Vol.2017-SE-195 No.12
2017/3/12

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

松田 隼士[1,a)]　奥野 拓[2]

概要：要求定義では，ビジネス要求を的確に反映し，ステークホルダが合意した要求仕様をまとめることが求められる．これらを支援する手法の一つに，ゴール指向要求分析がある．しかし，市場動向などの環境が変化した場合，記述されたゴールモデルへの影響範囲を特定することが困難という問題がある．本研究では，非機能要求に関連のあるゴールへの影響範囲を特定する手法を提案する．まず，ステークホルダがゴールモデルを作成する際に，各ゴールに非機能要求をタグとして付与しておく．次に，文章として表現された環境要因を，深層学習で作成した分類器に入力し，環境要因により影響を受ける非機能要求を出力する．それらとゴールモデルに付与された非機能要求を照らし合わせることで，環境要因によるゴールモデルの影響範囲を特定する．提案手法の有効性を確認するために，提案手法に基づき作成した分類器の精度を評価する実験を行った．

キーワード：ゴール指向要求分析，深層学習，文書分類手法，非機能要求，環境要因

A Method of Specifying Ranges of Influence by Environment Factors in a Goal-Oriented Requirements Analysis Using a Deep Learning Method

Hayato Matsuda[1,a)]　taku Okuno[2]

Keywords: Goal-oriented requirements analysis, Deep learning method, Documents classification, Non-functional requirements, Environment factors

1. はじめに

システム開発の上流工程である要求定義では，ビジネス要求を的確に反映し，ステークホルダが合意した品質の高い要求仕様をまとめることが求められる．これらを支援するための手法の一つとして，ゴール指向要求分析がある [1]．この手法は，ビジネス要求に基づき，トップダウンでシステム要求をゴールとして定義するものである．これにより，ステークホルダとの合意形成を円滑に進めることができる．この手法により構築されるモデルをゴールモデルという．さらにこの手法を基に，ステークホルダ間の要求の認識齟齬や対立を明確にできるよう拡張した AGORA(Attributed Goal-Oriented Requirements Analysis Method) という手法がある [2]．

ゴールモデルが完成し，ステークホルダ間で合意が取れた後にシステム開発を進めるが，開発中に市場動向などの環境の変化が発生し，ステークホルダの要求が変わる場合がある．その場合，ゴールモデルを修正する必要があるが，修正すべきゴールを特定するには手間がかかるという問題がある．そこで本研究では，特に非機能要求に着目し，非機能要求に関連のあるゴールへの，環境要因による影響範囲の特定を支援することを目指す．そのために，文章とし

[1] 公立はこだて未来大学大学院
Graduate School of Future University Hakodate
[2] 公立はこだて未来大学
Future University Hakodate
[a)] g2115027@fun.ac.jp

