表 5 要求仕様分類モデルのハイパーパラメータ

ハイパーパラメータ	値
テキスト中の単語数	100
ベクトル次元数	100
フィルタサイズ	2,3,4,5
ストライド	1
フィルタ数	100
活性化関数	ReLU
活性化関数 (出力部分)	Softmax
全結合層の次元数	200
Dropout	0.5
L2 ノルムの最大値	3
最適化アルゴリズム	Adam
学習率	0.001
損失関数	Categorical Crossentropy
バッチサイズ	100
エポック数	50

3.2.2 単語のベクトル化

前処理を行った文章をニューラルネットワークに入力するためには, 文章中に含まれる単語をベクトルとして表現する必要がある。ベクトル表現の次元数削減のため, 提案手法ではベクトル表現への変換処理に Word2Vec[12] を利用した。Word2Vec とは, 自然言語のコーパス (自然言語処理等の目的で収集されたテキストデータ) を訓練データとして学習したニューラルネットワークを利用し, コーパスにある単語を入力するとその単語の特徴をベクトルとして出力するツールである。Word2Vec モデルの学習には日本語版 Wikipedia のデータを自然言語のコーパスとして利用した。Word2Vec モデルのハイパーパラメータは表 4 の通りである。Word2Vec は Python のライブラリである gensim[13] の実装を利用した。

3.2.3 学習

本節では要求仕様の文章テキストを分類するためのニューラルネットワークの構造について示す。ネットワークの構造は文献 [7] と同様であり, 文献 [14][15] に示されている CNN を利用した構造をベースに, 実験を繰り返しながら改善を加えた。図 2 にニューラルネットワークの構造を示す。

ネットワークは大きく分けて, (1) 入力層, (2) Convolution-Pooling 層, (3) 出力層から構成される。

「(1) 入力層」では, 要求仕様テキストに対して前処理, ベクトル化を適用したデータを入力として受け付ける。入力データの次元は「テキスト中の単語数」×「ベクトル次元数」となる。「テキスト中の単語数」は入力となるテキストに含まれる単語数の最大値 (前処理によって除かれた単語は単語数のカウントから除く) であり, 表 5 のように設

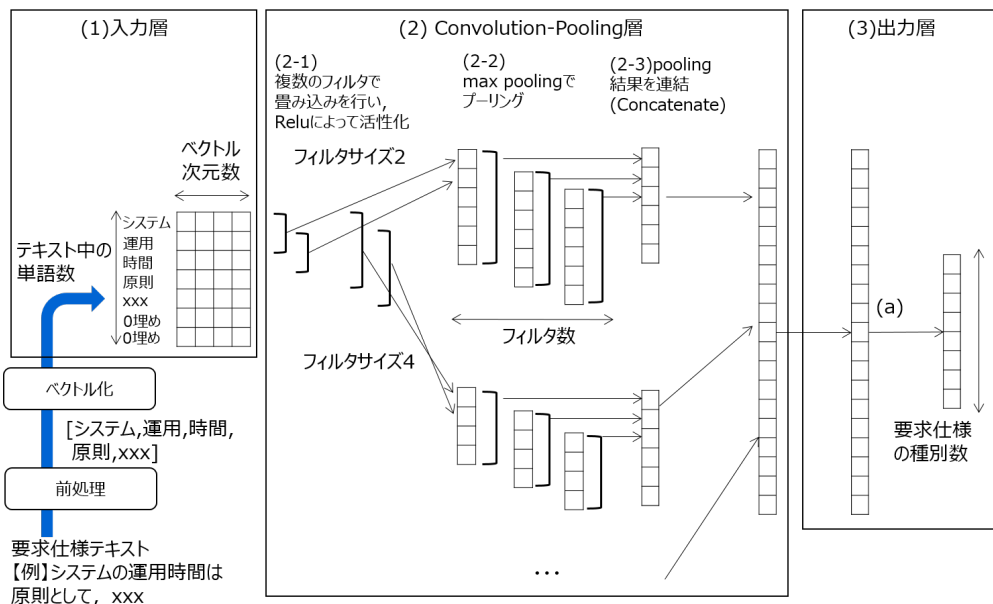


図2 要求仕様分類モデルのニューラルネットワーク構造

定した。設定した単語数に満たないテキストについては、残りの単語を0で埋め、設定した単語数を超えるテキストについては、超えた部分の単語を切り捨てた。「ベクトル次元数」は表4に示した次元数である。

