

RFPにおける機械学習による非機能要件の評価

齊藤康廣† 門田暁人† 松本健一†

本稿では、自然言語で記述された提案依頼書 (Request For Proposal ; 以下 RFP とする) に記載された非機能要件の記述の明確さを機械学習により評価する方法を提案する。提案方法では、まず、RFP から非機能要件に関するキーワード群を抽出し、個々の非機能要件の特性とマッピングさせる。次に、各キーワードの出現頻度と文脈ベクトルに着目して重み付けを行い、ランダムフォレストによって、非機能要件の記述の明確さをモデル化する。70 件の RFP を題材として、提案方法によって多数の非機能要件の記述の明確さを 3 段階で評価した結果、エキスパートによる評価に対する完全一致率の平均が 69.8% となった。また、完全不一致率 (評価が 2 段階外れること) は極めて小さかった。このことから、エキスパートがいない状況においても、機械学習によって RFP の品質の自動評価をある程度行えることが分かった。

Evaluation of RFPs Based on Machine Learning

Yasuhiro Saito † Akito Monden †
and Kenichi Matsumoto †

This paper proposes a machine learning approach to evaluate the clarity of non-functional requirements (NFRs) described in a Request For Proposal (RFP) written in a natural language. In the proposed method, keywords related to NFRs are extracted from a RFP, and mapped to each NFR category. Then, the clarity of NFRs is modeled by the random forest with weight factors based on appearance frequency and context vectors. As a result of an experimental to evaluate the clarity (low, mid or high) of many NFR categories in 70 RFPs, the proposed method showed 69.8% match to the expert's decision. Also, there were few cases where the model concluded as clarity=high while expert concluded clarity=low, and vice versa. These results suggest that the proposed machine learning approach could be used to automatically evaluate the quality of RFP without experts.

1. はじめに

委託ソフトウェア開発プロジェクトの超上流工程にて、ユーザ (発注企業) が作成する RFP のユーザ要件に関する自然言語で記述された非機能要件の内容が十分であるかどうか (記述の明確さ) を評価することは、後工程においてユーザ要件を満足するソフトウェアシステムを構築する上でユーザとベンダ (開発企業) の両者にとって極めて重要である。要求工学の観点からもソフトウェア開発の上流工程で自然言語のテキストから要求を抽出することがソフトウェアプロジェクトによって重要であるとされている[10]。また、評価を定量化することによる客観的評価は、ユーザとベンダ間での合意が得やすいという点から要求の定量化評価を支援するモデルが求められる。著者らの先行研究[4][17]では、ユーザ視点からの非機能要件を評価する評価指標で構成される非機能要件評価表を作成し、ケーススタディにおいてエキスパート技術者による評価を行った。しかし、非機能要件評価表に基づいた人手による評価は、エキスパートがいることが前提となり、また、評価にも時間を要する点が問題である。そこで、これらの問題を解決するために、本稿では、多数の RFP から抽出した非機能要件キーワードによる評価モデルを構築し、評価の自動化を試みる。

非機能要件の記述内容が十分であるかどうかを評価するためには、自然言語で記述された RFP から非機能要件に関連するキーワードを抽出しなければならない。自然言語で記述された文書の内容から知識としてのキーワードを抽出する手法としてテキストマイニングが注目されている。さらに、非機能要件特性と抽出されたキーワードをマッピングし、複数のキーワードで構成される非機能要件特性を自動的に評価する評価値推定モデルが必要である。モデリングにおいては、評価精度が高くない複数の結果を統合・組み合わせることで精度を向上させる機械学習方法として集団学習がある。そのなかでも、高次元データ解析に向いており、精度が高いとされている手法としてランダムフォレスト法がある。そこで、本稿では、非機能要件を定量評価するモデルとして、テキストマイニングを用いて抽出したキーワードに基づいて、ランダムフォレストにより非機能要件特性の評価モデルを構築し、RFP に記述された非機能要件が十分であるかどうかの判定結果を評価する一連の手続きにより半自動的評価することを目的とした。提案する非機能要件の評価モデルにより、委託ソフトウェア開発プロジェクトにおけるユーザが要求する非機能要件を客観的かつ定量的な評価を支援することが可能であることを示す。