© 2017 Information Processing Society of Japan

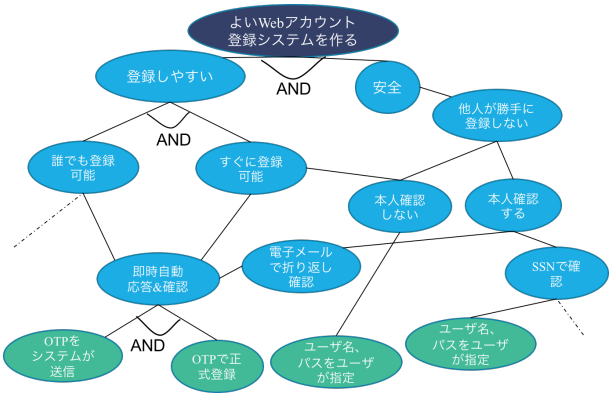


図 1　ゴールモデルの例

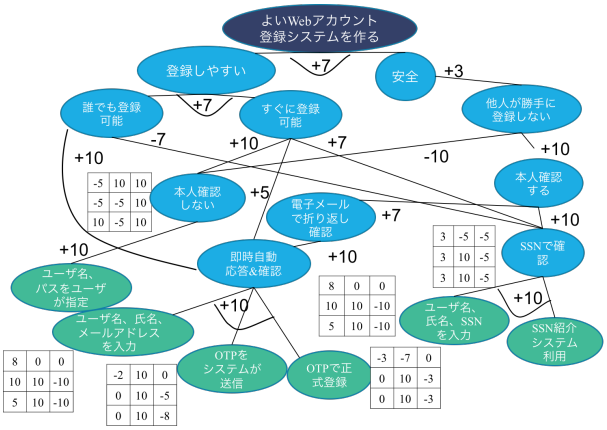


図 2　AGORA で作成されるゴールモデルの例

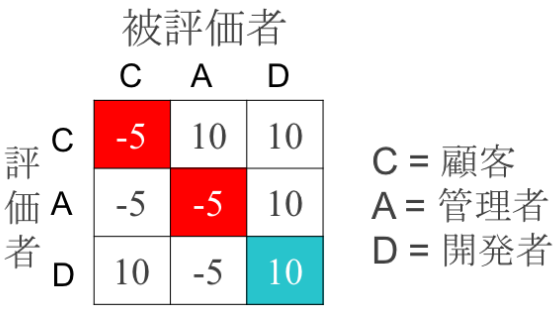


図 3　満足度行列の例

て表現された環境要因を，深層学習で作成した分類器に入力し，環境要因により影響を受ける非機能要求を出力する手法を提案する．

2. ゴール指向要求分析

2.1 ゴール指向要求分析の概要

ゴール指向要求分析とは，システム開発の目標としてのゴールを，ビジネス要求に基づき構造的にサブゴールとして詳細化し，システム要求として定義する方法論である [1]．この手法により構築されるモデルをゴールモデルという．ゴールモデルの例を図 1 に示す．

最初に，顧客やユーザのニーズを基に初期ゴールを設定する．この初期ゴールを起点として，ゴールをサブゴールへ分解していく．ここで，枝によって結ばれているゴール間で，上位に位置するゴールを親ゴール，下位に位置するゴールをサブゴールと呼ぶ．分解には AND 分解と OR 分解がある．AND 分解はサブゴールが全て達成されないと親ゴールも達成されない分解方法で，OR 分解はどれか一つのサブゴールが達成されれば親ゴールも達成される分解方法である．

図 1 の例では，初期ゴールとして「よい Web アカウント登録システムを作る」が設定されている．そのゴールは，「登録しやすい」と「安全」の二つのサブゴールに AND 分解されているため，どちらのゴールも達成されれば，「よい Web アカウント登録システムを作る」のゴールも達成されるということになる．また，「本人確認する」というゴールは，「電子メールで折り返し確認」と「SSN で確認」の二つのサブゴールに OR 分解されているため，どちらかのゴールが達成されれば，「本人確認する」のゴールも達成されるということになる．

この手法により，ビジネス要求に整合したシステム要求を洗い出すことができ，開発するシステムの妥当性を確認することができる．

2.2 AGORA

AGORA とは，ゴールモデルに「貢献度」と「満足度」という二種類の属性値を付加し，ゴール間のコンフリクトやステークホルダ間の認識不一致を検出する手法である．AGORA により作成されたゴールモデルの例を図 2 に示す．

貢献度とは，親ゴールとサブゴールを結ぶ枝に振られる値であり，-10 から 10 までの整数値をとる．この値は，サブゴールが達成された場合に，その親ゴールに対してどの程度貢献しているかの度合いを示すものである．値が大きいほど親ゴールの達成により大きく貢献することを意味する．

満足度は，各ステークホルダがそのゴールに対してどの程度賛成もしくは満足しているかの度合いを示すものである．貢献度と同様に，-10 から 10 の整数値をとる．ただし，自身の評価だけでなく，他のステークホルダがどの程度満足するのかを推測しながら，他のステークホルダの立場に立った評価も行う．そのため，満足度は行列として個々のゴールに振られるようになる. この行列を「満足度行列」という．満足度行列の例を図 3 に示す．

図 3 の例では，一行目は顧客の評価であり，左から順に，顧客（C）自身の立場での評価，管理者（A）の立場に立った評価，開発者（D）の立場に立った評価になる．この満足度行列では，顧客自身や管理者自身は低い評価を付けているが，開発者自身は高い評価を付けている．このことから，開発者はこのゴールを達成すべきだと考えているが，

© 2017 Information Processing Society of Japan

ユーザにとっては不要なゴールであるということがわかる．このようなステークホルダ間の認識不一致を検出した場合，ネゴシエーションなどの解消作業が必要になる．

AGORA では，貢献度や満足度行列を用いて，分析者が最終的にどのゴールを選択するかを判断する．

2.3 ゴール指向要求分析の問題

ステークホルダの要求が変化する要因として，環境の変化が挙げられる．例えば，市場動向や法令などの外部環境や，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が変わることで，ステークホルダの要求も変化する．そのため，ゴールモデルが完成し，ステークホルダ間の合意が取れた後に環境が変化すると，ゴールモデルを修正する必要がある．しかしゴールモデルを修正するためには，一つひとつゴールを確認しながら，要求変更に対する影響がないか確かめる必要がある．AGORA の場合はさらに，貢献度や満足度行列も影響がないか確認する必要がある．したがって，環境の変化によるゴールモデルへの影響の範囲を特定するには手間がかかるという問題がある．

そこで本研究では，その問題を解決する手法を提案する．なお本研究では，非機能要求に着目し，環境の変化による影響の範囲を特定する．非機能要求はシステム全体に関わるものが多く，要求を満たさないシステムはステークホルダに受け入れられない可能性が高いため，機能要求と同等以上に重要な要求である [3]．そのため，非機能要求に関するゴールへの影響範囲の特定を行う．