「(2)Convolution-Pooling層」では、CNNによる畳み込みおよびPoolingを行う。構造のベースは文献[15]に示されているものであり、入力データに対し、異なるサイズのフィルタを並行で適用し、畳み込みを行う(図2の(2-1))。活性化関数にReLUを、PoolingにはMax Pooling(図2の(2-2))を利用する。最後にそれぞれのフィルタからのPooling結果を連結(Concatenate)する(図2の(2-3))。

「(3)出力層」には、全結合層を2層を挿入している。出力部分の活性化関数には、一般的に3種類以上の種別の分類(多クラス分類)において利用されるSoftmax関数を利用し、要求仕様の種別を分類する。

図中では省略しているが、過学習を抑制するため「(1)入力層」と「(2)Convolution-Pooling層」の間、および「(3)出力層」の1層目の全結合層の後(図2の(a)の部分)にそれぞれDropout層を挿入している。同様に過学習の抑制のため、(2-1)の畳み込み層と(3)の全結合層にはL2ノルム制約を課している。学習の速度を上げるため、(2-1)の活性化関数の手前にBatch Normalization層を挟んでいる。

表5に要求仕様分類モデルのハイパーパラメータを示す。

学習モデルはKeras[16]を用いて実装した。Kerasのバックエンドとなる深層学習のエンジンにはTensorFlow[17]を利用した。

4. 評価実験

1章で掲げた研究設問に対する解を導くため、3節で示した手法を実装し、評価実験を行った。

表6 要求仕様の種別の内訳

種別	数	種別	数
機能	1,515	性能・拡張性	502
画面・帳票	899	運用・保守性	2,627
データ	448	移行性	602
外部インタフェース	561	セキュリティ	1,519
環境	2,065	システム環境・エコロジー	747
業務	399	プロジェクト管理	9,306
可用性	463	対象外	7,415
計			29,068

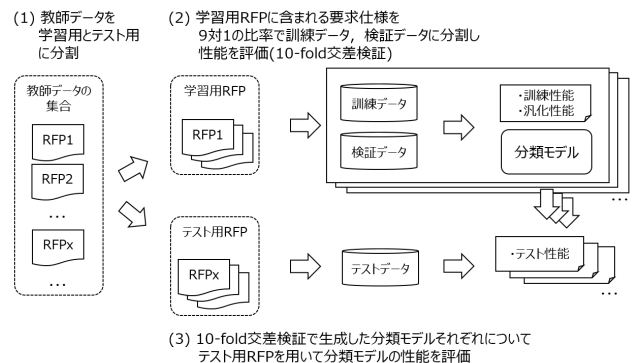


図3 評価実験の手順

実験で使用する教師データは、文献[7]で作成したものをベースに機能要求や対象外のラベルを新たに追加した。非機能要求についても教師データを見直し、データのクレンジングを行った。表6はラベル付けを行った要求仕様の種別の内訳である。

4.1 実験手順

図3に評価実験の手順を示す。未知の(学習時に利用していない)RFPに対する性能を評価(テスト)できるように、教師データに含まれるRFPを学習用RFPとテスト用

RFP に分割することとした (図 3(1)). 学習用 RFP は分類モデルの生成に使用し, テスト用 RFP は分類モデルの評価に使用する. 学習用 RFP/テスト用 RFP の選び方は 4.2 節に示す実験内容に応じて指定する.

図 3(2) では, 学習用 RFP 内の要求仕様を訓練データ, 検証データに分割し学習を行う. 本稿では訓練データに対する性能を訓練性能, 検証データに対する性能を汎化性能と呼ぶこととする. 性能の評価指標には, 精度 (Accuracy) と誤差 (Loss) の 2 種類を用いた. 訓練性能, 汎化性能は 10-fold 交差検証 (学習用 RFP 内の要求仕様を 10 個に分割し, そのうちの 9 つを訓練データ, 1 つを検証データとして評価する, という作業を検証データを変更しながら 10 回繰り返す検証手法) によって評価する. 10-fold 交差検証による 10 回の計測結果を平均した値を訓練性能, 汎化性能として算出する. 一般的に RFP は章構成を持ち, 特定の要求仕様が偏って発生する傾向がある (例えばセキュリティに関する非機能要求がセキュリティという節にまとめて記載される, 等). したがってデータ分布の偏りを抑制するため, 交差検証を行う際は RFP 内の要求仕様をランダムに並べ替えてから要求仕様の分割を行うこととした.