2. 関連研究と調査資料

(1) 先行研究

RFP の非機能要件を評価する研究として、著者らの先行研究[4][17]がある。この研究では、開発の初期段階に明ら

†1 奈良先端科学技術大学院大学 情報科学研究科
Graduate School of Information Science, Nara Institute of Science and
Technology

かにしておくべき要求が、RFPに十分に記述されているかどうかを評価することを目的として、RFPの段階において記述すべきメトリクス及び記述可能なメトリクスをガイドライン等から抽出し、RFP非機能要件評価表を構築した。評価手法として5段階評価手法を採用し、RFPに記述されている非機能要件メトリクスを定量的に評価する評価基準を定義した。

ソフトウェア開発の上流工程にて自然言語で記述された非機能要件を評価する国内での先行研究は少ない。ソフトウェア開発の上流工程にテキストマイニングを応用する実用的な研究として、日本国内では福田[5]による概念データモデリング、SOAのサービス抽出及びサービスのクラスター化を支援する試みが提案されているが、クラスター分析及び文書・語句行列による評価の試行にとどまっており、定量評価は行われていない。また、今村ら[1]による技術文書からの用語知識を自動的に獲得する実験は、共起関係としての「係り受け先」と「文節内の後方」が体系的意味分類として有効であることを指摘しており、自然言語で記述された技術文書から語句を抽出する場合に有益な手法を示唆している。しかし、機械学習によるモデル化は今後の課題となっている。また、自然言語で記述された文書の質判定については、英語で記述された論文を対象としたさまざまな手法による質判定の研究[2][3]が行われている。小林らは、論文の質が如実に反映される言語項目を素性として論文の質をランダムフォレストにより2クラス分類し、分類精度を評価している。本研究では、非機能要件キーワードによる非機能要件特性を多クラス分類し、その一致度を統計解析し評価している点が優れていると言える。

一方、海外の先行研究では、自然言語で記述された要求仕様を、ソフトウェア開発の上流工程において評価する重要性の視点から、テキストマイニング技術により抽出した語句にもとづいて要求分析を行い、ソフトウェア設計を支援する手法及び評価に関する先行研究[7][8][9][10][11][15]がある。本研究の対象である自然言語で記述された非機能要件に着目した先行研究としては、1)サポートベクタマシンによる非機能要件、特にアーキテクチャに関する分類器に関する先行研究[12]、2)セキュリティ、パフォーマンス、操作性などの非機能要件に関する語句をキーワードとして、独自の分類器を用いて評価する手法を提案している先行研究[13][14]及び3)コンポーネントベース開発を対象として非機能要件メトリクスにもとづいた品質評価に関する先行研究[16]がある。しかし、これらの先行研究は、非機能要件をパフォーマンス、セキュリティ、可用性などの非常に大きなカテゴリーのみで評価している。しかし、本研究の目的であるRFPに記述されている非機能特性の記述が十分であるかを評価し、RFPに記述されている非機能要件の品質向上を支援するにはより具体的な非機能要件についての分析及び評価が求められる。

(2) 調査資料

テキストマイニングのフリーソフトウェアとして、KH-coder[4]を使用した。テキストマイニングツールの多くは、有料で提供されている。英語版フリーのテキストマイニングツールはWEKAやTANAGRAなどがあるが、日本語版のフリーツールであるKH-Coderは、非常に充実した評価機能及びオプション機能があり、評価が高いことから本研究のテキストマイニングツールとして採用した。

3. テキストマイニングによる非機能要件抽出

自然言語で記述されたRFPの非機能要件の評価に関連するキーワードを抽出するためのテキストマイニング手法として、フリーソフトウェアであるKH-Coderを採用した。抽出された非機能要件としてのキーワードをその出現頻度でフィルタリングしたのちに、非機能要件評価表で分類した非機能要件特性にマッピングし、それぞれの特性を構成するキーワードによる判定をランダムフォレストにより推定した。以下にそのアプローチを述べる。

非機能要件特性評価判定モデル構築までのアプローチ

- 非機能要件評価表（既に作成済み）
- 70件のプロジェクトのRFPから非機能要件文書を抽出し、テキストマイニングにより形態素解析
- 形態素解析された結果から、非機能要件該当する名詞と複合語を出現頻度により手作業で抽出
- 絞り込まれた頻出名詞と複合語を非機能要件特性にマッピングし、非機能要件評価キーワード表を作成
- キーワードをタグとして、もう一度形態素解析を行い、非機能要件ごとのプロジェクトとキーワードの出現頻度で構成されるマトリクスを作成
- 形態素解析による文脈ベクトルを各キーワードへの重み付として、重み付きマトリクスを作成
- 重み付きマトリクスのキーワードを説明変数とし、評価を目的変数としたデータを基に、ランダムフォレスト法により評価判定