3. 関連研究

3.1 環境要因の分析による要求の影響範囲特定手法

中谷らは，ステークホルダを取り巻く環境を基に環境要因を分析し，変化が発生しそうな要求箇所を特定する手法を提案している．ステークホルダ間の依存関係を表す SD モデルを拡張し，外部環境も表現することで，環境の変化が発生した際に変更が起こり得る要求箇所を特定している [4]．

この手法では，考慮する環境が多くなるほどモデルが複雑になり，分析が困難になるという問題がある．そのため，本研究では手間を抑えつつ，環境要因によるゴールモデルへの影響の範囲を特定する手法を提案する．

3.2 ゴール修正時の影響範囲特定手法

佐伯らは，ゴールモデルにおけるゴールの追加や消去時に影響を及ぼす範囲を特定する手法を提案している [6]．品質特性である「Security」や「Usability」などをセマンティックタグとして各ゴールに付与し，タグ間の対立関係によって影響範囲を特定している．しかし，この手法では環境の変化を基に修正すべきゴールを特定することはできない．ただし，この手法により手間を抑えつつ影響範囲を特定することができるため，本研究ではこの手法を応用する．

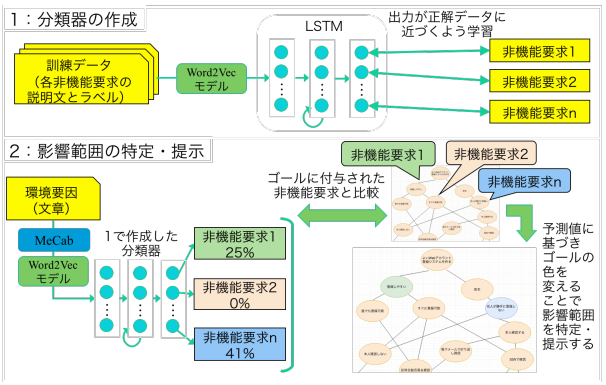


図 4 提案手法の全体図

4. 環境要因による影響範囲特定手法

本研究では，環境要因によるゴールモデルの影響範囲を特定する上で，環境要因として与えられることを前提とする．それを，深層学習のアルゴリズムの一種である LSTM (Long Short Term Memory) で作成した分類器を入力することで，その環境要因がどの非機能要求に対して影響を与える確率が高いかを求める [5]．分類器は，入力層を環境要因の単語ごとのベクトル値，中間層を LSTM，出力層を各非機能要求としたニューラルネットワークとする．

ステークホルダがゴールモデルを作成する際，3.2 のように，各ゴールに非機能要求をタグとして付与しておく．それらの非機能要求と，入力された環境要因に対して分類器で判定された非機能要求を照らし合わせることで，環境要因によるゴールモデルへの影響の範囲を特定する．特定後，予測値に基づきゴールの色を変更することで分析者に提示する．図 4 に提案手法の全体図を示す．

環境要因によるゴールモデルへの影響の範囲を特定するために，環境要因を各非機能要求に分類しなければならない．そのために，分類する非機能要求の種類の定義を行った後，分類器を作成する必要がある．分類器の作成には，各非機能要求の説明文が大量に必要である．それぞれの詳細について以下に述べる．

4.1 深層学習による非機能要求分類

4.1.1 分類する非機能要求の種類の定義

まず，分類する非機能要求の種類を定義する必要がある．本研究では，鵜飼らの研究を参考に，機能性など 8 種類を定義した [7]．定義した非機能要求を表 1 に示す．機能性～移植性の 6 種類は ISO/IEC9126-1 で規定されている品質特性である [8]．セキュリティは，ISO/IEC9126-1 では信頼性の品質副特性として規定され，ISO/IEC25010 では品質特性として規定されている [9]．コストは品質特性ではないが，非機能要求としては重要な項目であると考えられる．

© 2017 Information Processing Society of Japan

表 1　定義した 8 種類の非機能要求

非機能要求	内容
機能性	明示的及び暗示的必要性に合致する機能を提供する度合い
信頼性	機能が指定された達成水準を維持する度合い
使用性	理解，習得，利用の容易性の度合い
効率性	目的達成のために使用する資源の度合い
保守性	保守作業の容易性の度合い
移植性	別環境へ移した際にそのまま動作する度合い
セキュリティ	不正アクセスされないよう，情報またはデータを保護する能力の度合い
コスト	システムを開発する費用や手間