図 3(3) では, 未知の RFP に対する性能を評価するため, 10-fold 交差検証で生成した分類モデルそれぞれについてテスト用 RFP に含まれる要求仕様を用いて性能を測定し, 平均値を算出する. 本稿ではテスト用 RFP に対する性能をテスト性能と呼ぶこととする.

4.2 実験内容

1 章で掲げた研究設問に基づいて, 4.1 節に示した学習用 RFP/テスト RFP を適切に選択し, 実験を行う. 本稿では表 7 に示す 3 種類の実験を行う. 表 7 に示すドメインは表 1 に示す RFP のドメインを示している.

表 1 では「中央省庁」ドメインが最も文章数が多く, 妥当性の高い評価ができると考えたため, 実験 1, 実験 2 の「特定ドメインの RFP」には中央省庁の RFP を選択した. 実験 1 では, 学習用 RFP とテスト用 RFP を分割するため, 中央省庁の RFP の 12 ファイルを 3 ファイルずつ 4 グループに分け, 1 グループをテスト用 RFP, 残りの 3 グループを学習用 RFP として利用する, という操作を 4 グループ分繰り返す, 平均値を評価した. 実験 2 では, 実験 1 との比較のため, 実験 1 の交差検証の各イテレーションで生成した分類モデルを実験 2 の評価対象の RFP にも適用し, 平均値を評価した. 評価対象の「学習用 RFP と別のドメインの RFP」には, 比較的文章量の多い (2000 以上の文章を持つ) ドメインの中で「中央省庁」を除いた 4 ドメイン (「地方支分部局」, 「市町村」, 「都道府県」, 「公法人」) を選択した. 実験 3 では, 実験 1 との比較のため, 実験 1 でテスト用 RFP として利用した RFP のグループをテスト用 RFP として, テスト用 RFP を除いた RFP 全てを

表 7 実験内容

実験	学習用 RFP	テスト用 RFP
実験 1	特定ドメインの RFP	学習用 RFP と同じドメインの学習に利用しなかった RFP
実験 2	特定ドメインの RFP	学習用 RFP と別のドメインの RFP
実験 3	全 RFP (テスト用 RFP に含まれるものを除く)	学習に利用しなかった RFP

表 8 実験結果

実験	学習用 RFP	テスト用 RFP	汎化性能		テスト性能	
			精度	誤差	精度	誤差
実験 1	中央省庁	中央省庁	72%	0.85	57%	1.56
実験 2	中央省庁	地方支分部局	72%	0.85	46%	1.86
		市町村			47%	1.74
		都道府県			56%	1.50
		公法人			45%	2.00
実験 3	全 RFP	中央省庁	68%	0.98	62%	1.30

学習用 RFP として利用し, 評価を行った (実験 1 の 4 つのグループを 1 つずつ選択し評価).

4.3 実験結果

4.1 節に示す手順に従い, 表 7 に示す 3 種類の実験を行った. 実験結果を表 8 に示す.

汎化性能は学習時に算出する検証データに対する性能, テスト性能はテストデータに対する性能である. 実験 1 と実験 2 は訓練データが同じであるため, 汎化性能の値は同じとなる. 実験 1 と実験 2 の結果を比較すると, 学習用 RFP と同じドメインの RFP をテスト用 RFP として利用した実験 1 のほうがテスト性能が高いという結果が出ており, 1 節で仮説を立てた通りの結果となった. また, 実験 1 と実験 3 の結果を比較すると学習用 RFP として全 RFP を利用した実験 3 のほうがテスト性能が高いという結果が出た.

実験環境には Amazon Web Service(AWS) の仮想サーバホスティングサービスである Amazon EC2(Amazon Elastic Compute Cloud) の GPU インスタンス p2.xlarge を 1 台利用した. 実験環境のスペックは vCPU: Intel Xeon CPU E5-2686 v4@2.30GHz×4, RAM: 61GB, GPU: NVIDIA Tesla K80×1, GPU メモリ: 12GB である. 実験環境は Docker イメージとして構築し, 利用した. Docker イメージは NVIDIA 社が提供している CUDA のイメージ (nvidia/cuda:9.0-cudnn7-runtime) をベースに構築した.