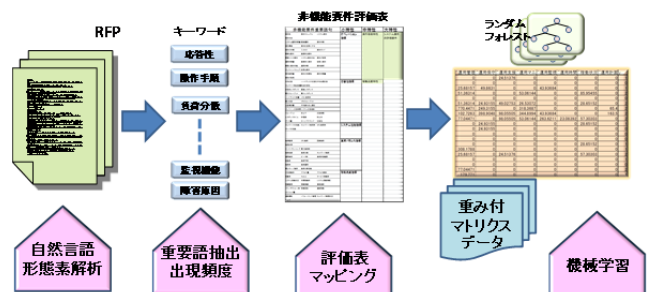


図1 非機能要件特性評価判定モデル構築までのアプローチ概要

Fig 1 Outline of Approach for NFR Evaluation Model Creation.

3.1 RFP 評価サンプルデータ

70 件の自然言語で記述されている RFP をサンプルデータとして評価を行った。評価対象とした RFP サンプルデータは、ウェブ上に公開されている図書情報システム (11 件)、病院情報システム (10 件)、大学情報システム (8 件)、政府機関情報システム (14 件)、自治体基幹情報システム (10 件)、地方自治体業務システム (14 件) 及びその他情報システム (3 件) の 70 件である。これらのプロジェクトの非機能要件に関する記述部分にもとづいて評価データとして生成したデータのサイズは、37,281 行の 1,357,655 語のテキストファイルである。非機能要件として RFP に記述された内容は、自然言語の文章として記述されている部分、表として記述されている部分及び図として記述されている部分で構成されている。評価データとして生成したテキストデータは、PDF 形式を TEXT 形式に変換して生成したが、図として記述されている部分は削除した。また、表で記述されている部分は、可能な限り TEXT 変換された内容から非機能要件記述部分をフィルタリングすることで評価データとして採用することとした。

3.2 非機能要件を構成する重要語の抽出のステップ

RFP に記述されている非機能要件を抽出する手法として、テキストマイニングにより語句を抽出してその出現頻度から非機能要件としての評価指標を決定した。テキストマイニングとしては、フリーソフトである KH-Coder を採用した。Web から検索した 70 件のプロジェクトに関する自然言語で記述された RFP から、非機能要件に関する記述をプロジェクトごとに章としてまとめたテキストデータを KH-Coder による形態素解析用のデータとした。評価を行うに当たって、過去の研究結果で作成した「非機能要件評価シート」の中特性のうち、重要度が高い非機能要件である「操作容易特性」、「稼働品質特性」、「障害検知特性」、「システム監視特性」、「セキュリティ対策特性」「冗長化特性」、「データバックアップ特性」、「障害予防特性」および「障害復旧特性」に関する 10 個を対象とした。

ステップ 1

各プロジェクトの非機能要件が記述されたテキストデータに対して、テキストマイニングツールである KH-Coder の TermExtract を使用して複合語抽出し、さらに形態素解析機能により名詞語句を抽出した。形態素解析の結果得られたこれら語句から非機能要件に関連する出現頻度の高い (2 回以上出現) 非機能要件関連する語句をフィルタリングした。フィルタリングにおいて、出現頻度は高いが一般的な語句 (オペレーション、性能、など) は除外し、RFP に記述されている非機能要件を特徴づける語句 (操作を容易にする、平均読み出し遅延、障害切り分けなど) を追加した。

ステップ 2

フィルタリングした語句を非機能要件の小特性評価のため

のキーワードとして非機能要件特性にマッピングし、非機能要件キーワードと大特性、中特性および小特性により構成される「非機能要件キーワード表」を作成した。

ステップ 3

再度、非機能要件キーワードをテキストマイニングツールである KH-Coder のタグとして指定し、小特性ごとのキーワード出現頻度を列とし、70 件のプロジェクトを行とするマトリクスを生成した。

ステップ 4

小特性に属する非機能要件キーワードに関する文脈ベクトルを求め、これをキーワード出現頻度の重みとして使用した。文脈ベクトルは、語句の周辺に現れる単語であり、その語句を特徴づけるベクトルとされている。これにより、各キーワードの類似表現を考慮することになり、出現頻度だけに依存しない方法を採用した。