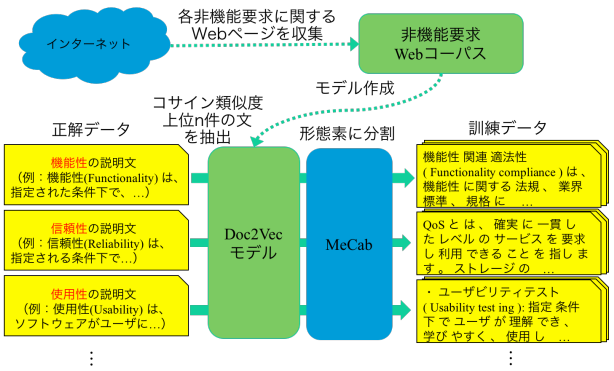


図 5　訓練データ作成のフロー

4.1.2　訓練データの作成

分類器を作成するには，大量の訓練データが必要になるが，それらを手作業で用意するのは困難である．そこで，非機能要求ごとに，ある一つの説明文を正解データとして用意し，大量の文書の中から説明文に類似する文を抽出することにより，訓練データを作成する．正解データの要件は，各非機能要求を簡潔に説明することができていることとする．訓練データ作成のフローを図 5 に示す．

類似文の抽出には Doc2Vec を用いる [10]．Doc2Vec とは，Word2Vec を拡張したモデルであり，テキストのコーパスに基づき任意の長さの文書をベクトル化し，分散表現を獲得することができる技術である [10], [11]．これにより，文書間の類似度の算出を行うことができる．類似文の抽出をする際には，コサイン類似度の上位 n 件を抽出することとする．

Doc2Vec で用いるコーパスには，非機能要求に関連のある文を抽出するために，Web コーパスを作成し，用いることとする [12]．Web コーパスとは，Web ページを情報源としたコーパスであり，ある特定の分野に特化したコーパスを容易に作成することができる．作成した非機能要求の Web コーパスを利用することで，多様な非機能要求の説明文を抽出することが期待できる．

非機能要求の Web コーパスから抽出した，それぞれの正解データの類似文を，形態素解析器である MeCab を用いて形態素に分割し，訓練データとした [13]．形態素に分割する理由は，後述する分類器に入力する際に必要となるためである．

4.1.3　分類器の作成

本研究では，LSTM を用いて分類器を作成する．LSTM は RNN (Recurrent Neural Network) という深層学習の手法を拡張したものである．RNN は過去の文脈を考慮することができ，より精度が高い分類ができるという特徴を持つ．例えば，「クラウドにより，オンプレミスの運用負担が大幅軽減できる」と「オンプレミスにより，クラウドの運用負担が大幅軽減できる」という二つの文章は逆の意味を持つが，従来の手法だと同じ文章として取り扱われしまう．RNN は文脈を考慮することで，異なる文章として扱うことができる．しかし RNN には，長い系列を入力するとパラメータの勾配が爆発もしくは消失し，長い履歴を考慮できなくなるという問題がある．この問題を解決しているのが LSTM である．

LSTM の入力はベクトル値である必要があり，訓練データを事前にベクトル化しなければならない．本研究では Word2Vec により得られる分散表現を用いることにする．Word2Vec とは，Doc2Vec と同じくテキストのコーパスを入力すると，コーパスにある単語の特徴量ベクトルを獲得することができる自然言語処理ツールである．これにより，訓練データに存在しない単語 (未知語) にも対処することができる．4.1.2 で訓練データを形態素に分割しておく理由は，Word2Vec で形態素ごとにベクトル化するためである．

訓練データの各文書をベクトル化した後に，本章の最初に述べたニューラルネットワークを構築する．それに訓練データを学習させ，非機能要求の分類器を作成する．

4.2　影響箇所の特定・提示

作成した分類器を基に，環境要因によるゴールモデルへの影響の範囲を特定・提示する．そのために，ゴールモデルを作成することができるシステムを構築する．そのシステムに，文章として表現された環境要因を入力することで，作成済みのゴールモデルへの影響範囲を特定・提示する．システムは，入力された文章を MeCab で形態素に分割し，Word2Vec で形態素ごとのベクトル値を獲得した後に，それらを 4.1.3 で作成した分類器に入力することで，各非機能要求に判別される確率を予測値として算出する．ゴールモデルに付与された非機能要求のタグと，分類器で判別された非機能要求を照らし合わせ，一致する非機能要求が存在した場合，予測値に基づきゴールの色を変更して分析者に提示する．予測値が 80〜100%であれば赤色に，60〜79%であれば紫色に，40〜59%であれば青色に，20〜39%であれば緑色に色を変更する．0〜19%の場合は色を変更しない．影響箇所の提示の例を図 6 に示す．