実験 1 と実験 2 は訓練データが同じであるため, まとめて実験を行った. 学習及び推論にかかった時間の合計は実験 1/2 が 2 時間 19 分, 実験 3 が 5 時間 48 分であった.

5. 考察

5.1 研究設問に対する考察

1 章に挙げた研究設問に対して考察を与える.

表 9 実験 3 における要求仕様の種別毎の評価

	機能要求	非機能要求	プロジェクト管理	対象外
適合率	0.49	0.63	0.71	0.73
再現率	0.43	0.52	0.83	0.70

RQ1: 分類対象を RFP に含まれるすべての文章に拡大した場合、RFP の調達元のドメインの違いが分類精度にどの程度影響を与えるか。

実験 1, 実験 2 のテスト性能を比較すると、学習用 RFP と同じドメインの RFP を用いて推論を行った実験 1 において最も高い精度 57% が出ており、学習時に利用した RFP と別のドメインの RFP を予測する場合 (実験 2) は精度が下がる結果となった。ただし、文献 [7] での結果に比べると、精度の低下の幅は小さく、ドメインの違いによる影響はそれほど大きくないと判断できる。これは、本稿で分類対象に加えた機能要求のデータの分布が、調達元のドメインだけでなく、調達対象システムの特性にも依存しているためであると考えられる。

また実験 3 ではテストデータに対する精度が 62% であり、実験 1, 実験 2 よりも高い精度を得ることができた。これは、文献 [7] で示した結果と同様に、訓練データの RFP のドメインを固定せず、ドメイン横断で RFP を収集し学習した場合の方が精度が高いことを示している。これらのことから、RFP の調達元のドメインの違いによる分類精度への影響は小さいことから、調達元のドメインを固定して少ないデータで学習させるよりも、様々な調達元のドメインから訓練データを収集して学習させたほうが、高い精度を得られる可能性がある。

RQ2: 分類対象を RFP に含まれるすべての文章に拡大した場合、文献 [7] に比べ分類精度に影響があるか。

実験 3 の結果より、全 RFP を学習データとした場合、テストデータに対する精度は 62%、誤差は 1.30 となった。文献 [7] では、全 RFP を学習データとした場合のテストデータに対する精度は 69%、誤差は 0.94 であり、本稿での実験では精度が 7% 低下し、誤差が 0.36 大きくなった。

精度が悪化した原因について考察するため、実験 3 のテストデータを利用した推論結果について、要求仕様の種別毎に適合率、再現率を算出した。表 9 に算出した適合率、再現率を示す。表 9 の「機能要求」、「非機能要求」は、それぞれの種別毎に算出した値の平均値を記載している。

表 9 では、機能要求の適合率、再現率の値が他と比べて低く、これが全体の精度を低下させている。機能要求の分類精度が低くなる原因としては、RQ1 の考察にも関連しているが、機能要求は調達対象システムの特性によって記述内容や記述レベル、使用される用語の差異が大きいことが挙げられる。また、「データ」に関する仕様は、データモデルの属性名など 1 つ 1 つの文章が短いものが多く含まれる

ため、適切に特徴を学習することが難しい。機能要求以外の種別については比較的高い精度で分類ができており、機能要求の分類精度向上が今度の課題となる。

5.2 分類モデルによる要求仕様の分類例

実験で作成した分類モデルを用いて具体的な要求仕様のテキストを分類し、考察を行った。機能要求のテキストは機能要件の合意形成ガイド [9] の「システム化業務一覧」、「画面一覧」、「エンティティ一覧」、「外部インタフェース一覧」、「バッチ処理一覧」、「帳票概要」の表記例から取得した。表記例には、「機能」、「画面・帳票」、「データ」、「外部インタフェース」の 4 種別に関する要求仕様の例文が示されていた。非機能要求のテキストは非機能要求グレード [10] に含まれる「02_利用ガイド利用編.pdf」の「表 3.1.2 例示受注システムの非機能要求の概要」から取得した。「表 3.1.2 例示受注システムの非機能要求の概要」には、非機能要求の 6 種別に関する要求仕様の例文が示されていた。これら要求仕様の例文を一文単位に分解し、分類モデルによって分類した。分類結果の抜粋を表 10 に示す。