3.3 非機能要件評価データの生成

「非機能要件評価表」と非機能要求キーワードによる「重み付き評価マトリクス」生成プロセスの概要を図 2 に示す。非機能要件キーワード表にマッピングされた各非機能要件キーワードについて、KH-Coder の「抽出語 x 文脈ベクトル表」作成機能を使用して文脈ベクトル表を生成した。生成した文脈ベクトル表より、各非機能要件キーワードの文脈ベクトルを加算した数値をそれぞれの重みとして採用し、プロジェクトを行とし、説明変数である非機能要件キーワードを説明変数とする出現頻度に乗ずることにより重み付き評価マトリクスを生成した。

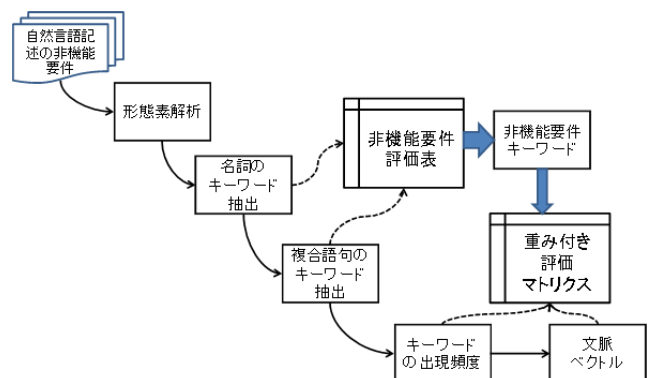


図 2 非機能要件評価データセット生成プロセス
 Fig 2 Creating Process of NFR Evaluation Data Sets.

3.4 非機能要件キーワード表の構成

「非機能要件評価表」の最下層は非機能要件キーワードであり、この非機能要件キーワードが小特性にマッピングされ、小特性は中特性に、中特性は大特性にマッピングされた 4 層構造をもつ非機能要件を評価する指標を特徴づけるものである。「非機能要件キーワード表」の一部を表 1 に示す。

表 1 非機能要件キーワード表

Table 1 Table of Non-Functional Requirements Key Words.

大特性	中特性	小特性	非機能要件キーワード
システム運用の評価要件	操作容易性	オペレーション指標	操作性, 操作マニュアル, システム操作, 操作方法, システム運用手順書, 画面遷移, 操作環境, 操作を容易にする, 操作手順, 操作の手順, 入力ミス, 操作ガイド, 簡単な操作, 直感的な操作, 操作が容易, 簡単に入力操作, システム操作方法, 容易に操作, 運用手順書, 直感的に操作,
	稼働品質特性	応答性指標	平均応答, ハードディスク応答性能, 平均処理応答, ネットワーク転送容量, 転送応答性, 最小レスポンス, 安定的レスポンス, 画面レスポンス, ターンアラウンド, 最大スループット, VPN スループット, 応答性, ハードディスク容量, メモリ使用率, 応答時間, 最小応答, 平均読み出し遅延, 秒以内
		システム性能指標	ネットワークの性能, ネットワーク使用率, MPU 使用率, サーバの性能, ディスク IO 負荷率, 性能監視機能, CPU 使用率, % 以下, SPEC, 端末性能, ネットワーク性能, 演算性能, サーバ性能, ハードウェア性能, ソフトウェア性能,
		負荷バランス指標	負荷監視, CPU 負荷, 回線負荷, 負荷計測, 最大負荷時, 負荷分散性能, ロードバランシング, 負荷分散, 負荷率, 負荷低減, ピーク時, ネットワーク負荷, 運用負荷, 負荷予測, 負荷許容範囲
		稼働品質指標	平均稼働率, アクセス量, アクセス頻度, 稼働率, % 以上, サービス稼働率, システム稼働率, 正常稼働, 安定システム, 稼働状況, 安定稼働実績, 稼働実績, 稼働期間, 正常に稼働, 24 時間 365 日稼働, 安定稼働, システム稼働管理, ダウンタイム,
運用監視要件	障害検知特性	システム異常検知指標	事故発生, 異常発生通知, 停電信号, イベント発生, 上限値警告, 状態監視, 障害通知機能, 状態監視機能, 異常, 障害監視機能, 自動運転監視, システム監視, 接続状況監視, 自己診断機能, 検知システム, 障害検知, 温度異常, モニタリング
		障害要因指標	運用管理ソフト, 運転状況, 運用マニュアル, 運用管理, 運用保守, 運用監視, 運用監視ソフト, 運用時間, 運用状況, 稼働状況, 稼働状況, 運用状態