図 6 の例では，初期ゴールとして「よい Web アカウント

© 2017 Information Processing Society of Japan

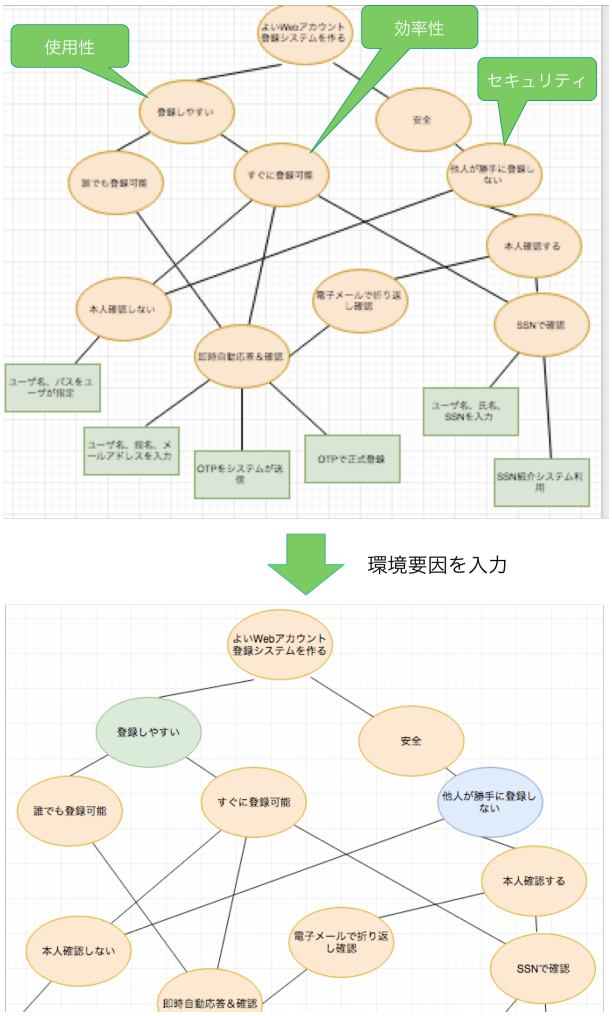


図 6 影響箇所特定・提示の例

登録システムを作る」を設定し，「登録しやすい」，「安全」の二つのゴールに分解している．そして，「登録しやすい」から「誰でも登録可能」と「すぐに登録可能」に，「安全」から「他人が勝手に登録しない」にそれぞれ分解し，その後も分解を続けている．このゴールモデルにおいて，「登録しやすい」には「使用性」のタグが，「他人が勝手に登録しない」には「セキュリティ」が，「すぐに登録可能」には「効率性」のタグが付与されている．

このゴールモデルは，すでにステークホルダの合意が得られているとする．その後，「他企業でアカウントが漏洩」という環境の変化が発生したと仮定する．その場合，ステークホルダの要求も変化する可能性があるため，ゴールモデルに修正箇所がないか確認する必要がある．

その際に環境要因として「他企業でアカウントが漏洩」を入力すると，各非機能要求の予測値はそれぞれ「使用性」が 25%，「効率性」が 0%，「セキュリティ」が 41%になったとする．その場合，「セキュリティ」のタグが付与されたゴールは青色に，「使用性」のタグが付与されたゴールは緑色になる．「効率性」のタグが付与されたゴールは色を変更しない．これにより，環境要因によるゴールモデルへの影響範囲を特定・提示することができる．提示された範囲内でゴールの内容を分析者が確認し，適時修正を行うことになる．この手法により，修正すべきゴールを特定する手間を抑えることができる．

5. 分類器の精度の評価と考察

5.1 実験方法

提案手法のうち，4.1.3 で述べた分類器の有効性を確認するために，実際に分類器を作成し，精度を評価した．まず，Web コーパスの作成を行った．コーパスの作成に必要な Web ページの収集は，検索エンジンである Bing の Web Search API を利用した [14]．クエリには，各非機能要求 (「機能性」など) に加え，「非機能要求」という単語を設定した．この理由は，各非機能要求のみの場合には，非機能要求とは関係がないページが検索結果に散見されたためである．そして，各 Web ページのコンテンツを HTML ファイルとして，各非機能要求ごとに重複なしで 700 件収集した．その後，日本語 Web コーパスを作成するためのツールである nwc-toolkit を用いて，HTML ファイルからテキストを抽出し，テキストファイルを生成した [15]．

次に，これらのテキストファイルをそれぞれ MeCab で形態素に分割し，一つのテキストファイルにマージすることでコーパスを作成した．MeCab の辞書には新語や固有表現も抽出することができる mecab-ipadic-NEologd を用いた [16]．また，「機能性」と「使用性」は，「機能」と「性」のように分割されたため，それぞれの単語を辞書に追加して対応した．作成したコーパスの容量は 82MB になった．

続いて，このコーパスを基に，Doc2Vec を用いて訓練データを作成した．正解データには，正解データの要件が満たされていた，オージス総研オブジェクトの広場の「非機能要求と ISO9126」の説明文を利用した [17]．ただし「移植性」は，同じ意味である「可搬性」として表記されていたため，「可搬性」の説明文を「移植性」の正解データとして用いることにした．また，「コスト」は記載されていなかったため，コーパス中から適切と思われる文章を抽出して利用した．訓練データを作成する上で使用した正解データの内容の一部を以下に記す．

- 機能性：「機能性 (Functionality) は、指定された条件下で、ソフトウェアがユーザニーズを満たすために提供する能力を表します。」 *1
- 移植性：「可搬性 (Portability) は、ソフトウェアの異なる環境への移しやすさを表します。」 *1
- コスト：「また、コストや品質に影響を与える度合いが大きい非機能要求項目を重要項目と定義している。重要項目は要求分析のなるべく早い段階で決めること

*1 https://www.ogis-ri.co.jp/otc/hiroba/technical/JavaPress_ISO9126/, (accessed 2017-02-09).