表 10 の「正解」は、要求仕様の例文の取得元で付与されていた種別 (分類結果の正解として利用)、「要求仕様テキスト」は、要求仕様の例文を一文単位に分解したもの、「分類結果」は、分類モデルが予測した上位 3 個の種別とその確率を示している。

本例では、機能要求は全 50 個の例文に対し 24 個の要求仕様を正しく分類 (表 10 の「正解」に記載の種別を 1 位として予測) することができた。この内訳は、「機能」が 17 個中 14 個、「画面・帳票」が 19 個中 10 個、「データ」が 6 個中 0 個、「外部インタフェース」が 8 個中 0 個であった。「機能」の分類の精度が高く、「データ」、「外部インタフェース」で分類の精度が低いことから、機能要求の種別間で、分類の精度に差があることが分かる。「画面・帳票」については「機能」と分類したものが 5 個と多いが、これは、教師データの作成時に「画面・帳票」に関する要求仕様を「機能」とラベル付けしてしまったケースがあるためであると考えられる。「データ」や「外部インタフェース」については、RQ2 の考察にあるように 1 つ 1 つの文章が短いものが多く適切に学習できていなかった可能性が高い。

非機能要求は、全 12 個の例文に対して 8 個の要求仕様を正しく分類することができ、全ての例文で分類結果の上位 3 個に正解が含まれていた。

5.3 分類モデルの精度に対する考察

実験 3 の結果では、テストデータに対する分類精度は 62% であったが、これが実際の要求定義の作業においてどの程度実用的であるか考察する。

分類モデルの実用性を評価するため、評価実験で作成した分類モデルに対して、著者らが所属するシステムベンダ

企業に在籍する要求定義の実務者 2 名にインタビューを実施した。実務者へのインタビューでは、重点的にレビューが必要な重要項目について、要求仕様書から要求仕様をきちんと洗い出しているか人手で確認するのは多大な作業時間を要するため、要求仕様の洗い出しを補完するような目的で分類モデルを利用することで作業時間を削減できるのではないかと、またそのような目的であれば現在の精度でも十分有用なのではないかと、というコメントを得た。このようなコメントから、分類モデルのユースケースによっては現在の精度でも十分実用的であると判断することができる。

ただし、本稿では、採用した深層学習のアルゴリズムやネットワーク構成、ハイパーパラメータの設定値については簡易的な予備実験の結果を元に決定しており、網羅的に評価実験を行った上で決定を行った訳ではない。よって、網羅的に評価実験を行い適切な選択をすることで、より精度の高い分類モデルを生成することができる可能性がある。

5.4 今後の課題

本実験では、表 6 に示すように、要求仕様の種別のデータ数が不均衡であった。機械学習では、アンダーサンプリング等の手法によりデータ数を均衡させてから学習を行うのが一般的であるが、本実験では収集したデータの規模が小さく、またデータの偏りが大きかったため、データ数を均衡させることができなかった。今後、要求仕様の種別のデータ数を均衡させ、評価実験を行う必要がある。

本稿では、調達対象システムのドメインの違いに着目した実験を行ったが、RQ1, RQ2 の考察で述べたように、機能要求のデータ分布は調達対象システムの特性に依存する可能性があるため、調達対象システムの特性に着目した実験を行うことも今後の課題となる。

本稿で分類した機能要求や非機能要求のように、要求仕様の種別は、階層構造をもつデータと捉えることができる。本稿では、階層構造を無視してフラットなデータとして扱ったが、その他にも階層毎に学習、分類を行う手法など、いくつかのアプローチが提案されている。今後、要求仕様の階層構造をどのように扱うかについては検討していく必要がある。

6. まとめ

本稿では、深層学習によって要求仕様を自動で分類する手法を提案した。評価実験の結果、62%の精度で要求仕様の自動分類ができるという結果を得た。分類モデルの精度は、訓練データのドメインを特定するよりも、訓練データの量を増やす方が分類の精度の向上に寄与することが分かった。不均衡な訓練データの扱いや、要求仕様の階層構造の扱いが今後の課題である。