4. ランダムフォレストを用いた非機能要件評価実験

本研究では、精度が高く、多くの説明変数を扱うことができるランダムフォレストによる分類器を生成する手法を採用した。データセットとして、3 章で述べたテキストマイニングによりプロジェクトごとの非機能要件キーワード出現頻度を説明変数とし、プロジェクトごとの段階評価結果を目的変数として生成したマトリクスを用いた。

評価対象とした 10 個の「中特性」を構成する各「小特性」にマッピングされた「非機能要件キーワード」を説明変数とし、教師データとしての評価を目的変数として、ランダムフォレストの分類問題による評価を行った。サンプルデータの 2/3 を学習データとしてモデルを作成し、残り 1/3 をテストデータとして判定のための分類を行った。目的変数としての評価データは、各小特性に対する評価として 5 段階評価, 3 段階評価および 2 段階評価の 3 種類を与えて、ランダムフォレストモデルによる評価判定結果の相違点について検証を行った。

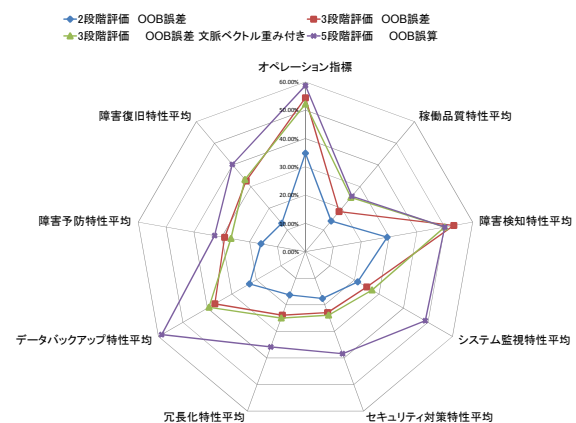


図 3 評価法別 ランダムフォレストモデル OOB 誤差
 Fig 3 Random Forest Model OOB error of Each Evaluation Way.

4.1 評価手法の分析

ランダムフォレストの学習用データ及びテスト用データの目的変数である評価は、5 段階, 3 段階及び 2 段階とし、これらを比較評価した。5 段階評価, 3 段階評価及び 2 段階評価のランダムフォレストの OOB 誤差は図 3 となった。この結果から、5 段階評価が最も誤差が大きく、2 段階評価の誤差が最も小さい結果となった。また、3 段階評価では大きな差異はないが、文脈ベクトル重み付きの 3 段階評価が比較的 OOB 誤差が少ないと言える。そこで、ランダムフォレストによる予測結果を評価するために、非機能要件キーワード表の各小特性に対する評価結果を中特性ごとに平均値を求め、ランダムフォレスト予測結果の一致率を評

価する手法として κ 統計値による検定を行った。この結果を図 4 に示す。2 段階評価は非常に高い一致率（データバックアップ特性、冗長化特性）と非常に低い一致率（障害予防特性、障害復旧特性）の差が大きいことが分かった。また、5 段階評価は、すべての非機能特性において一致率が最も低い結果となった。3 段階評価の κ 統計値については、文脈ベクトル重み付きデータセットのランダムフォレスト予測結果を重み付け κ 統計値で評価した結果が最も一致率が高い結果となった。これらの、評価方法による結果の相違から、RFP を非機能要件キーワードに基づいて評価する手法として、非機能要件キーワードの出現回数にその非機能要件キーワードの文脈ベクトルを重みとして考慮したデータを重み付き κ 統計値で評価した結果が最も有効であることが分かった。

そこで、3 段階評価文脈ベクトル重み付きを分析対象として採用し、非機能要件小特性に関する一致率分析を行った。その結果を図 5 に示す。重み付け κ 統計値による各非機能要件小特性の一致率分析結果を図 4 に示す。 κ 統計値は偶然によらない一致率が計算され、0 から 0.4 は低い一致率として判定する。この判定基準によると、26 個の非機能要件小特性のうち 11 個が低い一致率として判定され、残りの 15 個が中程度の一致、かなりの一致あるいは高い一致と判定される結果となった。