© 2017 Information Processing Society of Japan

表 2 Word2Vec モデルを作成した際のハイパーパラメータ

パラメータ	値
次元数	200
単語を無視する際の出現頻度の閾値	1
ウィンドウサイズ	15

表 3 分類器を作成した際のハイパーパラメータ

パラメータ	値
入力層のユニット数	200
中間層	1 層の LSTM
中間層のユニット数	256
バッチサイズ	8
シーケンス長	281
ドロップアウト	0.2
最適化手法	Adam
学習率	0.001
エポック数	50
活性化関数	ソフトマックス関数
損失関数	クロスエントロピー関数
出力層のユニット数	8

が望ましい。」 [*2]

非機能機能ごとに類似文を 100 件ずつ，計 800 件抽出した．これらの類似文を MeCab で分かち書きし，訓練データとした．以下に機能性，信頼性，使用性の訓練データの例を順に示す．なお，分類器の評価のために，作成した 800 件の訓練データのうち，90%(720 件) を訓練データ，10%(80 件) をテストデータとして用いた．

- 機能性 関連 適法性 (Functionality compliance) は 、 機能性 に関する 法規 、 業界標準 、 規格 に ソフトウェア が 沿っ て いる か を 表し ます 。
- 時間 挙動 (Time behavior) は 、 指定 さ れ た 条件下 で 、 ソフトウェア が 適切 な 応答時間 、 処理 時間 、 スループット で 機能 を 実行 する 能力 を 表し ます 。
- ・ ユーザビリティテスト (Usability test ing): 指定 条件下 で ユーザ が 理解 でき 、 学び やすく 、 使用 し やすい システム か を 判定 する ため の テスト

訓練データとテストデータの単語それぞれに Word2Vec を用いてベクトル値を獲得した．Word2Vec のモデルを作成する際に利用したコーパスは，Doc2Vec で利用した Web コーパスと同じものである．Word2Vec のモデルには，skip-gram モデルを採用した．Word2Vec のモデルを作成した際のハイパーパラメータを表 2 に示す．

続いて，TensorFlow ベースの深層学習ラッパーライブラリである Keras を用いて分類器を作成した [18], [19]．表 3 に分類器を作成した際のハイパーパラメータを示す．勾配法にはミニバッチ法を用いた．

その後，環境要因として「他企業でアカウントが漏洩」と「東日本大震災の発生」を設定し，分類器に入力した．それぞれの環境要因が，どの非機能要求に影響を与える確率が高いと判別されるか確認した．また，学習過程における精度と誤差率を確認し，十分な学習が行われているかどうかについても確認した．

[*2] http://taityo-diary.hatenablog.jp/entry/2015/09/28/063507, (accessed 2017-02-09).

表 4 「他企業でアカウントが漏洩」を入力した場合の分類結果

非機能要求	予測値
使用性	0.32
セキュリティ	0.20
効率性	0.18
機能性	0.14
信頼性	0.08
コスト	0.04
移植性	0.03
保守性	0.00

表 5 「東日本大震災の発生」を入力した場合の分類結果

非機能要求	予測値
信頼性	0.48
保守性	0.38
使用性	0.04
移植性	0.03
効率性	0.03
コスト	0.01
機能性	0.01
セキュリティ	0.00

5.2 実験結果

構築した分類器に「他企業でアカウントが漏洩」という環境要因を入力した結果を表 4 に示す (小数点以下第 3 位を四捨五入).「セキュリティ」と判別される確率が一番高くなることが期待されたが，二番目に高い確率となった．一番確率が高かった非機能要求は「使用性」で，約 32%の確率でこの環境要因と関連している．セキュリティは約 20%の確率で関連しているという結果になった．

続いて，「東日本大震災の発生」という環境要因を入力した結果を表 5 に示す (小数点以下第 3 位を四捨五入)．信頼性が約 48%と一番高くなり，その次に保守性が約 38%という結果になった．この二つは震災と関わりが強いと考えられるため，妥当な結果になったといえる．

次に，学習過程における精度の評価を行った．TensorFlow の可視化ツールである TensorBoard を用いて，訓練データとテストデータそれぞれに対する精度と誤差率を確認した．TensorBoard により生成されたグラフを図 7，8，9，10 に示す [20]．縦軸は図 7,9 が精度，図 8,10 が誤差率であり，横軸はエポック数である．

図 7 は訓練データに対する精度を示している．エポック数が増加するごとに，精度が向上していることがわかる．最終的に，約 95%の値で安定しているため，十分な結果といえる．図 8 は訓練データに対する誤差率を示している．