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] 山田 さつき, 大森 隆行, 大西 淳 : 要求文書からの信頼性要求の抽出と検証, 情報処理学会研究報告, Vol.2018-SE-199, No.28(2018).
- [3] 大東 誠弥, 福井 克法, 宮崎 智己, 大平 雅雄 : 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, ソフトウェア・シンポジウム 2019 in 熊本, pp.36-44 (2019).
- [4] Dekhtyar, A. and Fong V. : RE Data Challenge: Requirements Identification with Word2Vec and TensorFlow, Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference(RE), pp.484-489(2017).
- [5] Kurtanovic, Z. and Maalej, W.: Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning, Proceedings of the 2017 IEEE 25th International Requirements Engineering Conference(RE), pp.490-495(2017).
- [6] 松田 隼士, 奥野 拓 : 深層学習を用いたゴール指向要求分析における環境要因による影響範囲の特定手法, 情報処理学会研究報告, Vol.2017-SE-19S, No.12(2017).
- [7] 北川 貴之, 長岡 武志 : 深層学習による日本語で記述された要求仕様の自動分類の試み, 情報処理学会研究報告, Vol.2019-SE-201, No.17(2019).
- [8] 丸山宏: 機械学習工学に向けて, 日本ソフトウェア科学会第 34 回大会 (2017 年度) 講演論文集, (2017).
- [9] IPA/SEC : 機能要件の合意形成ガイド, IPA/SEC(2010).
- [10] IPA/SEC : 非機能要求グレード 2018, IPA/SEC(2018).
- [11] Kudo, T., Yamamoto, K., Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis, Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.230-237 (2004).
- [12] Mikolov, T., Chen, K., Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space, Proceedings of the International Conference on Learning Representations 2013(ICLR) Workshop Track, arXiv:1301.3781, (2013).
- [13] gensim, <https://radimrehurek.com/gensim/>
- [14] Kim, Y. : Convolutional Neural Networks for Sentence Classification, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1746-1751, arXiv:1408.5882 (2014).
- [15] Zhang, Y. and Wallace, B.: A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification, Proceedings of the 8th International Joint Conference on Natural Language Processing (IJCNLP), pp.253-263(2017), arXiv:1510.03820.
- [16] Keras: Python の深層学習ライブラリ, <https://keras.io/ja/>
- [17] Abadi, M., Agarwal, A., Barham, P., et.al: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, arXiv:1603.04467(2016).

表 10 要求仕様の分類例

No	正解	要求仕様テキスト	分類結果 (上位 3 個)
1	機能	QA システムを利用するためにユーザ情報の登録を行う。	1. 機能:0.60 2. 業務:0.07 3. 運用・保守性:0.06
2	画面・帳票	ペットストアのポータル。購入、検索、カ タログ選択などのメニューを表示する。	1. 画面・帳票:0.89 2. 機能:0.11 3. データ:0.00
3	データ	顧客の個人情報	1. プロジェクト管理:0.63 2. 対象外:0.18 3. セキュリティ:0.07
4	外部インタフェース	受注情報 ○ 販売管理システム 新規 ftp XML バッチ (定時) 日中	1. プロジェクト管理:0.34 2. 対象外:0.18 3. 環境:0.13
5	可用性	システムは、24 時間電源を投入する。	1. 運用・保守性:0.35 2. 可用性:0.26 3. 環境:0.12
6	性能・拡張性	今後 5 年間で取引先が現在の 2 倍に増加し ても対応可能な拡張性を確保しておきたい。	1. 性能・拡張性:0.63 2. 可用性:0.09 3. 対象外:0.07
7	運用・保守性	システムを監視し、システムが停止した場 合には、運用部門に即時に報告される仕組 みにしたい。	1. 運用・保守性:0.78 2. プロジェクト管理:0.17 3. 移行性:0.02
8	移行性	受注に使用する取引先マスタ、商品マスタ を移行する。	1. 移行性:0.83 2. プロジェクト管理:0.12 3. 対象外:0.02
9	セキュリティ	特定の取引先との接続のみを許すが、さら に、取引先マスタ、商品マスタ等のデータ は保護したい。	1. 運用・保守性:0.21 2. セキュリティ:0.19 3. プロジェクト管理:0.12
10	システム環境・エコロジー	耐震は通常レベルの対策で良い。	1. 対象外:0.53 2. プロジェクト管理:0.21 3. システム環境・エコロジー:0.11