次に、見かけ上の一致率を計算するために、ランダムフォレストによる 1/3 テストデータ予測結果に対して、見かけ上の完全一致率及び見かけ上の一致率を下記の 2 つの式で計算した。その結果を表 2 に示す。すべての小特性における見かけ上の完全一致率及び見かけ上の一致率の平均値はそれぞれ、69.8%及び 97.2%であった。

見かけ上の完全一致率 = 完全一致ケース / 全テストデータケース

見かけ上の一致率 = ((完全一致ケース) + (±1 差一致ケース)) / 全テストデータケース

ここで、完全一致ケース：データセットをランダムサンプリングして 1/3 をテストデータとした場合にランダムフォレストによる予測と教師データが一致したケース数

±1 差一致ケース：データセットをランダムサンプリングして 1/3 をテストデータとした場合にランダムフォレストによる予測と教師データが±1 差であったケース数

4.2 マニュアル評価と自動評価の比較

評価手法の中で最も一致率が高い「3 段階評価文脈ベクトル重み付き」の小特性のランダムフォレストモデルと人手による評価得点平均を比較した結果を図 6 に示す。ランダムフォレストのテストデータ評価得点を人手による評価得点と比較した結果は、すべての中特性においてランダムフォレストモデルによる予測結果が人手より低い評価結果となった。小特性では、「応答性指標」、「運用管理指標」、「不正アクセス指標」及び「バックアップ管

理指標」がランダムフォレストモデル予測評価と人手による評価がほぼ一致しているが、その他の指標ではいずれも人手による評価が高い結果となった。評価対象となったプロジェクトの各小特性の平均評価得点がすべて 2 点以下の低い評価結果であった。また、個別のプロジェクトの評価得点で見ても、3 点と評価されたプロジェクトは、ランダムフォレストモデルによる予測及び人手による予測のそれぞれにおいて、全体の 2.8%と 9.8%の不均衡データであった。プロジェクトサンプルデータ数が 70 件と少なく、非機能要件の記述が十分でなく、評価点が低いことがランダムフォレストモデルによる予測評価が低い結果となっていることが考えられる。今後は、オーバーサンプリングあるいはアンダーサンプリングの手法を用いてサンプル数の調整を行ってランダムフォレストモデルの予測を行う、あるいは、非機能要件の記述の十分性が高いプロジェクトをサンプルデータとして収集し、モデルを構築することが課題である。

表 2 ランダムフォレスト評価予測結果

Table 2 Result of Random Forest Evaluation Prediction.

NFR 小特性	完全一致	±1 差一致	完全不一致	見かけ上の完全一致率	見かけ上の一致率
オペレーション指標	12	11	1	50.00%	95.83%
稼働品質指標	20	2	2	83.33%	91.67%
応答性指標	21	3	0	87.50%	100.00%
負荷バランス指標	20	3	1	83.33%	95.83%
システム性能指標	14	10	0	58.33%	100.00%
システム管理指標	18	6	0	75.00%	100.00%
運用管理指標	19	5	0	79.17%	100.00%
障害要因指標	16	7	1	66.67%	95.83%
アクセス権限指標	13	11	0	54.17%	100.00%
ウイルス対策指標	17	7	0	70.83%	100.00%
セキュリティ管理レベル指標	15	7	2	62.50%	91.67%
セキュリティ対応指標	17	7	0	70.83%	100.00%
パスワード管理指標	18	3	3	75.00%	87.50%
暗号処理指標	22	2	0	91.67%	100.00%
情報漏洩対策指標	19	5	0	79.17%	100.00%
認証機能指標	16	8	0	66.67%	100.00%
不正アクセス指標	17	7	0	70.83%	100.00%
バックアップシステム指標	15	9	0	62.50%	100.00%
バックアップ管理指標	12	11	1	50.00%	95.83%
バッチ処理指標	18	5	1	75.00%	95.83%
リカバリ処理指標	13	10	1	54.17%	95.83%
障害管理指標	18	5	0	78.26%	100.00%
障害対策指標	15	9	0	62.50%	100.00%
停止処理指標	16	8	0	66.67%	100.00%
RAID 構成指標	18	6	0	75.00%	100.00%