© 2017 Information Processing Society of Japan

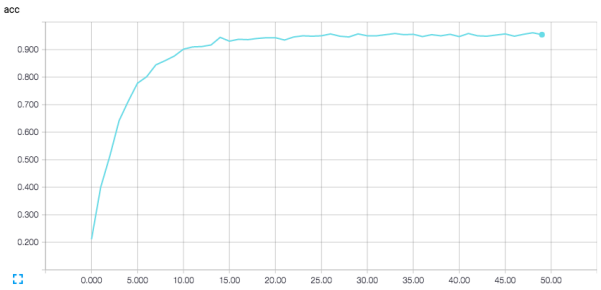


図 7　訓練データに対する精度

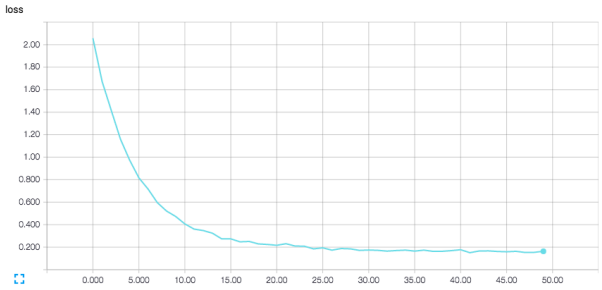


図 8　訓練データに対する誤差率

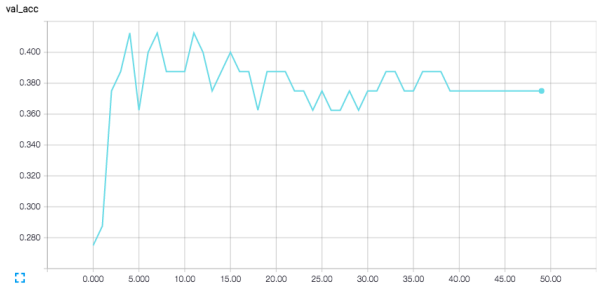


図 9　テストデータに対する精度

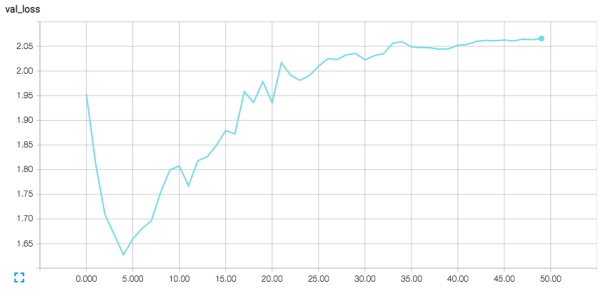


図 10　テストデータに対する誤差率

エポック数が増えるにつれて，誤差率が減少していることがわかる．

一方，図 9 はテストデータに対する精度を示しているが，最終的に約 30%の値で安定している．訓練データに対する精度と比較すると，65%ほど低い値であり，分類器の精度としては不十分である．図 10 はテストデータに対する誤差率を示している．誤差率はエポック数が 5 を超えた辺りから増加し続けている．テストデータに対する精度と誤差率から，未知のデータに対しては正しく判別できていないことがわかる．

5.3　考察

環境要因として「東日本大震災の発生」を入力した場合は妥当な結果になったが，「他企業でアカウントが漏洩」を入力した場合は，期待どおりの結果にはならなかった．セキュリティの訓練データに「アカウント」や「漏洩」などの単語が含まれていなかったことが原因として考えられる．改善案として，訓練データを作成する際に非機能要求の説明文だけでなく，例文も用いることが挙げられる．例えば，セキュリティに関する例文として「重要情報の情報漏えい対策を」[*3]などがある． このような例文を訓練データに用いることで，分類器の性能が向上すると考えられる．

また，訓練データに対する精度と誤差率は十分ではあったが，テストデータに対する精度と誤差率は十分ではなかった．原因として，訓練データの質と量が共に不足していることが挙げられる．量を増やすには，類似文の取得数を 100 件から増加させれば良いが，その分，質の低下が見込まれる．質を向上させるには，手作業でデータを集めるという手段の他に，Doc2Vec で用いた Web コーパスの質を向上させる, ということが考えられる．Web コーパスの中身を確認すると，大量のノイズが含まれていることが分かった．例えば，「ソフトウェア」の「信頼性」ではなく，「食材」,「材料」の「信頼性」に関連する文章などがある．これらが含まれた原因は，Web ページを収集する際に，完全一致検索を行わなかったことにより，「非機能要求」が含まれず，「信頼」や「信頼性」が含まれる Web ページを収集したためである．

これらのノイズに影響を受けたため，類似文の抽出精度が低くなってしまったと考えられる．そのため，Web コーパスを作成する際に，「食材」や「材料」に関する単語などをあらかじめストップワードとして設定し，ノイズを除去する必要がある．

6.　おわりに

本研究では，環境要因によるゴールモデルへの影響の範囲の特定を支援することを目標とした．そのためにまず，ステークホルダがゴールモデルにあらかじめ非機能要求をタグとして付与しておく．それらと，文章として表現された環境要因を深層学習で作成した分類器に入力し，出力された非機能要求と照らし合わせることで，ゴールモデルへの影響の範囲の特定を行った．また，分類器の予測値に基づきゴールの色を変更することで，分析者に影響の範囲を提示する手法を提案した．

分類器の精度を評価する実験を行った結果，環境要因を各非機能要求に十分に分類することができる結果とはならなかった．分類器の訓練データの質や量が精度に大きく影響を与えるため，これらを改善する必要がある．

[*3] https://japan.zdnet.com/article/35011544/2/, (accessed 2017-02-09).

© 2017 Information Processing Society of Japan

参考文献

[1] Lamsweerde, A.V.: Goal-Oriented Requirements Engineering: A Guided Tour, Proc. *the Fifth IEEE International Symposium on Requirements Engineering* (*RE'01*), pp.249-263 (2001).
[2] Kaiya, H., Horai, H. and Saeki, M.: AGORA:AttributedGoal-Oriented Requirements Analysis Method, Proc. *IEEE Joint International Conference on Requirements Engineering* (*RE'02*), pp.13-22 (2002).
[3] 大森久美子，岡崎義勝，西原夫：ずっと受けたかったソフトウェアエンジニアリングの新人研修 第 2 版，pp.61，翔泳社（2014）.
[4] 中谷多哉子，妻木俊彦：要求変更の予測を行うための分析枠組みの提案，電子情報通信学会技術研究報告，KBSE，知能ソフトウェア工学，Vol.113，No.277，pp.43-48（2013）.
[5] Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, *Neural Computation*, Vol.9, No.8, pp.1735-1780 (1997).
[6] 佐伯元司，林晋平，海谷治彦：属性つきゴール指向要求分析法の支援のための統合ツール，電子情報通信学会技術研究報告，KBSE，知能ソフトウェア工学，Vol.109，No.41，pp.13-18（2009）.
[7] 鵜飼孝典，林晋平，佐伯元司：ソフトウェア開発会議におけるステークホルダと関心事の可視化ツール，研究報告グループウェアとネットワークサービス，Vol.2011-GN-79，No.16，pp.1-8（2011）.
[8] ISO/IEC 9126-1:2001, Software engineering – Product quality – Part 1:Quality model (2001).
[9] ISO/IEC 25010:2011, Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models (2011).
[10] Le, V.Q. and Mikolov, T.： Distributed Representations of Sentences and Documents, Proc. *the 31st International Conference on Machine Learning* (*ICML 2014*), pp.11881196 (2014).
[11] Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed representations of words and phrases and their compositionality, Proc. *the 26th International Conference on Neural Information Processing Systems* (*NIPS 2012*), pp.3111-3119 (2013).
[12] 関口洋一，山本和英：Web コーパスの提案，情報処理学会研究報告，学会自然言語処理研究会，NL-157-17，pp.123-130（2003）.
[13] Kudo, T., Yamamoto, K., and Matsumoto, Y.: Applying Conditional Random Fields to Japanese Morphological Analysis", Proc. *the 2004 Conference on Empirical Methods in Natural Language Processing* (*EMNLP 2004*), pp.230-237 (2004).
[14] Microsoft: Web Search API, Learn to Develop with Microsoft Developer Network (online), available from 〈`https://msdn.microsoft.com/en-us/library/mt711415.aspx`〉, (accessed 2017-02-09).
[15] zen: xen/nwc-toolkit: Automatically exported from code.google.com/p/nwc-toolkit, GitHub (online), available from 〈`https://github.com/xen/nwc-toolkit`〉, (accessed 2017-02-09).
[16] neologd: neologd/mecab-ipadic-neologd: Neologism dictionary based on the language resources on the Web for mecab-ipadic, GitHub (online), available from 〈`https://github.com/neologd/mecab-ipadic-neologd`〉, (accessed 2017-02-09).
[17] 株式会社オージス総研グループ：アクセス非機能要求と ISO9126 — オブジェクトの広場（オンライン），入手先 〈`view-source:https://www.ogis-ri.co.jp/otc/hiroba/technical/JavaPress_ISO9126/`〉（参照 2017-02-09）.
[18] Abadi, M. Agarwal, A. Barham, P., et al.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, Proc. *the 12th USENIX conference on Operating Systems Design and Implementation* (*OSDI '16*), pp.265-283 (2016).
[19] Keras: Keras Documentation (online), available from 〈`https://keras.io/`〉, (accessed 2017-02-09).
[20] TensorFlow: TensorBoard: Visualizing Learning — TensorFlow (online), available from 〈`https://www.tensorflow.org/how_tos/summaries_and_tensorboard/`〉, (accessed 2017-02-09).

© 2017 Information Processing Society of Japan