冗長化指標	16	8	0	66.67%	100.00%
全体	435	175	13	69.84%	97.92%

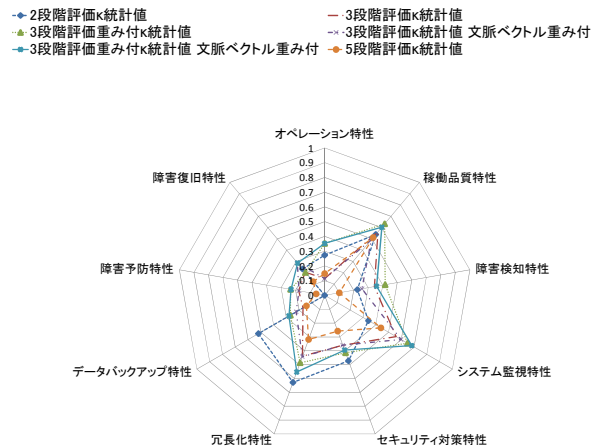


図4 評価法別 ランダムフォレストモデルの κ 統計値
Fig 4 κ Statistical Value of Random Forest of Each Evaluation

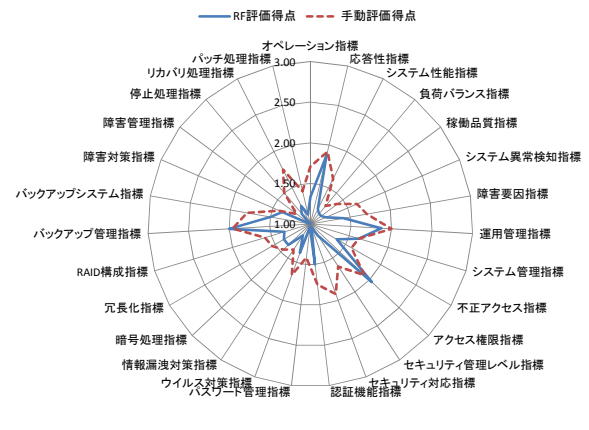


図6 小特性 RF 評価平均と手動評価平均の比較
Fig 6 Comparison between RF Evaluation and Manual Evaluation.

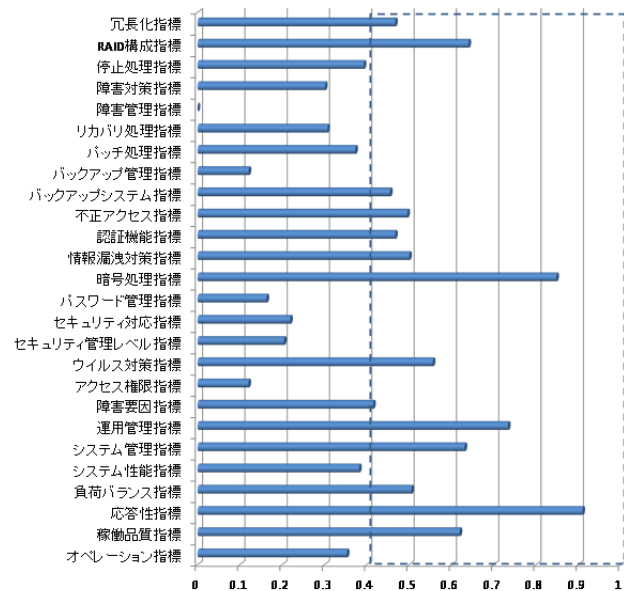


図5 3段階評価文脈ベクトル及び重み付け κ 統計値
Fig 5 κ Statistical Value with Weight for Three(3) Grade Evaluation Considering with Context Vector.

5. まとめと今後の課題

本研究では、委託ソフトウェア開発の上流工程でユーザ要件が自然言語で記述されるRFPを対象として、筆者の先行研究で構築した「非機能要件評価シート」の小特性に関連する非機能要件キーワードをマッピングすることにより、「非機能要件評価シート」の改善を行った。次に、「非機能要件評価シート」の大特性、中特性及び小特性をRFPからテキストマイニングにより抽出した非機能要件キーワードにより自動評価することを目的として、ランダムフォレストによる予測モデルを構築し、評価を行った。その結果、RFPの非機能要件記述テキストからテキストマイニング技法により、説明変数としての非機能要件キーワードの出現頻度を抽出した結果に対して文脈ベクトルで重み付けたデータをサンプルデータとした予測評価と目的変数としての3段階評価による評価の一致率が最も高い結果となった。しかし、26個の非機能要件小特性のうち11個が低い一致率として判定され、今後の一致率向上のための改善が必要である。しかし、見かけ上の完全一致率は69.8%であり、本研究の目的であるRFPに記述されている非機能要件記述が十分であるかを自動評価することに貢献することができた。今後は、本論文で評価対象外とした「非機能要件評価シート」の非機能特性についても同様に評価し分析すること、及び、非機能要件キーワードとしての名詞あるいは複合語に限らず、自然言語で記述された内容の文脈について、自然言語処理の技術を取り入れることにより一致率を向上させる研究を行う予定である。

参考文献

1) 今村誠, 高山泰博, 三上崇志, 岡田康裕: 技術文書からの用

- 語知識自動獲得の検討, 情報処理学会研究報告, 2007-FI-86, 2007-DD-60(2007).
- 2) 小林雄一郎, 田中省作, 富浦洋一: N-gram を素性とするパターン認識を用いた英語科学論文の質判定, 情報処理学会研究報告, (2012).
 - 3) 小林雄一郎, 田中省作, 富浦洋一: メタ談話標識を素性とするパターン認識を用いた英語科学論文の質判定, 人文科学とコンピュータシンポジウム論文集, pp.51-58(2011).
 - 4) 齊藤康廣, 門田彰人, 松本健一: Request For Proposal(RFP)における保守・運用要件指標の抽出と評価, 情報処理学会研究報告会(2012)
 - 5) 樋口耕一: KH_Coder2.x レファレンスマニュアル, p84(2012).
 - 6) 福田淳一: テキストマイニングのシステム開発上流工程適用の試み, Journal of the Society of Project management, Vol.13, N0.2(2011).
 - 7) Agustin Casamayor, Daniela Godoy, Marcelo Campo : Functional grouping of natural language requirements for assistance in architectural software design, Knowledge-Based Systems 30, pp.78-86 (2012).
 - 8) Agustin Casamayor, Daniela Godoy, Marcelo Campo: Mining textual requirements to assist architectural software design: a state of the art review, Artif Intell Rev38, pp.173-191(2012).
 - 9) Carlos Huertas, Reyes Juárez-Ramírez : NLARE, A Natural Language Processing Tool for Automatic Requirements Evaluation, CUBE 2012, September, pp.3-5(2012).
 - 10) Dan Port, Allen Nikora, Jane Huffman Hayes, LiGuo Huang : Text Mining Support for Software Requirements: Traceability Assurance, Proceedings of the 44th Hawaii International Conference on System Sciences, (2012).
 - 11) Leonid Kof : NATURAL LANGUAGE PROCESSING FOR REQUIREMENTS ENGINEERING: APPLICABILITY TO LARGE REQUIREMENTS DOCUMENTS : (2004).
 - 12) Gokhan Gokyer, Semih Cetin, Cevat Sener, Meltem T. Yondem : Non-Functional Requirements to Architectural Concerns: ML and NLP at Crossroads, The Third International Conference on Software Engineering Advances, pp.400-406 (2008).
 - 13) Jane Cleland-Huang, Raffaella Settini, Xuchang Zou, Peter Solc : Automated classification of non-functional requirements, Requirements Engineering12, pp.103-120(2007).
 - 14) Jane Cleland-Huang, Raffaella Settini, Xuchang Zou, Peter Solc : The Detection and Classification of Non-Functional Requirements with Application to Early Aspects : 14th IEEE International Requirements Engineering Conference (RE'06), (2006).
 - 15) Olga Ormandjieva, Ishrar Hussain, Leila Kosseim : Toward a Text Classification System for the Quality Assessment of Software Requirements Written in Natural Language : SOQUA'07, September, pp.39-45 (2007).
 - 16) Simrandeep Singh Thapar, Hardeep Singh, Karanjeet Singh Kahlon : METRICS-BASED EVALUATION OF QUALITY OF NON-FUNCTIONAL SPECIFICATIONS, International Journal of Information Technology and Knowledge Management, Volume 2, No. 1, pp. 131-134(January June 2009).
 - 17) Yasuhiro Saito, Monnden Akito, Matumoto Kennichi : Evaluation of Non Functional Requirements in a Request For Proposal(RFP) : IWSM-MENSURA(2012)